Application System/400™

SC21-9590-1

Communications: Programmer's Guide

Application System/400™

SC21-9590-1

Communications: Programmer's Guide

Second Edition (September 1989)

This major revision makes obsolete SC21-9590-0.

See "About This Manual" for major changes to this edition of the manual.

This edition applies to Release 2, Modification Level 0, of the IBM Operating System/400 Licensed Program (Program 5728-SS1), and to all subsequent releases and modifications until otherwise indicated in new editions or technical newsletters.

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

Changes are periodically made to the information herein; all such changes will be reported in subsequent revisions or technical newsletters.

This publication is for planning purposes only. The information herein is subject to change before the products described become available. Also, 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 for 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. To report an error, use the Reader's Comment Form at the back of the publication. If the form has been removed, address your comments to: IBM Corporation, Information Development, Department 245, Rochester, Minnesota, U.S.A. 55901. IBM reserves the right to use or distribute the information you supply in any way it believes appropriate without incurring any obligation to you.

About This Manual

This reference manual contains programming information for writing application programs that use the OS/400 intersystem communications function (OS/400-ICF), hereafter referred to as ICF.

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

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

Who Should Use This Manual

This manual is intended primarily for the AS/400 system application programmers and remote system application programmers who write communications programs that use ICF.

What You Should Know

Before using this manual, you should be familiar with the following information:

- The AS/400 system programming terminology and programming using C/400, COBOL/400, or RPG/400.
- Concepts of data communications as described in *Data Communications Concepts*, GC21-5169.

How This Manual Is Organized

This manual is organized as follows:

- Chapter 1 introduces you to AS/400 system communications and describes system planning, installing, configuring and writing programs that use ICF.
- Chapter 2 discusses the types of communications you can accomplish using the AS/400 system ICF.
- Chapter 3 through Chapter 5 provide information about ICF sessions and files.
- Chapter 6 through Chapter 8 discuss the use of DDS keywords and system-supplied formats, as well as AS/400 system programming considerations.
- Chapter 9 through Chapter 11 provide program explanations and examples for C/400, COBOL/400, and RPG/400.
- Chapter 12 shows how to trace intersystem communications functions and operations using the TRCICF command.
- Appendix A contains tables of language operations, DDS keywords, and system-supplied formats.

- Appendix B contains information about communications error handling, and provides tables and descriptions of ICF return codes as well as reason codes for failed program start requests.
- Appendix C contains information about the open and I/O feedback area.
- Appendix D provides charts that show the EBCDIC and ASCII character sets.
- Appendix E describes the File Transfer Support (FTS) function.

The last part of this manual contains 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 the page on which the topic is covered.

How This Manual Has Changed

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

- A description of the three new communications types supported by ICF: finance, intrasystem, and retail communications.
- A description of selective prompting for the CMNTYPE parameter on the program device entry commands in Chapter 4.
- The addition of C/400 program segments showing how system-supplied formats are used, described in Chapter 7.
- A description of prestart jobs for communications in Chapter 8.
- C/400 program explanations and examples in Chapter 9.
- Improvements made to all COBOL/400 and RPG/400 program examples in Chapter 10 and Chapter 11.
- A description of the Trace ICF function in Chapter 12.
- Changes to the presentation of major and minor return codes in Appendix B. Tables have been added to show all the communications types and the return codes that are valid for each. Descriptions of the return codes have also been updated.
- The addition of C/400 and CL program examples for file transfer support in Appendix E. Improvements have also been made to the COBOL/400 and RPG/400 program listings for file transfer support.

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 following AS/400 system manuals contain information you may need.

- *Communications: Advanced Program-to-Program Communications and Advanced Peer-to-Peer Networking User's Guide*, SC21-9598, provides the information necessary to define and use the AS/400 system Advanced Program-to-Program (APPC) and Advanced Peer-to-Peer Networking (APPN).
- *Communications: Asynchronous Communications Programmer's Guide*, SC21-9592, contains information for creating an asynchronous communications definition, writing programs that use asynchronous communications, and responding to return codes.

- *Communications: BSC Equivalence Link Programmer's Guide*, SC21-9593, provides information on using the Binary Synchronous Communications Equivalence Link with the IBM AS/400 system.
- *Communications: Intrasystem Communications Programmer's Guide*, SC21-9864, contains information about how two application programs can communicate with each other on the same AS/400 system.
- *Communications: Communications and Systems Management User's Guide*, SC21-9661, contains information about preparing a system for remote management activities and about using the change management feature distributed systems node executive (DSNX).
- *Communications: Distributed Data Management User's Guide*, SC21-9600, contains the information needed to use DDM on a network. It includes AS/400 system DDM concepts, preparing for DDM communications, and all DDM-related programming information needed by the DDM programmer.
- *Communications: Finance Communications Programmer's Guide*, SC21-8099, provides information on financial communications between devices at multiple locations.
- *Communications: Retail Communications Programmer's Guide*, SC21-9858, provides information for setting up and starting retail communications between devices at multiple locations.
- *Communications: Remote Job Entry Facility User's Guide and Reference*, SC09-1168, provides information for using the Communications Utilities remote job entry (RJE) to submit jobs to an IBM host processor.
- *Communications: SNA Upline Facility Programmer's Guide*, SC21-9594, contains information on using the Systems Network Architecture Upline Facility with the IBM AS/400 system.
- *Communications: Transmission Control Protocol/Internet Protocol Guide* (available at a later date), provides information on TCP/IP configurations, IBM-supplied applications, and user-written applications.
- *Communications: User's Guide*, SC21-9601, provides communications support information for the AS/400 system. This includes setting and changing communications values, and starting and stopping communications.
- *Communications: 3270 Device Emulation User's Guide*, SC21-9602, provides information on setting up and starting either 3270 Binary Synchronous Communication (BSC) or Systems Network Architecture (SNA) device emulation.
- *Network Planning Guide*, GC21-9861, provides an overview of network planning tasks and helps you determine which communications functions to use. This manual is a starting point for network planning.
- *Data Communications Planning Guide*, GA21-9902, provides information on how to order and plan for communications lines and modems, and also helps the customer gather communications configuration information at planning time.
- *Languages: C/400 User's Guide* (available at a later date), provides the information needed to use the C/400 programming language to code programs for the AS/400 system.
- *Languages: COBOL/400 User's Guide*, SC09-1158, provides the information needed to write, test and maintain COBOL/400 programs for the AS/400 system.

- *Languages: RPG/400 User's Guide*, SC09-1161, provides the information needed to use the RPG/400 programming language to code programs for the AS/400 system.
- *Programming: Control Language Programmer's Guide*, SC21-8077, provides a wide-ranging discussion of AS/400 system programming topics.
- *Programming: Control Language Reference*, SBOF-0481, provides information on Control Language commands.
- *Programming: Data Description Specifications Reference*, SC21-9620, contains information about coding data description specifications for physical, logical, display, printer, and ICF files.
- *Programming: Data Management Guide*, SC21-9658, provides information to help the system programmer manage key aspects of the system, such as how to override and copy files, describe device files to the system, and create jobs and output queues.
- *Programming: Database Guide*, SC21-9659, provides information about the AS/400 database management system, and describes how to set up and use a database on the AS/400 system.
- *Programming: Security Concepts and Planning*, SC21-8083, contains information about general security concepts for the system.
- *Programming: System Reference Summary*, SC21-8104, contains information summaries of system values, DDS keywords, and summary charts.
- *Programming: Work Management Guide*, SC21-8078, contains information on how to create and change a work management environment.
- *System/36 System Reference*, SC21-9020 contains information on System/36 naming conventions you might need for file transfer support (FTS).
- *System Operations: Operator's Guide*, SC21-8082, provides information for the system operator on how to use the system unit control panel.

Contents

Figures

Chapter 1. Introduction to AS/400 System Communications

This chapter describes, in general, the different sources and background information needed to use communications on the AS/400[1] system. Detailed instructions are available in other manuals referred to in this chapter.

Planning for Data Communications

Data communications planning should already be complete. The *Network Planning Guide* contains information to help you determine which communications functions to use. The *Data Communications Planning Guide* contains information to help you plan for communications lines and modems.

Installing Communications Hardware

Communications hardware, such as modems and cables, must be installed before you can start running your programs. However, if your hardware is not yet installed, you can read this manual and begin writing your programs.

Configuring Your System for Data Communications

The *Communications User's Guide* explains data communications configuration. Although you cannot run your programs until the system is properly configured, you can read this manual and begin writing your programs.

You may need to configure the remote system to allow communications with the AS/400 system. The *Communications User's Guide* contains configuration considerations for remote systems that require configuration for the AS/400 system.

Writing Programs That Use the Intersystem Communications Function (ICF)

You can write communications programs using C/400[1], COBOL/400[1], or RPG/400[1]. For an explanation of the communications application interface provided by the intersystem communications function (ICF), read Chapter 3 through Chapter 8. You can then refer to Chapter 9 through Chapter 11 for programming examples that you can use to help write and run programs on the AS/400 system.

You also need the appropriate communications programming manual for the communications type you are using (for example, the *APPC and APPN User's Guide*), the programming language manuals for the language you plan to use, and the *DDS Reference.*

---

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

Operating Communications on the AS/400 System

To use communications on the AS/400 system, you must be familiar with the base operating system as well as the commands unique to communications. Refer to the *Operator's Guide*, the *Command Reference Summary*, and the *CL Reference* for information on the general operation of the system.

Chapter 2. Communications Features

This chapter introduces the AS/400 system communications features, including:

- ICF communications types
- AS/400 system communications line support
- Base operating system support
- High-level language support
- Additional programming support

Intersystem Communications Function Communications Types

Communications between application programs are accomplished using the AS/400 system ICF and underlying support provided by various **communications types**. Several communications types are provided so that the AS/400 system can communicate with remote systems having different communications methods. Some of the communications methods are:

- Binary synchronous communications (BSC)
- Systems Network Architecture (SNA). Examples of SNA are:
  - Systems Network Architecture upline facility (SNUF)
  - Advanced program-to-program communications (APPC)
  - Finance communications
  - Retail communications
- Asynchronous communications

A communications type, designed for a specific remote system, makes it unnecessary to handle most system-dependent and protocol considerations when coding the AS/400 system application programs. The following communications types are supported by ICF:

- Advanced program-to-program communications (APPC)
- Systems network architecture upline facility (SNUF)
- Binary synchronous communications equivalence link (BSCEL)
- Asynchronous communications
- Intrasystem communications
- Finance communications
- Retail communications

An AS/400 system program uses high-level language operations and **communications functions** to communicate with a remote system through ICF. A **return code**, made up of major and minor return codes, informs the program of the success or failure of each operation. You can use several of the communications functions and return codes with any of the communications types. You can use some functions and return codes with only one or two communications types. You can use a program written for use with one communications type, with little or no change, to communicate with a different communications type. The level of change required in the program depends on the two communications types, the communications functions, and the return codes.

Your configuration device descriptions identify the devices on your local system with which communications occur. Each communications type has a corresponding configuration device description of the same type.

AS/400 System Communications Types

The following is a description of the AS/400 system communications types supported by ICF, including a brief overview of the remote systems and devices supported by each type.

Advanced Program-to-Program Communications (APPC)

APPC allows the system to communicate with other IBM systems that support the SNA logical unit type 6.2 (LU6.2) architecture. Using APPC allows the system to communicate with:

- Another AS/400 system
- A System/38
- A System/36
- Any other IBM systems (such as the Customer Information Control System for Virtual Storage, CICS/VS) with similar levels of APPC support (refer to the *APPC and APPN User's Guide* for more specific information on supported systems).

APPC allows AS/400 system application programs to start programs on remote systems, and allows remote programs to start programs on the AS/400 system. The networking capability of Advanced Peer-to-Peer Networking (APPN) is available through the APPC interface.

Refer to the *APPC and APPN User's Guide* for more information.

Systems Network Architecture Upline Facility (SNUF)

SNUF allows the system to communicate with CICS and Information Management System (IMS) applications on other IBM systems. You can use SNUF to communicate with the following host systems:

- System/370
- 30xx
- 43xx

SNUF allows AS/400 system application programs to start programs on remote host systems, and allows programs on remote host systems to start programs on the AS/400 system. Both interactive and batch operations are supported.

3270 Application Program Interface (3270 API): The SNUF 3270 API allows an AS/400 application to communicate with a 370, 30xx, or 43xx VTAM host application by sending and receiving 3270 data streams.

Refer to the *SNA Upline Facility Programmer's Guide* for more information.

Binary Synchronous Communication Equivalence Link (BSCEL)

AS/400 system BSCEL provides the following:

- Distributed data processing support to the AS/400 system users who want to communicate with another system or device at a remote location using BSC.
- Online and batch communications between application programs on different systems (such as System/38) using BSC.
- Communications with another AS/400 system, System/36, or System/34 using BSCEL.
- Communications with another AS/400 system, System/36, or System/34, with RPG II support for telecommunications.

BSCEL allows AS/400 system applications to start programs on remote systems that support BSCEL, and allows remote programs to start programs on the AS/400 system.

Refer to the *BSC Equivalence Link Programmer's Guide* for more information.

Asynchronous Communications

The system can use asynchronous communications support to communicate with another asynchronous communications location or with a packet assembler/disassembler (PAD) that gives the system access to an X.25 packet-switching data network (PSDN). The system can use X.25 support to communicate directly through an X.25 network, or to emulate a PAD using the International Telegraph and Telephone Consultative Committee (CCITT) recommendations X.3, X.28, and X.29.

The system can use the AS/400 system asynchronous communications support to communicate with:

- Another AS/400 system
- A System/36
- Asynchronous devices

Asynchronous communications support allows AS/400 system applications programs to start programs on remote systems, and allows remote programs to start programs on the AS/400 system.

Refer to the *Asynchronous Communications Programmer's Guide* for more information.

Intrasystem Communications

Intrasystem communications allows communication between two application programs on the same AS/400 system. A source program can acquire more than one session for a given device description, and can have more than one transaction at the same time. However, a source program cannot have a transaction with two different programs on the same session.

Note: Intrasystem communications does not support the concept of a remote system or a remote program. When these terms are used in this manual with regard to intrasystem communications, they refer to the program with which your program is communicating.

Refer to the *Intrasystem Communications Programmer's Guide* for more information.

Finance Communications

Finance communications allows the AS/400 system to communicate with banking terminals or devices attached to 3694, 4701, and 4702 controllers that support the SNA logical unit type 0 (LU0) architecture.

The AS/400 system communicates with the 3694, 4701, and 4702 finance controllers using the synchronous data link control (SDLC) protocol. The 4701 and 4702 controllers can also use the X.25 data link protocol, and can share the X.25 line with any AS/400 session types possible for X.25 communications.

Refer to the *Finance Communications Programmer's Guide* for more information.

Retail Communications

Retail communications allows you to attach retail controllers (3651, 3684, 4680, 4681, 4684, and 4692) to the AS/400 system using the SDLC protocol.

Note: The 4681 controller is the double-byte character set (DBCS) equivalent of the 4680 controller, and the 4692 is the DBCS equivalent of the 4684 controller.

In addition, retail communications allows the AS/400 system to act as an in-store processor in the retail environment.

You can use the AS/400 system in several different retail environments:

- Retail in-store processor environment

  You can have a host system such as a System/370 at a remote site with several retail controllers and terminals in your store. The AS/400 system can be installed in your store as an in-store processor to coordinate communications between the host and the retail controllers.

- Retail host processor environment

  The AS/400 system can also function as a host system to several retail controllers.

Refer to the *Retail Communications Programmer's Guide* for more information.

Non-Intersystem Communications Function Communications

You can also run non-ICF communications on the AS/400 system, such as the following:

- 3270 device emulation and 3270 BSC application program interface
- Remote job entry (RJE)
- Finance communications
- Transmission Control Protocol/Internet Protocol programs (TCP/IP)

Because these communications functions are not part of ICF, they are described in other manuals, identified in the list of related manuals in "About This Manual" on page iii. Refer also to the *Information Directory*.

Communicating with Remote Work Stations

No communications programming is required to communicate with remote work stations. The necessary communications programs are provided by the system based on the information provided when the remote work station is configured. The application program interface for remote work stations is the same as the program interface for local work stations. Refer to the *Data Management Guide* for information on the application interface to remote work stations.

Combinations of Communications Types

You can configure multiple communications device descriptions in the AS/400 system. Multiple communications configurations can be active at the same time. All active configurations do not have to be of the same type. The number of configurations that can be active is determined by the number of communications lines available, and whether any lines are being shared by SNA-type communications. A configuration becomes active when you vary on the configuration, as described in "Varying on Communications Configurations" on page 3-3.

AS/400 System Communications Line Support

The AS/400 system supports the following telecommunication lines (all the lines do not have to be the same):

- Switched point-to-point (manual or automatic answer, manual or automatic call)
- Nonswitched point-to-point
- Nonswitched multipoint
- IBM Token-Ring Local Area Network
- X.25 network

Each ICF communications type (except intrasystem communications) requires at least one communications line to communicate with a remote system.

Refer to the *Communications User's Guide* for more information.

Operating System/400

Following is a description of the Operating System/400 (OS/400) support provided for AS/400 system communications.

Communications Configuration

Before you can use communications on the AS/400 system, you must define the environment through the communications configuration function. This support allows you to create, change, display, and delete the communications line, controller, and device descriptions.

APPC/APPN requires mode descriptions and class-of-service descriptions. The configuration support provides this function.

Refer to the *Communications User's Guide* for more information on communications configuration. APPN support provides the ability to communicate with a remote system without having to manually configure the remote system. Refer to the *APPC and APPN User's Guide* for more information.

Intersystem Communications Function File

The ICF file is used to send and receive data between two application programs, and to describe how to present that data. The ICF file contains the file description identifying the record formats used by the communications application program.

The ICF file allows you to define a single file and the program devices used by that file. An ICF file supports any combination of program devices for all the supported communications types. The application program can then write data to or receive data from any of the program devices defined in the file.

Refer to Chapter 4 for information on creating and using the ICF file.

Data Description Specifications (DDS)

DDS defines the format of the data and the characteristics of the operation used on the data. This information is specified as part of the ICF file, the display file, and the printer file.

Certain DDS functions are unique to communications. These functions are described in Chapter 6. (For general information on coding DDS, refer to the *DDS Reference*.)

System-Supplied Formats

System-supplied formats that provide functions similar to those accomplished by using DDS keywords are provided as part of the ICF support, and can be used to do specific communications functions. Refer to Chapter 7 for more information about system-supplied formats.

Control Language

With control language (CL) commands, you can create, change, display, and delete the various communications configurations. A menu interface is also provided to assist you in this function.

ICF file commands are provided that allow you to create, change, and override the file descriptions. Commands are also provided that allow you to add, change, remove or override device entries for the file. Chapter 4 describes the file commands and their use.

You cannot use CL commands to do ICF communications functions.

For more information on the CL commands for configuring communications, refer to the *Communications User's Guide*.

Security

The security provided on the AS/400 system controls who can use communications device descriptions, and the commands that are used with the device descriptions. Security on both the local and remote systems must be considered in writing and running applications.

See the *Security Concepts and Planning* for general system security information and Chapter 8 for communications-specific security considerations.

Error Handling

Major and minor return codes are provided to the application program so that error conditions can be properly handled. Applications written in C/400, COBOL/400 and RPG/400 can access the return codes to help diagnose problems. In addition, messages are entered in the job log to identify the error that occurred. COBOL/400 and RPG/400 provide language-defined file status that can be used either in place of, or in addition to, the major and minor return code. C/400 does not have file status values.

You can recover from many communications errors with little or no operator involvement. You may be able to reestablish the session or close and reopen the file to accomplish recovery within the user program. The *Communications User's Guide* describes line errors. Appendix B discusses program error recovery.

High-Level Language Support

You can use AS/400 system communications support to write application programs in the supported high-level languages.

C/400, COBOL/400, and RPG/400 support the ICF interface. Chapter 9 through Chapter 11 provide program examples written in C/400, COBOL/400, and RPG/400.

The programs presented in this manual serve as examples only. They are used to show concepts and techniques and may not represent the most efficient programming methods.

Additional Programming Support

Support is also provided in addition to the ICF interface to allow the application program to send or retrieve database file members from one system to another. This support is provided by file transfer support (FTS).

Appendix E describes this support.

Chapter 3. Introduction to Intersystem Communications Function

ICF allows program-to-program communications between the AS/400 system and other systems. It also provides program-to-device communications between the AS/400 system and hardware devices.

This chapter provides an introduction to:

- How ICF works
- Some of the terms used to describe ICF
- Configuring for and starting communications
- Defining your ICF file
- How to write a program to use ICF

Notes:

1. The two examples shown in this chapter allow two AS/400 systems to communicate with each other. One program is started by the local AS/400 system operator, which then starts the program on the remote AS/400 system.
2. Although both communications types in these examples must be advanced program-to-program communications (APPC) or binary synchronous communications equivalence link (BSCEL), the examples give you a general understanding of how to write a program that uses any communications type under ICF.
3. Not all communications types require all the operations shown in this chapter. Refer to the appropriate communications programming manual for the communications type you are using for information about a specific communications type.

In Figure 3-1, an application program on the local AS/400 system (local program) sends data to an application program on a remote AS/400 system (remote program) and then receives data.

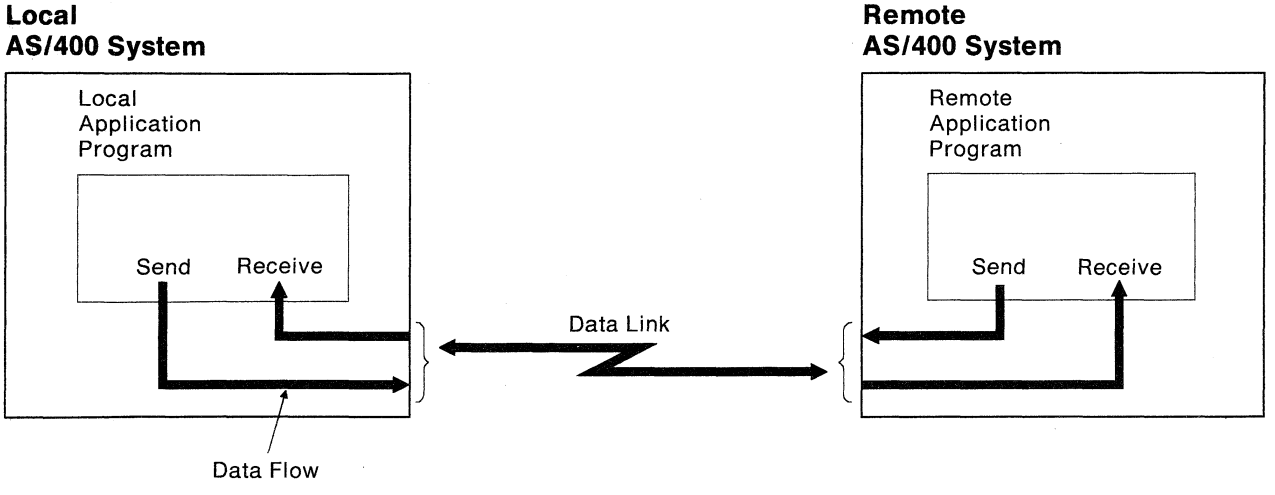


Figure 3-1. Sending Data from a Local Program to a Remote Program

Either program can send data first. You must determine which system is to send data first before you write a program, so you know which operations to do first. Use **ICF communications functions** and high-level language operations to handle communications within an application program. The ICF functions are described in Chapter 6 and Chapter 7. The language operations you use are the same operations you use when your program is not using ICF. Although these operations are summarized in Chapter 9 through Chapter 11, they are not fully described in this manual. Refer to the appropriate language reference manual for more information.

Figure 3-2 shows the major parts of ICF. The local **application program** is the program you write to allow your system to communicate with a remote system. **ICF data management** handles the communications functions and data from your program. The underlying support provided by the communications type handles the communications protocol needed to connect your AS/400 system to the remote system.

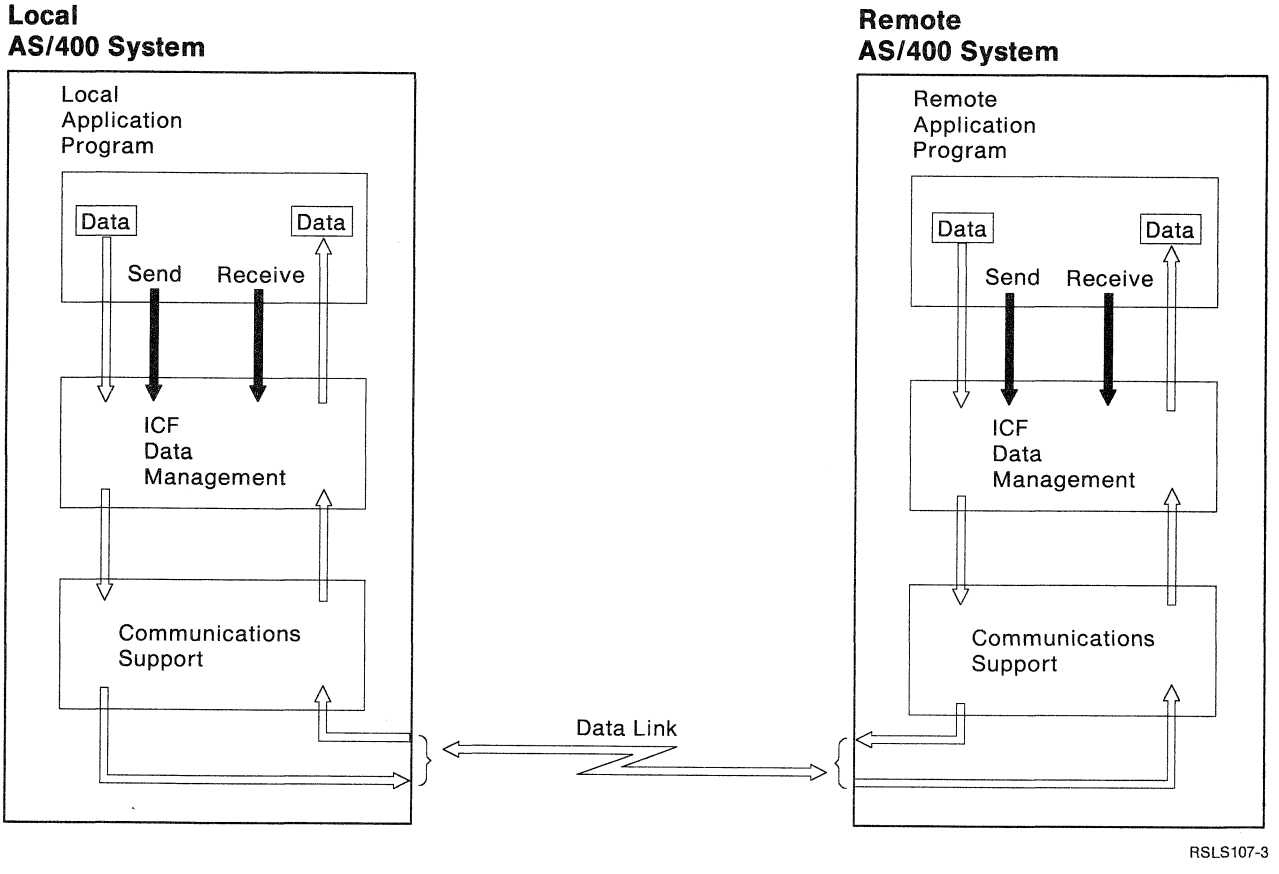


Figure 3-2. Major Parts of ICF Data Management

ICF data management supports several communications types. Use the communications type that enables you to communicate with your remote system. Refer to Chapter 2 for a list of the communications types and for an overview of the remote systems they support. See the appropriate communications programming manual for a complete description of the remote systems supported by a specific communications type.

Hardware and system-supplied programs handle sending and receiving data on the communications line. Since you do not need to know about these system programs

to write an application program using ICF, these programs and hardware are not described in this manual.

Configuring for Communications

Before communications can occur between two systems, both systems must be configured. You must configure your system to define the appropriate communications hardware and characteristics before you can use your programs. The AS/400 system communications configuration support allows you to create, change, delete, display, and print the following configurations:

- Line descriptions
- Controller descriptions
- Device descriptions
- Mode descriptions (APPC/APPN only)
- Class-of-service descriptions (APPN only)
- Configuration lists

Not all of the listed configurations are used by all the communications types.

The *Communications User's Guide* explains how to configure the system for use with communications. The APPN support provides the ability to communicate with the remote system without having to manually configure devices on the local system. Refer to the *APPC and APPN User's Guide* for more information.

Part of the connection between the application and configuration is through the remote location name which is defined as part of the device description. Refer to "The Intersystem Communications Function File" on page 3-4 for more information.

If programs on your system can be started from a remote system, you can define the distribution of work across your subsystems. The AS/400 system considers the communications device to be another source of work for a subsystem. Therefore, you must define a communications entry within the subsystem description to identify the communications devices for which work can be received by the subsystem.

Default communications entries are shipped with the system. However, you can change these entries with the Add, Change, and Remove Communications Entry (ADDCMNE, CHGCMNE, and RMVCMNE) commands. See the *Communications User's Guide* for more information on using these commands. Refer to the *Work Management Guide* for information on subsystems and communications entries.

Varying on Communications Configurations

You must vary on the particular communications configurations you want to use before running your communications applications. (The configurations must already be defined.) The Vary Configuration (VRYCFG) command is used to vary on the appropriate line, controller, and device configurations.

Note: You can specify that the configurations be automatically varied on at IPL when you create your configurations.

The VRYCFG command does the following:

- Ensures compatibility between the configuration and the communications hardware.

- Determines whether the requested data link is available.
- Establishes a physical connection with the remote system. For SNA configurations, SNA communications may be established with the remote system, depending on the line type (switched or nonswitched) and the configuration parameters you have chosen.

The VRYCFG command prepares only the local end of the link to communicate with the remote system. You must also prepare the remote system. Communication can begin when you have prepared both ends and have established a physical connection between the two. For APPC communications, a mode must be started before you establish a session. Generally, the mode starts automatically when the device is varied on or when a request to establish a session is received. You can also use the Start Mode (STRMOD) command to start a mode.

Refer to the *Communications User's Guide* for more information on the VRYCFG command. Refer to the *APPC and APPN User's Guide* and the *Communications User's Guide* for more information on starting modes.

The Intersystem Communications Function File

An ICF device file defines the layout of the data sent and received between two application programs and links you to the configuration objects that you will use to communicate with the remote system. You identify and use this file in your high-level language application.

Defining the File

The following commands are used to define the file:

- The Create Intersystem Communications Function File (CRTICFF) command is used to create the ICF file.

  Note: If you use system-supplied formats (described in Chapter 7) IBM supplies a file called QICDMF for your use and you do not need to do this step.

- The Add Intersystem Communications Function Device Entry (ADDICFDEVE) or Override Intersystem Communications Function Device Entry (OVRICFDEVE) command is used to define a program device entry. This program device entry is that part of the file that provides the connection to the configuration objects that you will use to communicate with the remote system.

Using the File

An application program uses the file as follows:

- A program communicates through a program device name. The program device name used in the application maps you to the program device entry in the ICF file that contains the same program device name.
- The program device entry also contains a remote location name. This remote location name (specified as part of the device description) provides the final step in completing the link between the application and the device description.

Refer to Chapter 4 for more information on creating the ICF file and on defining program device entries to the ICF file. Also refer to Chapter 4 for more information on the remote location name.

Figure 3-3 on page 3-5 shows the relationship between the program, the ICF file, and the communications configurations.

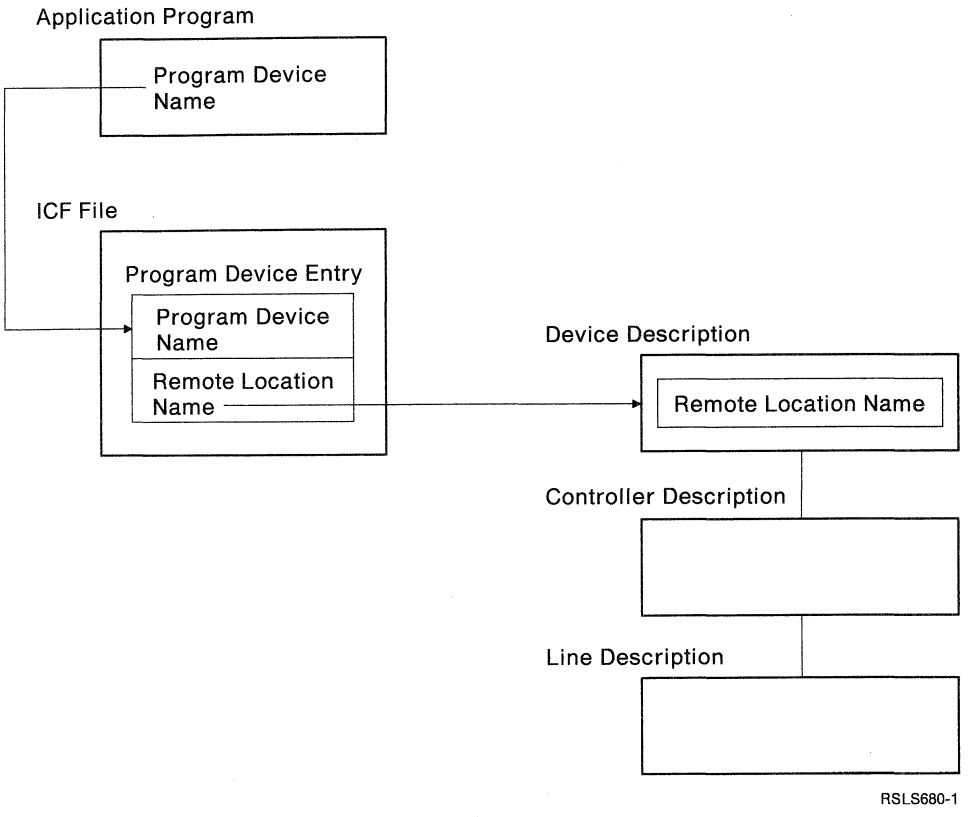


Figure 3-3. ICF File-Configuration Relationship

Not all of the communications types require that all of these configurations be explicitly created.

Starting Your Program

Your application program can be started by an operator at your system, or by a request from the remote system.

A remote system starts an application program on your local AS/400 system by sending a special record, called a program start request, to your system. Refer to "Starting a Program on the Remote System" on page 3-9 for more information about the **program start request**. Refer to the appropriate communications programming manual for the communications type you are using for a description of this special record.

Opening the Intersystem Communications Function File

Before communications can occur, your program must open an ICF file (previously created with the CRTICFF command). All communications functions are issued through the ICF file.

Starting Communications with the Remote System

Before your local program can communicate with the remote system, you must establish a **communications session**. A communications session is a logical connection between two systems through which a local program can communicate with a program at a remote location. A communications session is established with an acquire operation and is ended with a release operation or end-of-session function.

In Figure 3-4, your program establishes a session using an **acquire** operation with PGMDEVA specified as the program device name.

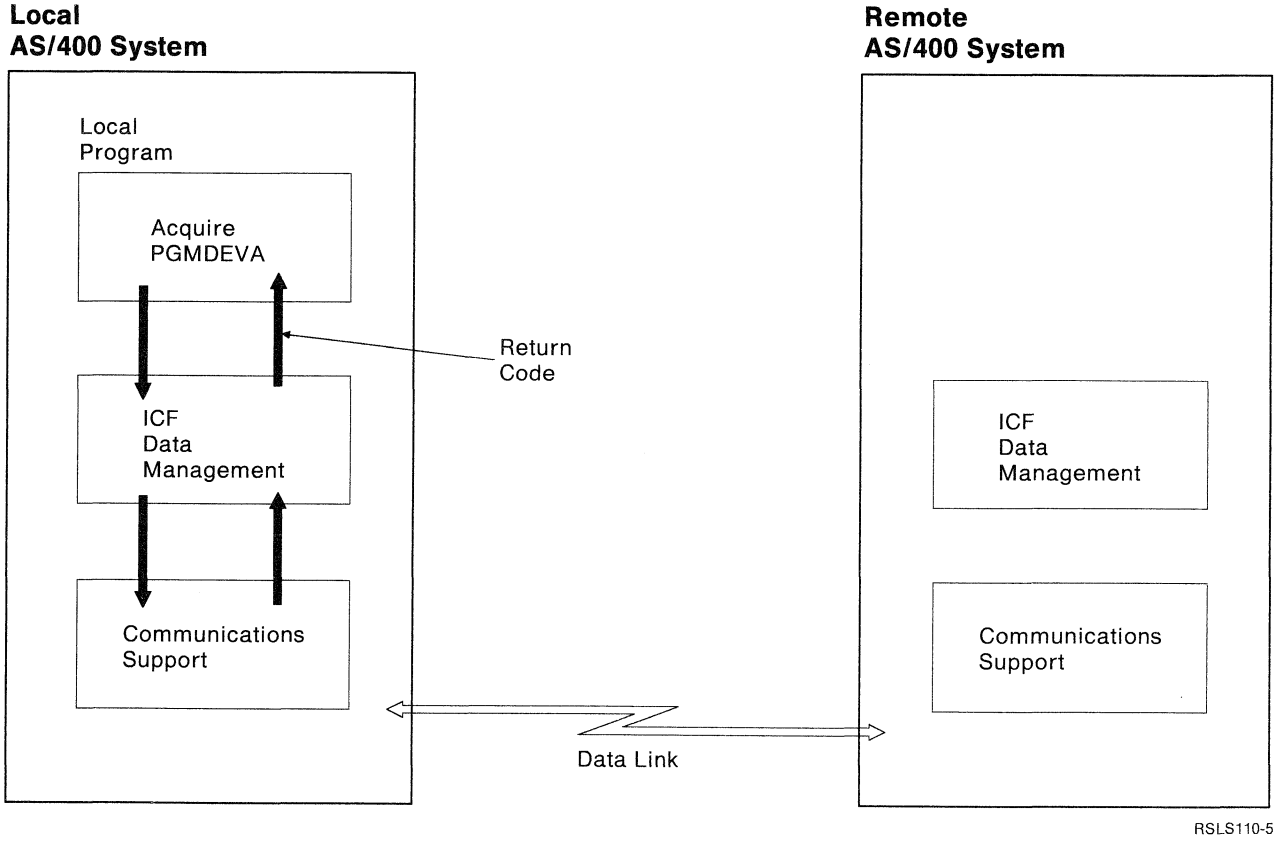


Figure 3-4. Establishing a Session

The program device name specified on an acquire operation must correspond to a program device entry in the ICF file with the same program device name. The remote location name associated with that program device entry identifies the remote system with which the session is to be established.

The program device entry is defined with the ADDICFDEVE or OVRICFDEVE command. The PGMDEV parameter specifies the program device name. The RMTLOCNAME parameter specifies the remote location name. The remote location name (also specified as part of the device description) provides the link between the program device entry and the device description.

The following is an example of how a control language program and a high-level language application program are used to acquire a program device. You can use either the ADDICFDEVE or OVRICFDEVE command. This example uses the ADDICFDEVE command.

```
YOURCL
 ADDICFDEVE FILE(ICFFILE) PGMDEV(PGMDEVA)   RMTLOCNAME(CHICAGO)
                 |               |                    |
                 |               |                    Identifies the remote
                 |               |                    location with which
                 |               |                    your program will
                 |               |                    communicate.
                 |               |
                 |              Identifies the name known
                 |              by the program (PGMDEVA).
                 |
                Identifies the ICF file
                to which the definition is added.
 CALL YOURPROG


YOURPROG

   .
   .
   .
   ACQUIRE PGMDEVA
   .
   .
   .
```

Note: You can use other parameters with the ADDICFDEVE and OVRICFDEVE commands to define attributes, such as format selection (FMTSLT), to be used during this session. See Chapter 4 for more information about the ADDICFDEVE and OVRICFDEVE commands and their parameters.

When the program issues an acquire operation, ICF data management returns a return code to your program indicating whether it can communicate (whether a session is established) with the remote system at this time. If communications cannot be established, the return code tells your program why communications failed. See Appendix B for more information about return codes.

Your program cannot send or receive data until the acquire operation succeeds. Therefore, your program must check the return code. In our example, the return code indicates that communications was started. Therefore, a communications

session exists between the local AS/400 system and the remote AS/400, as shown in Figure 3-5.

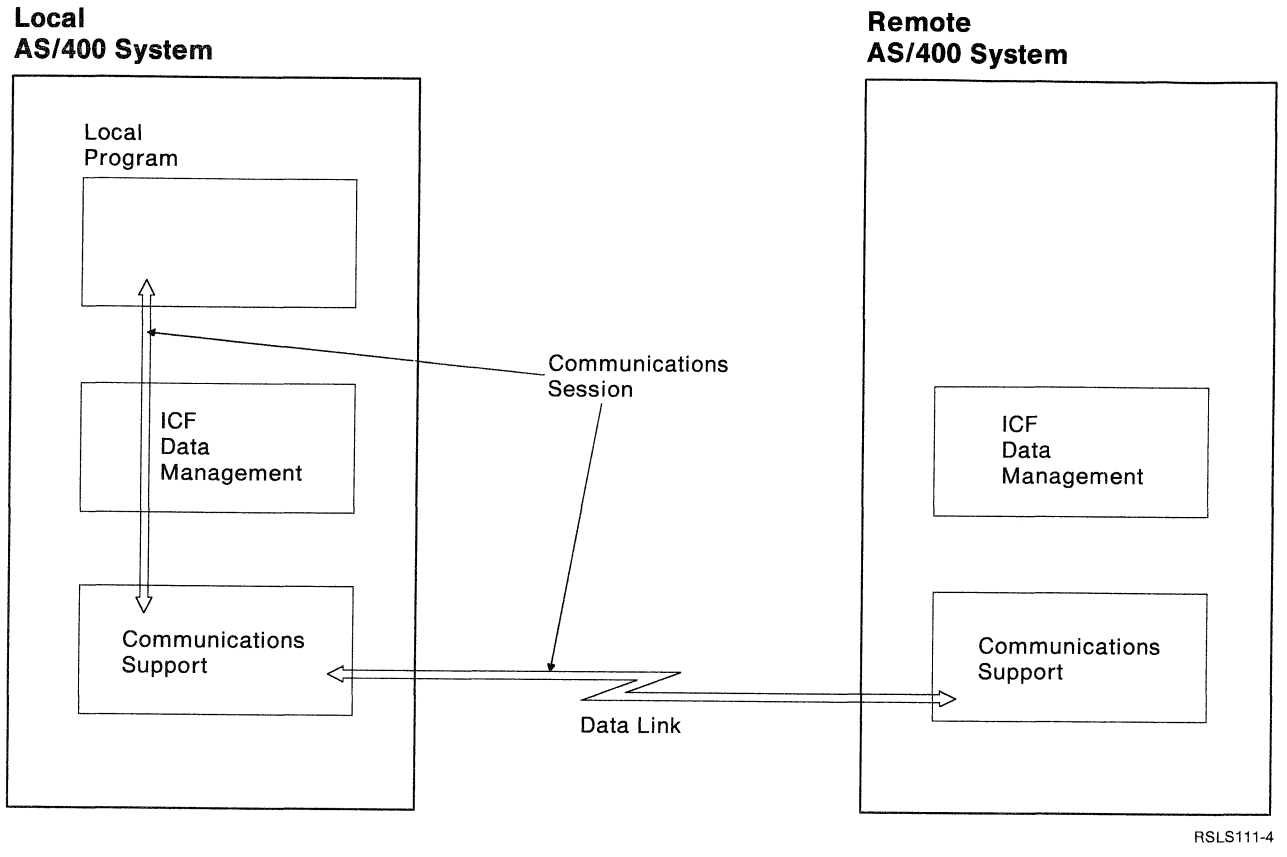


Figure 3-5. Communications Session Established

The acquire can be done automatically as a part of the open file operation by specifying the desired program device name (in this example PGMDEVA) on the ACQPGMDEV parameter of the CRTICFF command. Refer to Chapter 4 for more information.

Even though the session has been started, the application program at the remote system has not yet started. "Starting a Program on the Remote System" on page 3-9 describes how an application program is started at the remote system.

Starting a Program on the Remote System

Your program must specify and start the program at the remote system with which it will communicate. After this remote program has been started, a **communications transaction** has been started. A communications transaction is a logical connection between two programs on a session. A communications transaction is started by an **evoke** function and is ended by a detach function. After the communications transaction starts, data can be exchanged between the two programs.

Use the evoke function with the necessary parameters to send the name of the program that you want started at the remote system. These parameters include the program name (from either a high-level language program or a control language program), the remote system library where the program is stored, and security information (when required). When your program issues a write operation with the evoke function specified, a program start request is sent to the remote system.

The program that issues the evoke function is the **source program**. The program started on the remote system is the **target program**. In this example, the local program is the source program (it issued the evoke), and the remote program is the target program.

In Figure 3-6, the evoke function is used to start the program named TGTPGM at the remote system.

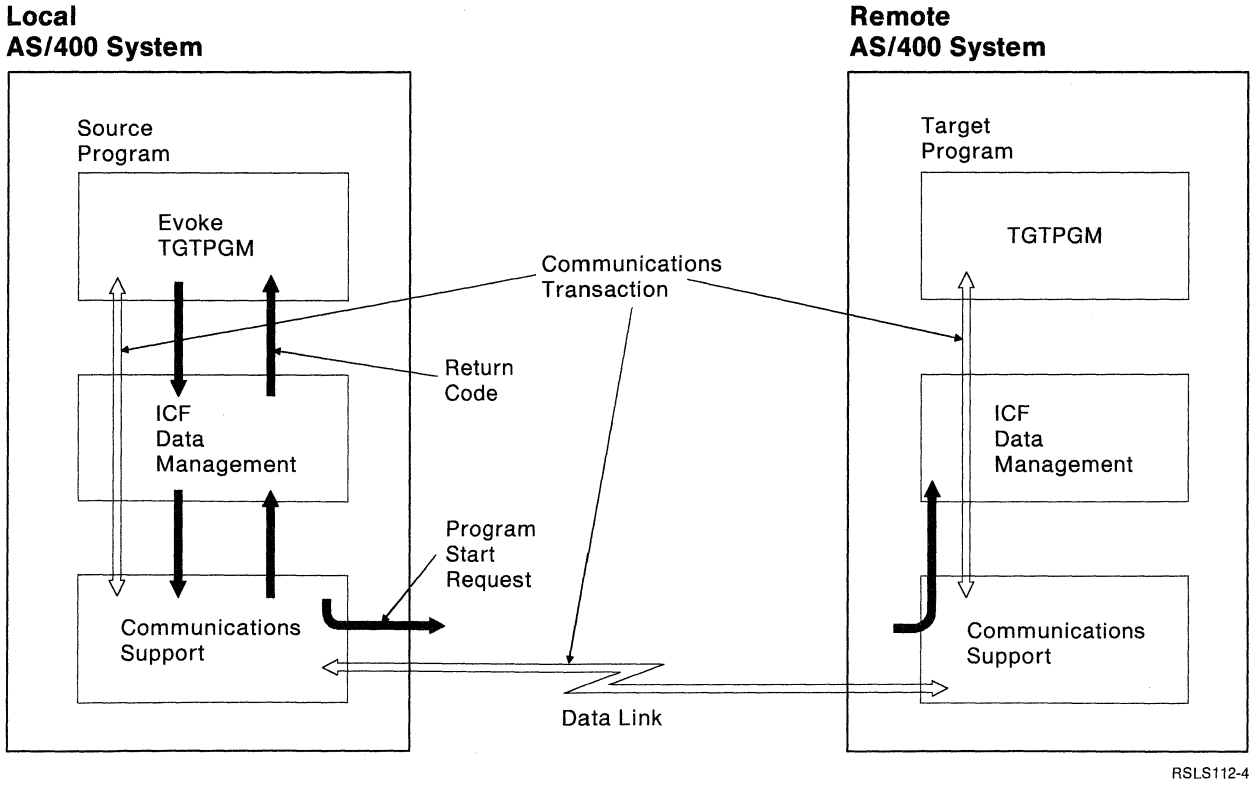


Figure 3-6. Program Started at Remote System by Evoke Function

A return code is always given to your program to indicate the status of the evoke function. In Figure 3-6, the return code tells your program that the evoke request was accepted and a program start request was sent to the remote system. If the

program start request succeeds, the remote system program and the communications transaction start.

Your program can also send program initialization parameters with the evoke function. If the remote system is an AS/400 system, the target program can access any parameters specified with the evoke as if they were parameters passed on a call command.

The type of evoke function you use depends on the communications type you use and on the type of remote system with which you communicate. For more information about the evoke functions, refer to Chapter 6, Chapter 7, and to the appropriate communications programming manual for the communications type you are using.

Connecting to the Session — Target Program

Before a target program can send or receive data, it must first be associated with the session in which the program start request was received. This association is established by opening an ICF file and acquiring a program device associated with a special remote location name of ***REQUESTER**.

A remote location name of *REQUESTER specifies that:

- The remote location used is the remote location specified in the device description that received the program start request.
- There is no specific remote location assigned to the program device by the ADDICFDEVE or the OVRICFDEVE command.

Any program device name defined in a program device entry with a remote location name of *REQUESTER is referred to as a **requesting program device**.

The target program identifies the requesting program device in the same way that the source program does. The target program specifies, on an acquire operation, the same program device name as the name specified on the PGMDEV parameter on the ADDICFDEVE or the OVRICFDEVE command.

Figure 3-7 on page 3-11 shows the relationship between the program, the ICF file, and the communications configurations for a requesting program device.

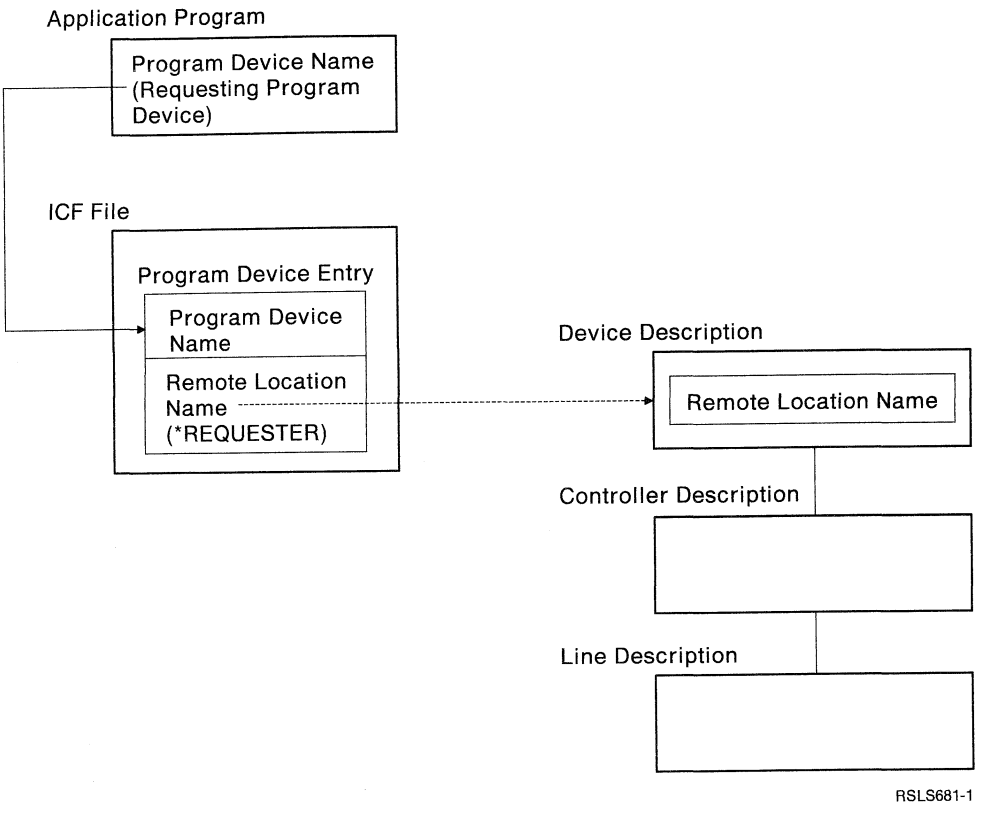


Figure 3-7. Requesting Program Device Relationship

Note: The device description that receives the program start request is the device description that is selected when the acquire operation is issued to the requesting program device.

When the target program issues an acquire operation to the requesting program device, a new session does not start. The acquire only establishes a logical connection between the target program and the session and transaction that were started by the source program.

The remote program cannot send or receive data until the acquire operation is successful.

The following shows how a control language program and high-level language application program can be used to acquire a requesting program device.

```
TGTCLPGM
 OVRICFDEVE PGMDEV(PGMDEVB)   RMTLOCNAME(*REQUESTER)
                    |                    |
                    |                    Identifies that you want to
                    |                    communicate with the device
                    |                    description that receives the
                    |                    program start request.
                    |
                   Identifies the name known
                   by the program (PGMDEVB).
 CALL TGTPGM


TGTPGM

   .
   .
   .
   ACQUIRE PGMDEVB
   .
   .
   .
```

Note: The target program started as a result of a program start request can be a high-level language program or a control language (CL) program. In this example, the CL program containing the OVRICFDEVE command and the call statement is the program that is started as a result of the program start request. The CL program calls the high-level language program.

In Figure 3-8, the target program establishes a logical connection to the session and transaction (started by the source program) by acquiring the requesting program device named PGMDEVB (as assigned by the ADDICFDEVE or OVRICFDEVE command).

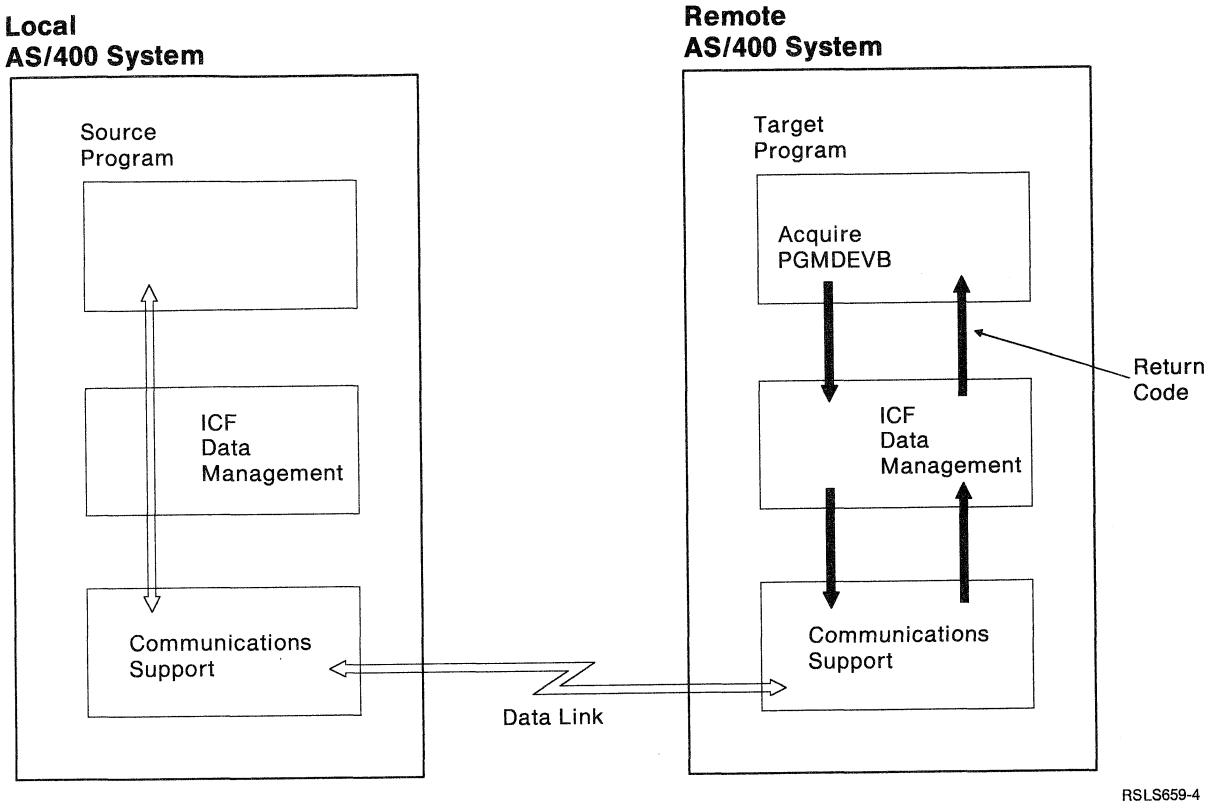


Figure 3-8. Establishing a Logical Connection between the Target Program and the Session

The acquire can be done automatically as part of the open file operation, by specifying the requesting program device name (in this example, PGMDEVB) on the ACQPGMDEV parameter of the CRTICFF command. Refer to Chapter 4 for more information.

Sending and Receiving Data

In Figure 3-9, the source program sends data first. To obtain that data, the target program must issue a receive request as the first operation following the acquire.

You use the same program device name specified on the acquire operation on each send and receive request. In the example, the source program uses the program device name PGMDEVA and the target program uses PGMDEVB.

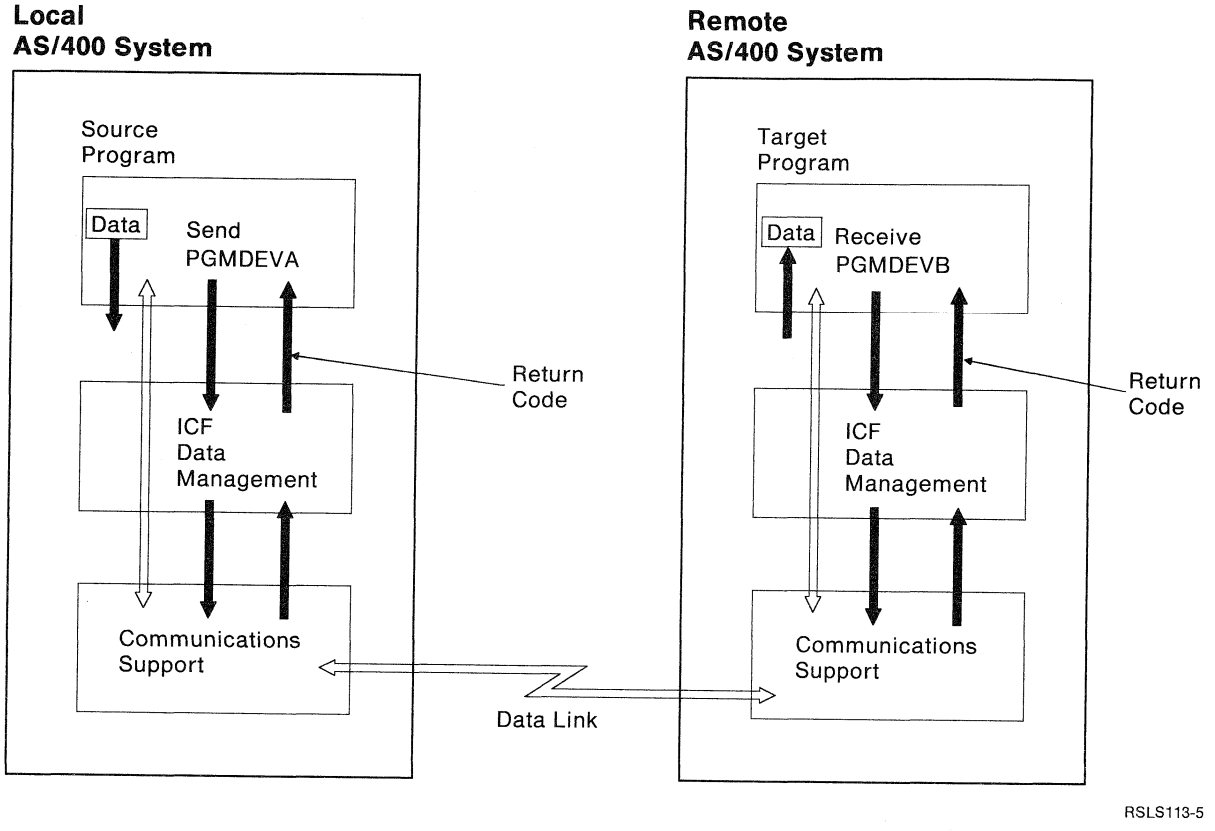


Figure 3-9. Data Sent by a Send Request

Again, the source program gets a return code indicating the status of the send request. Since the remote system in this example is an AS/400 system, the target program is also given a return code indicating the status of the receive request.

Ending Communications with the Remote System

You must end both the communications transaction and the communications session to end communications with the remote system. You can end the communications session in one of the following two ways:

- Explicitly by the program, as shown in Figure 3-10 and Figure 3-11
- Implicitly ending all sessions and transactions associated with the source program by a close of the ICF file

The transaction and session can be ended by either the source or target program.

Ending the Transaction

The sending and receiving of data continues until one of the two programs ends the communications transaction (either the source or target program can end the transaction). The **detach** function is used to tell the remote program that your program has no more data to send and has ended the communications transaction.

Figure 3-10 shows the source program issuing a detach function to end the communication transaction.

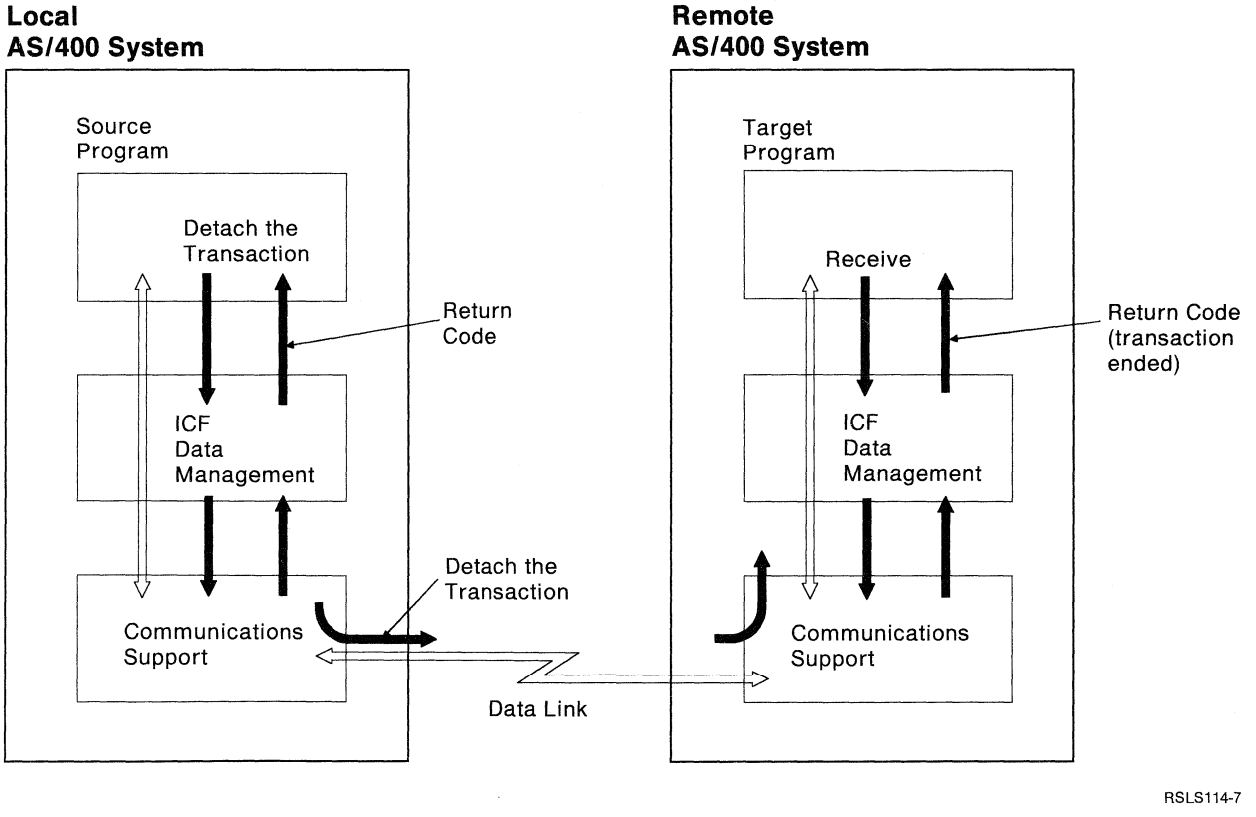


Figure 3-10. Ending a Communications Transaction: Detach Function

In this example, the target program is given a return code indicating that the transaction has ended. The target program can continue or end processing, but it can no longer communicate with the source program. However, the target program must end the logical connection to the session by ending the session.

The communications session still exists for the source program. The source program can start another program at the remote system and another transaction, or it can end the communications session and stop communicating with the remote system.

If a target program issues the detach, its logical connection to the session as well as the transaction is ended.

Ending the Session

When a session is no longer needed, it should be ended. A source program ends the session by issuing a **release** operation or **end-of-session** function. However, a target program must also sever the connection to the session by issuing a release operation or end-of-session function.

Figure 3-11 shows the source program using the release operation and the target program using an end-of-session function to end the session.

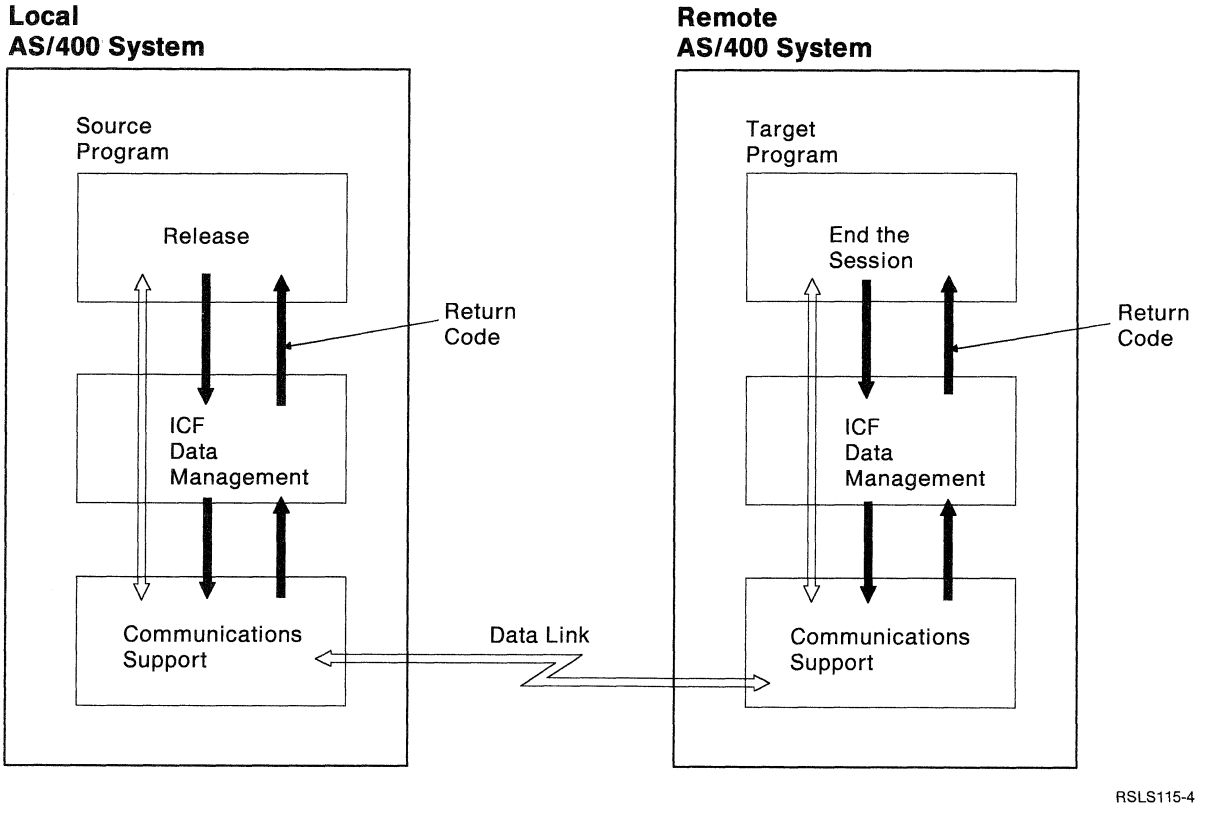


Figure 3-11. Ending a Session: Release Operation and End-of-Session Function

When the source program issues the release operation, ICF data management tries to end the session. If the communications transaction has ended, the session ends and the source program receives a return code indicating that the session has ended.

If the session cannot be ended, the source program receives a return code indicating that the release operation was not successful. (For example, the transaction may not have ended.) If your program cannot recover from the error, you can use the end-of-session function to force the session to end. The end-of-session function always ends the session.

If you issue an end-of-session, you may not be able to determine:

- If the transaction has ended normally
- If all the data has been sent or received

Note: When you use the end-of-session function, your program must make sure all data is received.

The program at the remote system may (depending on the communications type) receive a return code indicating that the session did not end normally.

After ending the transaction and session, a source program can start another session and transaction, continue local processing, or end.

A target program can continue local processing or end after ending a session.

Closing the Intersystem Communications Function File

Your program should close the ICF file when you are done processing. Closing the ICF file also implicitly ends any active transactions or sessions for the program.

Varying off Communications Configurations

When you no longer need a communications configuration, you can use the VRYCFG command to vary off the configurations you previously varied on.

If you are using an APPC device, you can end any active modes with the End Mode (ENDMOD) command before you use the VRYCFG command. If you do not use the ENDMOD command, any active modes associated with the device are ended automatically as part of the VRYCFG.

Refer to the *Communications User's Guide* for information on the VRYCFG command. Refer to the *Communications User's Guide* and the *APPC and APPN User's Guide* for information on the ENDMOD command.

Additional Information on Sessions and Transactions

The information presented in this chapter has only described the flow of two programs using a single session and transaction to communicate with each other.

The following sections describe variations of sessions and transactions.

Multiple Transactions

Figure 3-12 shows how a source program on a single session can start and end multiple transactions. Only one transaction can be active on a session at a time.

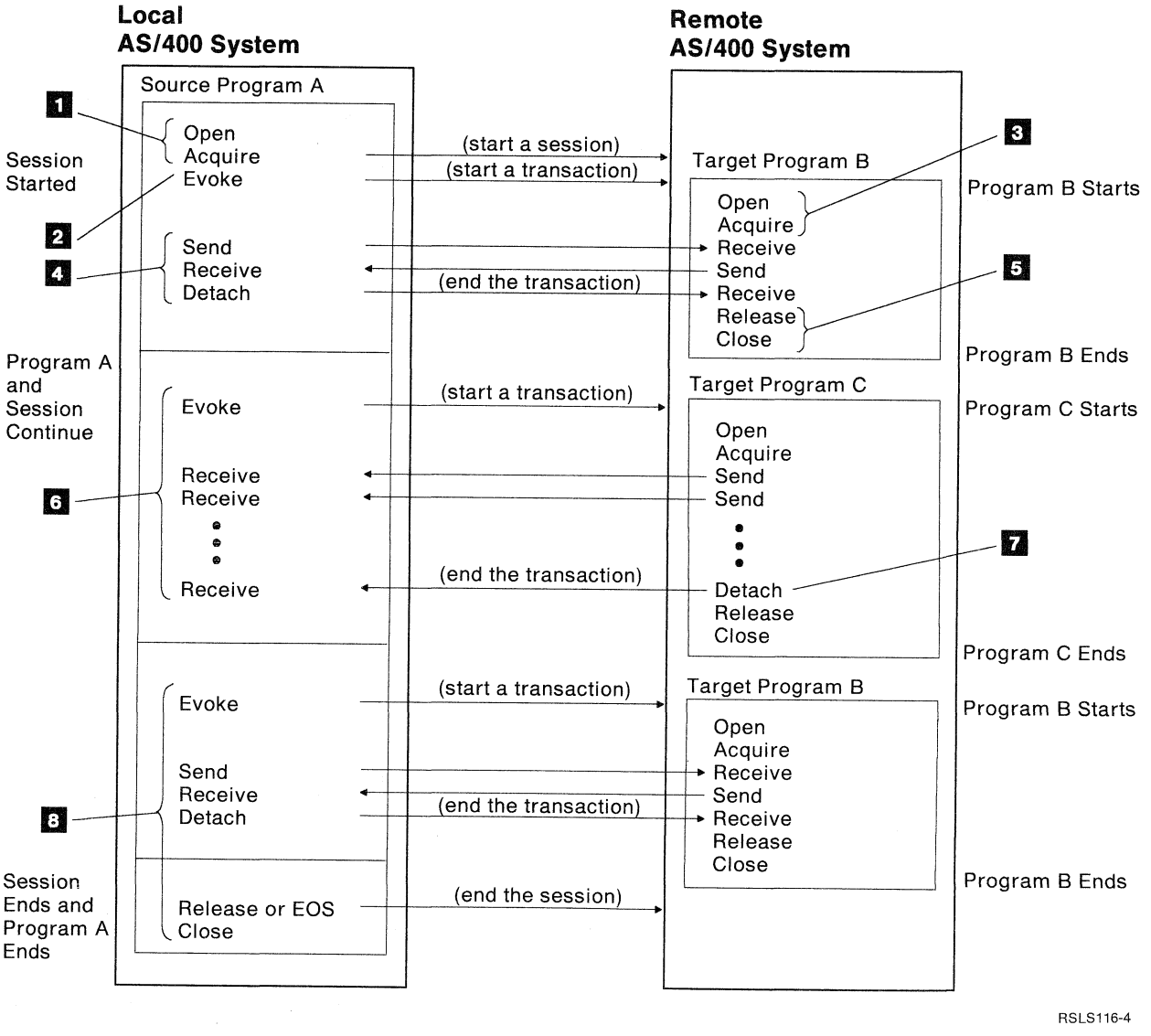


Figure 3-12. Starting and Ending Sessions and Transactions

1. Program A, on the local AS/400 system, opens the ICF file and then issues an acquire operation to start a session with the remote AS/400 system.
2. Program A issues the evoke function, which starts the communications transaction, to start Program B on the remote AS/400 system.
3. Program B must open the ICF file on the remote AS/400 system and issue an acquire operation for the requesting program device to establish a logical connection to the session and transaction.
4. Programs A and B exchange data. Program A ends the transaction. Program B can end (as shown) or continue processing. Program B cannot, however, communicate with the local AS/400 system.
5. Program B releases the session it previously acquired and closes the ICF file.
6. Program A starts a transaction with Program C on the remote AS/400 system and exchanges data.
7. Program C on the remote AS/400 system ends the transaction. (Either program can end the communications transaction.) Program C releases the session and closes the ICF file.
8. Program A starts and ends another transaction with Program B. Program A then releases the session with the remote AS/400 system, and closes the file on the local AS/400 system.

Multiple Sessions

A program can communicate over multiple sessions to the same system or different systems, and can have all the sessions at the same time. When a program is communicating over multiple sessions, it can be both a target and a source program, but it cannot be both on the same session.

A program started by a program start request is the target program for that session. However, this program can also become a source program by establishing a session with another remote system. Figure 3-13 shows how a target program can start a session and a transaction.

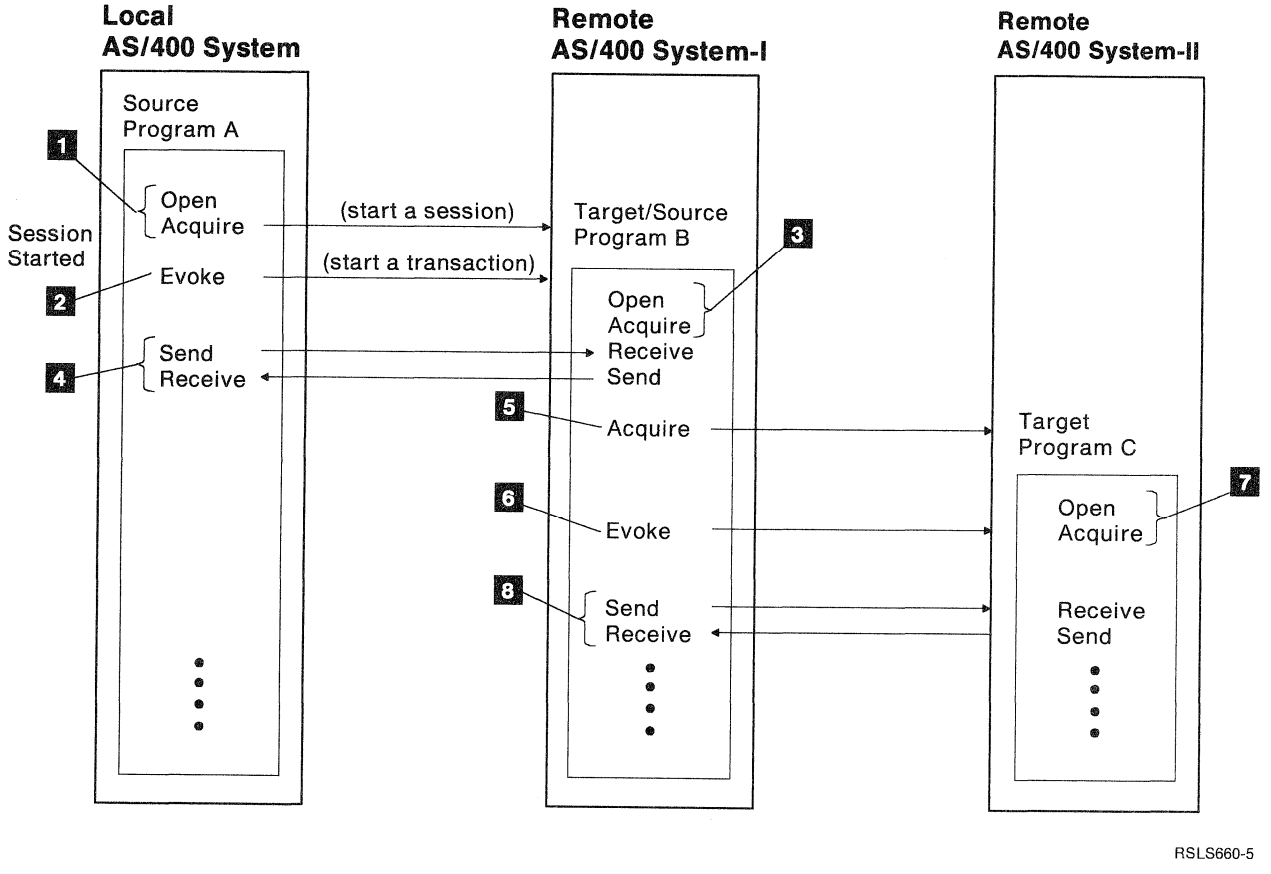


Figure 3-13. Remotely Started Program Starts a Session and Transaction

1. Program A, on the local AS/400 system, opens the ICF file and then issues an acquire operation to start a session with the remote AS/400 system.
2. Program A uses the evoke function to start Program B on the remote AS/400 system-I, which starts a communications transaction.
3. Program B must open the ICF file and issue an acquire operation for the requesting program device to establish a logical connection with the session and transaction.
4. Programs A and B can exchange data.
5. Program B issues an acquire operation to start a session with the remote AS/400 system-II.
6. Program B uses the evoke function to start Program C on the remote AS/400 system-II, which starts a communications transaction.
7. Program C must open the ICF file and issue an acquire operation for the requesting program device to establish a logical connection.
8. Programs B and C can exchange data.

Summary

The major tasks you do to use ICF for a source program and a target program are explained in the following sections.

Source Program

Figure 3-14 shows the sequence of events your AS/400 system application program follows when it starts a session with the remote system.

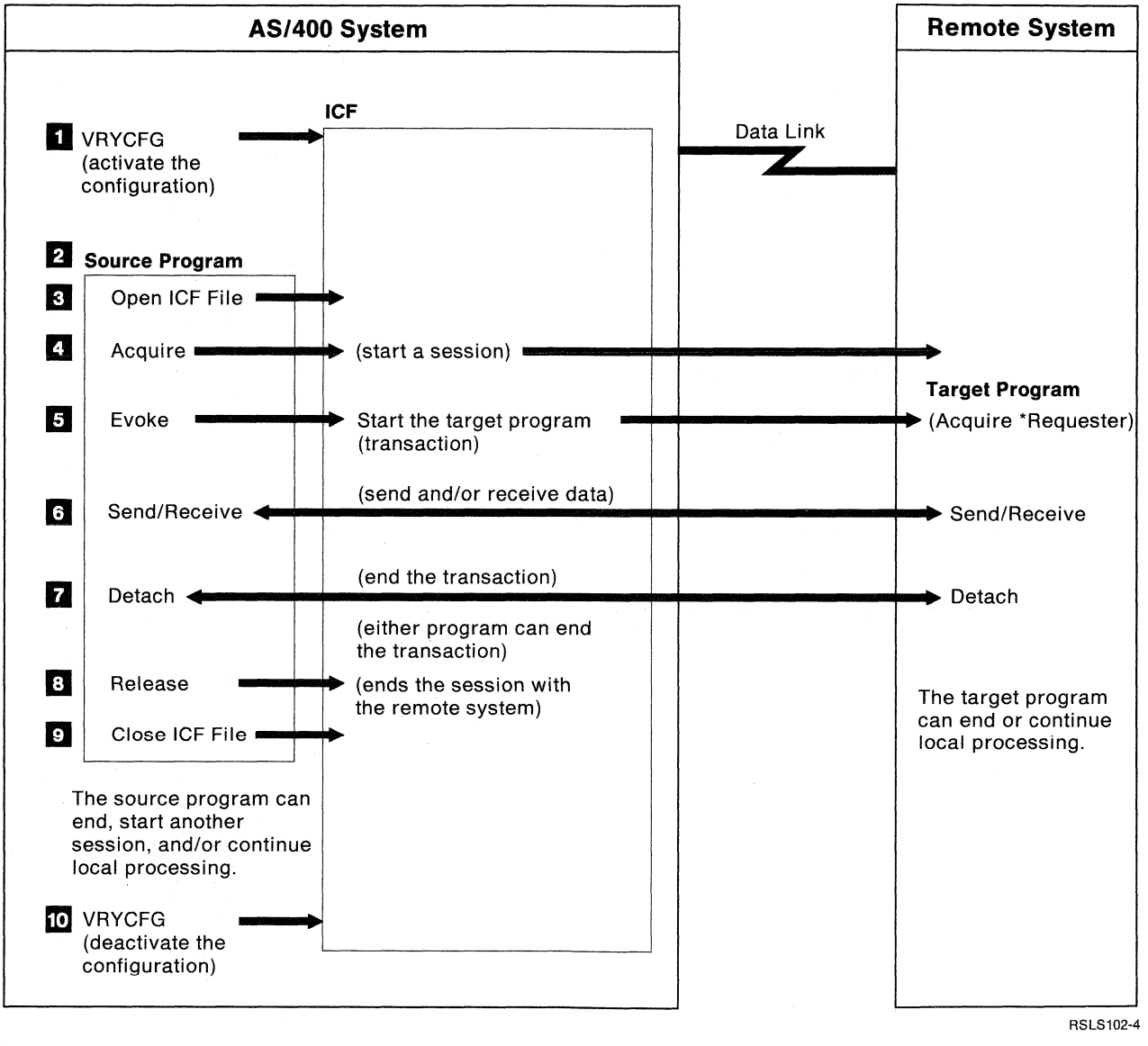


Figure 3-14. The AS/400 System Application Starts a Session with a Remote System

1. You must vary on the communications configurations before programs can use them to communicate with a remote system. Use the VRYCFG command to vary on the configurations. You can do the VRYCFG either within the application CL or interactively.

2. You must start the AS/400 system application program (source program) that communicates with the program at the remote system.

3. The application program must open an ICF file.

4. The AS/400 system program must start a session with the remote system before communications can begin. Your program starts a session when it issues an acquire operation.

   When your program starts (establishes) the session with an acquire operation, an ADDICFDEVE or OVRICFDEVE command specifies the program device name and the remote location name (identifying the remote system) associated with the session.

   Note: The acquire can be done implicitly as part of the open operation.

5. Within each session, you can start (evoke) transactions to allow your program to communicate with target programs. A transaction starts when your program uses the evoke function to start a specified target program.

6. Within each transaction, data can be sent and received between the source program and the target.

7. Either program can end the transaction when all data has been sent or received. Your program uses the detach function to end the transaction. When the remote system ends the transaction, your program receives a return code indicating that the transaction has ended. If a target program issues the detach, the logical connection to the session is ended implicitly by the detach (a release operation is not needed).

8. When all transactions have ended, your program should end the session. Your program can end the session by using the release operation or the end-of-session function.

9. Your program must close the ICF file.

10. Use the VRYCFG command to vary off the communications configurations when they are no longer needed. You can use the VRYCFG command either within the application CL or interactively.

Target Program

Figure 3-15 shows the sequence of events that occurs when the remote system starts the session by sending a program start request.

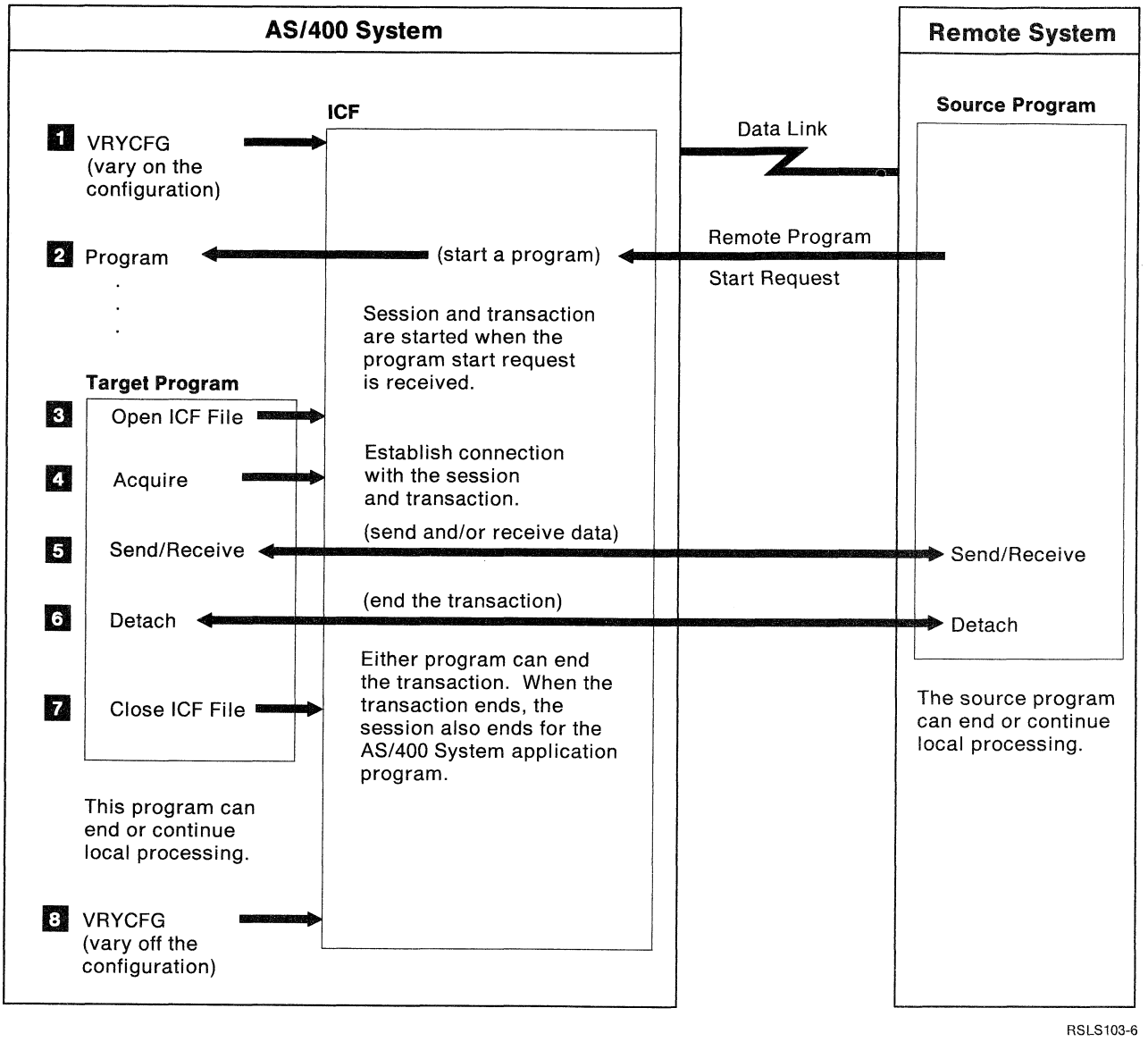


Figure 3-15. Remote System Starts a Session with a Program Start Request

1. You must vary on the communications configurations before programs can use them to communicate with a remote system. Use the VRYCFG command to vary on the configurations.

   Note: Before your system can process incoming program start requests, you must define subsystem communication entries using the ADDCMNE command. Refer to the *Communications User's Guide* for more information.

2. A program on the AS/400 system is started when your system receives the program start request from the remote system.

3. The program must open the ICF file.

4. The program must acquire the requesting program device to establish a logical connection to the session and the transaction. The program device name specified on the acquire operation must be associated with a remote location name of *REQUESTER (RMTLOCNAME(*REQUESTER), specified on either the ADDICFDEVE or the OVRICFDEVE command).

5. Your program can send or receive data, depending on the procedures previously set up with the remote system.

6. Either program can end the transaction when all data has been sent or received.

7. Your program must close the ICF file.

8. Use the VRYCFG command to vary off the communications configurations when they are no longer needed.

The previous outline summarizes the sequence of events needed for both source and target programs. Overall, every event is required, but different subsets of events can be repeated without repeating the whole series of events. For example, you can acquire and release multiple program devices in the same program. You can also run multiple programs without varying on and varying off the communications configurations.

Chapter 4. Intersystem Communications Function Files

This chapter describes the ICF files, including:

- Using ICF file commands
- Creating and changing ICF files
- Identifying the program devices used with ICF files

Chapter 5 describes how you use ICF files.

Introduction to Intersystem Communications Function Files

A device file is a description of how input data is presented to a program from a device and how output data is presented to a device from a program. A device can be a physical device or a remote system. For example:

- For asynchronous communications, a device can be an ASCII terminal.
- For advanced program-to-program communications (APPC), a device can be a logical unit on a remote system.

Device files do not contain data. Device files contain the file description identifying the device to be used and the record formats used by the application programs. The record formats and associated processing keywords are defined in the data description specifications (DDS) source.

The type of device file used for communications is the ICF file. The ICF file allows the definition of program devices for different communications types. The communications types are APPC, Systems Network Architecture upline facility (SNUF), binary synchronous communications equivalence link (BSCEL), asynchronous, intrasystem, finance, and retail communications. Your application program writes data to and reads data from the file. You can specify whether the data is to be read from a specific program device or from the first program device that responds to a request. You always write data to a specific program device.

The ICF file allows multiple sessions with different remote systems. You can define and use up to 256 program devices with an ICF file. The program devices can be a combination of different communications types. You must create the necessary communications configuration descriptions for the program devices defined to the file.

Multiple programs (in the same job or separate jobs) can use the same ICF file simultaneously. Each program can have 256 program devices per file.

The file description information for an ICF file is derived from the parameters on the Create Intersystem Communications Function File (CRTICFF) command or the Change Intersystem Communications Function File (CHGICFF) command. The record format information is derived from the DDS that define each record format in the device file and from the fields within each format.

You must also use either the Add Intersystem Communications Function Device Entry (ADDICFDEVE) or the Override Intersystem Communications Function Device Entry (OVRICFDEVE) command to specify the program devices used with the file. These commands provide the connection between the program device name and the remote location name.

The ICF file has attributes unique to ICF and attributes common to other types of device files. Figure 4-1 provides an overview of ICF files.

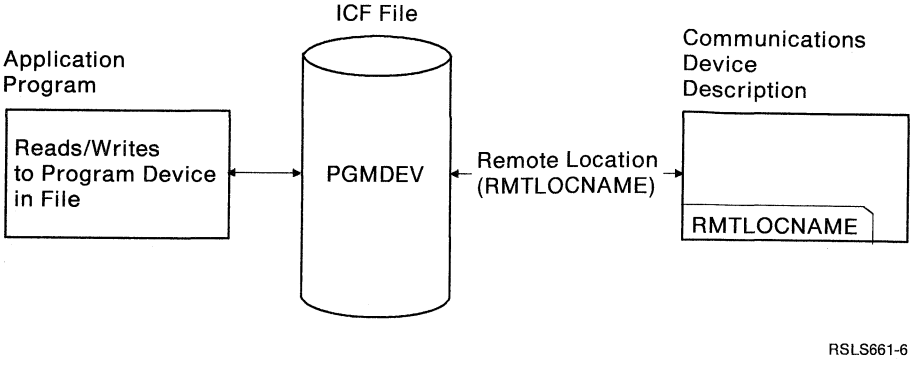


Figure 4-1. ICF File Overview

Notes:

1. The ICF file is created by using the CRTICFF command.
2. The RMTLOCNAME parameter on the ADDICFDEVE command associates a remote location name to a program device. The remote location name is used to select the appropriate device description.
3. Not all communications types require an explicitly-created device description. For more information, see the appropriate communications programming manual for the communications type you are using.

If you use system-supplied formats (described in Chapter 7) ICF supplies a file for your use. This file is QICDMF in QSYS. If you use this file in your program, you do not need to define DDS or create a file. However, you do need to define the program device entry with the OVRICFDEVE command.

DDS and system-supplied formats provide parallel functions for ICF. System-supplied formats provide a majority of the function without the need to code DDS or to create an ICF file. DDS provides the following additional functions:

- Externally described data
- Additional processing (for example, CONFIRM processing for APPC)
- Indicators to determine session state information
- More flexibility — DDS keywords can be used together in multiple combinations

Intersystem Communications Function File Commands

Three types of commands apply to ICF files: file-level attribute commands, program device entry commands, and commands for displaying information.

File-Level Attribute Commands

The file-level attribute commands are:

Create Intersystem Communications Function File (CRTICFF)
This command creates an ICF file that can be used with communications devices. After the command runs, the file contains the file attributes and the record format definitions.

Change Intersystem Communications Function File (CHGICFF)
This command changes the file attributes of an ICF file.

Override with Intersystem Communications Function File (OVRICFF)
This command can (1) override (replace) the file named in the program, (2) override certain parameters of a file used by the program, or (3) override the file named in the program and override certain parameters of the file to process.

Delete Override (DLTOVR)
This command deletes the effect of the OVRICFF command.

Delete File (DLTF)
This commands deletes the file from the system and frees the storage space allocated to that file.

Program Device Entry Commands

The program device entry commands are:

Add Intersystem Communications Function Device Entry (ADDICFDEVE)
This command adds a program device entry with the specified device name and attributes to the file. You can use this command multiple times to add multiple program device entries to the same file.

Change Intersystem Communications Function Device Entry (CHGICFDEVE)
This command changes the program device entry that was defined with the ADDICFDEVE command.

Override Intersystem Communications Function Device Entry (OVRICFDEVE)
This command is used either (1) to override attributes specified in the ADDICFDEVE command or (2) to temporarily associate the specified program device name and attributes to the file. This command is different from the ADDICFDEVE command because it does not permanently change the ICF file. The association between the program device entry and the file is only for the job in which the command runs. You can use this command multiple times to override multiple program device entries to the file.

Delete Override Device Entry (DLTOVRDEVE)
This command deletes the effect of the OVRICFDEVE command.

Remove Intersystem Communications Function Device Entry (RMVICFDEVE)
This command removes one or more program device entries from the file.

Display Information Commands

The commands used to display information are:

Display File Description (DSPFD)
This command displays information about the attributes of a device file.

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

Display Override (DSPOVR)
This command displays file overrides at any active call level for a job.

Creating an Intersystem Communications Function File

Use the CRTICFF command to create an ICF file. The ICF file contains a file description made up of information specified in two places:

- The source file containing the DDS
- The CRTICFF command

Figure 4-2 shows ICF file creation.

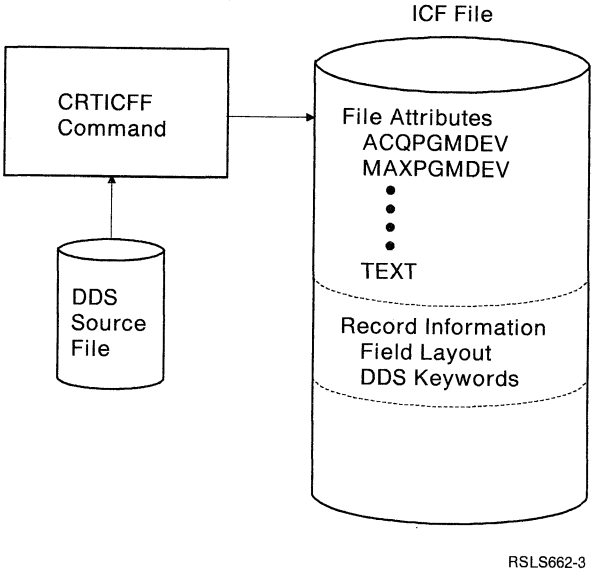


Figure 4-2. Creating an ICF File

Defining the Record Formats for an Intersystem Communications Function File

DDS provides two functions. The first function is to describe the data format as used by the program, by defining record formats and the fields within the records. The second function is to define the characteristics of the operation to be done on the record by the use of DDS keywords. DDS is supplied in the source file specified on the SRCFILE parameter of the CRTICFF file command.

Refer to the *DDS Reference* for information on using DDS to define record formats and fields. Chapter 6 describes the function of DDS keywords unique to communications. Information on coding the DDS keywords is in the *DDS Reference*.

File Attributes

Table 4-1 identifies the attributes used with an ICF file. These attributes are specified by the parameters on the CRTICFF command.

Table 4-1. ICF File Attributes

Parameter	Description
FILE	Name of the file
SRCFILE	Name of the source file containing the DDS
SRCMBR	Name of the member within the source file containing the DDS
OPTION	Output listing options
GENLVL	Severity level of DDS messages that cause the file create to fail
ACQPGMDEV	Program device to acquire when the file is opened
MAXPGMDEV	Maximum number of program devices the program can acquire using this file (This parameter also restricts the number of device entries that can be added with the ADDICFDEVE command.)
MAXRCDLEN	Maximum record length used with the file
WAITFILE	Length of time to wait for file resources to become available
WAITRCD	Length of time to wait for a record to be returned when performing a read-from-invited-program-devices operation
SHARE	Specifies whether the open data path for the file is shared with other opens of the same file in the routing step
LVLCHK	Record format level indicators check
AUT	Default authority granted to the public
TEXT	Descriptive text describing the file

Acquiring a Program Device when the File Is Opened

Use the acquire program device (ACQPGMDEV) parameter on the CRTICFF command to specify the program device you want to acquire when the file opens. The values for the ACQPGMDEV parameter are described below.

***NONE**: Specifies that no program devices are acquired when the file opens. This value is the default for the ACQPGMDEV parameter.

When you specify *NONE, the program can open the file without consideration of whether the devices to be used are available. In addition, the program does not need a routine to handle errors that occur if the program device cannot be acquired when the file is opened.

The program must acquire at least one program device to the file before doing any input/output (I/O) operations.

Program Device Name: Specifies the name of a program device to be acquired to the file when the file is opened. You can specify the name of any program device associated with the file using the ADDICFDEVE or OVRICFDEVE commands. See "Identifying the Devices Used with an Intersystem Communications Function File" on page 4-11 for more information on how to define a program device to an ICF file. The specified program device must be associated with the file before the file is opened.

When you specify a program device name for the ACQPGMDEV parameter, space is reserved in the file for the specified program device. Refer to the next section for information about reserving space in the file for program devices.

Determining the Maximum Number of Program Devices

The maximum program device (MAXPGMDEV) parameter on the CRTICFF command specifies the maximum number of program devices you want to use in the file. Use the ADDICFDEVE or the OVRICFDEVE command to associate program devices to the file. Following are guidelines for specifying the value for the MAXPGMDEV parameter:

- An ICF file can either be a single- or a multiple-device file. If your program uses the file as a single-device file, specify a value of 1 on the MAXPGMDEV parameter. If your program uses the file as a multiple-device file, specify the number of program devices simultaneously active to the file. If your file is a single-device file, only one session can be active in the program and the use of the read-from-invited-program-devices operation is restricted. If your file is a multiple-device file, more than one session can be active simultaneously and the read-from-invited-program-devices operation is allowed. Refer to the appropriate language reference manual to learn:
  - How to indicate that the program should use the file as a single- or multiple-device file
  - The differences between single- and multiple-device files
- The value specified on the MAXPGMDEV parameter restricts the number of program device entries you can add to the file using the ADDICFDEVE command.
- The value specified for the MAXPGMDEV parameter indicates the maximum number of program devices you want to have simultaneously active for this file. If you specify a program device name on the ACQPGMDEV parameter, space is reserved in the file for the program device to be acquired when the file opens, and this device must be counted when determining the MAXPGMDEV value. You must, however, still define a program device entry to the file for this program device.

  For example, if you specify a program device name of PGMDEVA on the ACQPGMDEV parameter and a 1 on the MAXPGMDEV parameter, the only device that can be added to the file with the ADDICFDEVE command is PGMDEVA. PGMDEVA is also the only device that can be used with the file. If you specify a 2 for the MAXPGMDEV parameter, you can add and use PGMDEVA and one additional device with the file.
- The value you specify on the MAXPGMDEV parameter should be no larger than necessary. If you specify a larger number of program devices than your program requires, the program uses unnecessary system resources. If the requirements for the maximum number of devices change, you can use the CHGICFF command to change the MAXPGMDEV parameter. Refer to "Changing an Intersystem Communications Function File" on page 4-9 for more information.
- The number of devices a program can handle (while maintaining a reasonable response time) is determined by the amount of processing the program does for each program device.

Determining the Maximum Record Length

Use the maximum record length (MAXRCDLEN) parameter on the CRTICFF command to specify the maximum record length you want to use with the file. This length is used in calculating the size of allocated I/O buffers and this determines the largest I/O operation that can be performed against the file. The following are guidelines for specifying the value for the MAXRCDLEN parameter:

- If your program uses externally described data and you do not vary the defined length of record formats, use the default value of *CALC. The system then generates the maximum record length based on the largest record defined in the file.
- If you use system-supplied formats in combination with an externally described file, this parameter defines the maximum length you can specify on the system-supplied formats. Since this parameter determines the allocation of I/O buffers and the system rejects output requests that are longer than the allocated I/O buffers, this parameter is important if you try to use a system-supplied format to write a record larger than the largest record in the file. Refer to Chapter 7 for more information about system-supplied formats.
- The value specified on the MAXRCDLEN parameter should be no larger than necessary. The value specified can be smaller than the largest record in the file. This can be used to minimize the size of I/O buffers allocated for a program that is using a common file that contains a larger record length than is used by the program.

Determining the Wait-for-File Resources Value

Use the wait file (WAITFILE) parameter on the CRTICFF command to specify the maximum amount of time the open and acquire operations wait before a file resource (such as a device description) becomes available for use by the file. When a file resource is available to an ICF file, the resource is allocated to the job using that file, and is not available to other jobs.

Some communications types also use the WAITFILE parameter to determine the amount of time to wait for remote communications session resources to become available.

The following are guidelines for specifying the value for the WAITFILE parameter:

- If a session is not available for allocation to the job in which your program is running, the system waits until the session is available or until the specified amount of time elapses.
- If you specify an extremely large value, your program waits a long time before it is notified that the session cannot be established.
- The wait time needs to be increased if your program fails at open or acquire time while trying to acquire a program device to a session that appears to be available.

Determining the Wait-for-Record Value

Use the wait record (WAITRCD) parameter on the CRTICFF command to specify the maximum number of seconds that the read-from-invited-program-devices operation waits for a response from the invited program devices. Although the normal response is from an invited program device, the read-from-invited-program-devices operation may also complete with a job-being-canceled (controlled) indication.

The value specified for the WAITRCD parameter has no effect on input operations directed to a specific program device. Instead, a read operation to a specific program device waits until a response is available from that program device.

Refer to Chapter 5 for more information on the read-from-invited-program-devices and read operations.

If your program does not use the read-from-invited-program-devices operation, you need not be concerned about the value specified on this parameter.

The following are guidelines for selecting the WAITRCD parameter value if your program uses the read-from-invited-program-devices operation:

***NOMAX**: Indicates that the read-from-invited-program-devices operation should wait until a response is available from an invited program device. This is the default.

When *NOMAX is specified on the WAITRCD parameter, the read-from-invited-program-devices operation does not return control to the program unless a response is available from an invited program device or the job is ending in a controlled way. If the invited program devices are unable to return a response, the program waits until the job is ended.

***IMMED**: Indicates that the read-from-invited-program-devices operation should not wait for a response from an invited program device.

Specifying *IMMED allows the program to receive a response (if available) from an invited program device. If no response is available, the program receives a 0310 return code without waiting for a time limit to end.

Number of Seconds: Specifies the number of seconds that the read-from-invited-program-devices operation waits for a response from an invited program device. If no response is received from the invited program devices within the specified amount of time, the program is informed through a major/minor return code.

Specifying the number of seconds allows the program to receive a response from an invited program device if a response is available within the specified amount of time. If no response is available within the specified amount of time, the program receives a 0310 return code, indicating that the time limit has ended.

Determining Other CRTICFF Command Parameter Values

Refer to the CRTICFF command in the *CL Reference* to determine the appropriate values for the SRCFILE, SRCMBR, OPTION, GENLVL, LVLCHK, SHARE, AUT, and TEXT parameters.

Changing an Intersystem Communications Function File

Use the CHGICFF command to change the file-level attributes of an ICF file. The changes made to the file are system-wide and affect all programs that open the file after the CHGICFF has been done. Any programs that already have opened the file are not affected during the current run. Use the parameters in Table 4-2 for changing file-level attributes values specified on the CRTICFF command.

Table 4-2. File Attributes for Changing an ICF File

Parameter	Description
ACQPGMDEV	Program device to be acquired when the file is opened
MAXPGMDEV	Maximum number of program devices that can be acquired by the program using this file; this parameter also restricts the number of device entries that can be added with the ADDICFDEVE command.
MAXRCDLEN	Maximum record length used with the file
WAITFILE	Length of time to wait for file resources to become available
WAITRCD	Length of time to wait for a record to be returned when performing a read-from-invited-program-devices operation
SHARE	Specifies whether the open data path for the file is shared with other opens of the same file in the routing step
LVLCHK	Record format level indicators check
TEXT	Descriptive text for describing the file

Overriding an Intersystem Communications Function File

Use the OVRICFF command to temporarily override the file named in the program, the file-level attributes of the file, or both. The OVRICFF command affects only the job in which it is run. Use the parameters in Table 4-3 for overriding file-level attribute values specified on either the CRTICFF or CHGICFF command.

Table 4-3. File Attributes for Overriding an ICF File

Parameter	Description
FILE	Name of file to override (same file name as application)
TOFILE	Name of file
ACQPGMDEV	Program device to be acquired when the file is opened
MAXRCDLEN	Maximum record length used with the file
WAITFILE	Length of time to wait for file resources to become available
WAITRCD	Length of time to wait for a record to be returned when performing a read-from-invited-program-devices operation
SHARE	Specifies whether the open data path for the file is shared with other opens of the same file in the routing step
LVLCHK	Record format level indicators check
TEXT	Descriptive text for describing the file
SECURE	Specifies whether this file is secure from previously called override commands

The scope of the override within the job is determined by its call level. An override command entered at one call level affects all lower call levels. The effects of the override command are deleted when you exit the call level in which the command is entered.

There are two common ways of using the override processing. One way is to do all your override processing for the job at the main call level, as shown in the following example:

```
MAINCLPGM
   OVRICFF  FILE(ICFFILE) TOFILE(ICFFILE) WAITRCD(2)
   CALL CPGM
   CALL CBLPGM
   CALL RPGPGM
```

In the example, the OVRICFF applies to all the other programs, because MAINCLPGM is at a higher call level than any of the subsequent programs. The WAITRCD value of 2 seconds is in effect for the programs CPGM, CBLPGM, and RPGPGM.

The second approach is to do your override processing at the lowest possible call level, as shown in the following example:

```
MAINCLPGM
   CALL CCLPGM
   CALL CBLCLPGM
   CALL RPGCLPGM

CCLPGM
   OVRICFF  FILE(ICFFILE) TOFILE(ICFFILE) WAITRCD(2)
   CALL CPGM

CBLCLPGM
   OVRICFF  FILE(ICFFILE) TOFILE(ICFFILE) WAITRCD(4)
   CALL CBLPGM

RPGCLPGM
   OVRICFF  FILE(ICFFILE) TOFILE(ICFFILE) WAITRCD(4)
   CALL RPGPGM
```

When CPGM is called, the WAITRCD value in effect is 2 seconds. The effects of the OVRICFF in CCLPGM are deleted when CCLPGM exits to the MAINCLPGM program. The WAITRCD specified on the CRTICFF is now in effect again. When the CBLPGM or RPGPGM is called, the WAITRCD value in effect is 4 seconds.

Refer to the *Data Management Guide* for the rules governing the use of override commands.

Use the DLTOVR command to delete the effect of the OVRICFF command.

You can use the DSPOVR command to display the file override in effect.

Identifying the Devices Used with an Intersystem Communications Function File

A program communicates through a program device in an ICF file. A program device entry has two functions:

- Associates a program device name with a remote location name
- Establishes a set of program communications-type-dependent attributes

The program device name need not be the same as the name of the device description. To establish an association between the name used in the program (program device name) and the communications configurations, you must define a program device entry to the file.

You must define one program device entry for each name used in the program (even if the name of the configuration is the same as the name used in the program). If the program is written to handle the requesting program device, you must define a program device entry for the requester by specifying a special value of *REQUESTER for the RMTLOCNAME parameter on the ADDICFDEVE or the OVRICFDEVE command.

You must define the program device entry before the program device can be acquired. If you specified a program device on the ACQPGMDEV parameter of the CRTICFF command, you must define the appropriate program device before the file can be opened.

Defining Program Device Entries

You can define a program device entry in numerous ways. For example, the ADDICFDEVE command permanently adds the program device entry to the file, while the OVRICFDEVE command provides the same function without changing the file. Figure 4-3 shows how to define a program device entry to an ICF file.

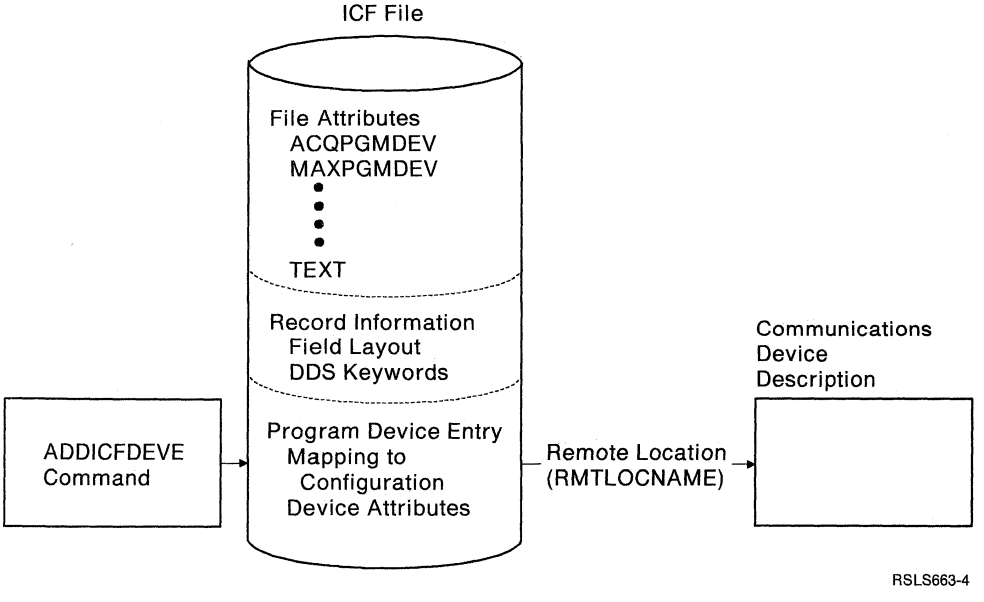


Figure 4-3. Defining a Program Device Entry to an ICF File

Note: Figure 4-3 shows only one program device entry. Multiple program device entries can be defined. The maximum number of entries is determined by the MAXPGMDEV parameter specified at file creation.

You can define a program device entry to an ICF file at either the file or job levels. A definition at the file level is system-wide and affects all users of the file. A file-level definition adds the program device entry to the specified file. A definition at the job level does not change an ICF file. The definition is only associated with the job for which the command is entered. Because changes made at the job level are not directed to a specific ICF file, all ICF files associated with the job are affected.

File-Level Definition

Use the ADDICFDEVE command to add a program device entry to an ICF file. The program device entry is added to the file specified in the FILE parameter.

Use the CHGICFDEVE command to change a program device entry previously added to an ICF file with the ADDICFDEVE command. The PGMDEV parameter identifies the entry to change. You can use the CHGICFDEVE command to change the association to the communications configurations, the program communications-type-dependent attributes, or both.

Use the RMVICFDEVE command to remove a program device entry previously added to an ICF file with the ADDICFDEVE command. The PGMDEV parameter identifies the entry to remove.

Job-Level Definition

Use the OVRICFDEVE command to override a program device entry defined with an ADDICFDEVE command or to temporarily define a program device entry. You do not have to precede an OVRICFDEVE command with an ADDICFDEVE command.

The OVRICFDEVE command follows the same override processing rules as the OVRICFF command. The following example demonstrates the use of the OVRICFDEVE command at the main call level:

```
MAINCLPGM
   OVRICFDEVE  PGMDEV(PGMDEVA) RMTLOCNAME(CHICAGO)
   CALL CPGM
   CALL CBLPGM
   CALL RPGPGM
```

In the example, the OVRICFDEVE applies to the three called programs, because MAINCLPGM is at a higher call level than any of the subsequent programs. Each of the programs, CPGM, CBLPGM, and RPGPGM, can use PGMDEVA without coding an additional OVRICFDEVE command.

In the following example, the OVRICFDEVE command is used in the lowest call level:

```
MAINCLPGM
   CALL CCLPGM
   CALL CBLCLPGM
   CALL RPGCLPGM

CCLPGM
   OVRICFDEVE  PGMDEV(PGMDEVA) RMTLOCNAME(CHICAGO)
   CALL CPGM

CBLCLPGM
   OVRICFDEVE  PGMDEV(PGMDEVA) RMTLOCNAME(NEWYORK)
   CALL CBLPGM

RPGCLPGM
   OVRICFDEVE  PGMDEV(PGMDEVA) RMTLOCNAME(NEWYORK)
   CALL RPGPGM
```

When CPGM is called, PGMDEVA is associated with RMTLOCNAME(CHICAGO). The effects of the OVRICFDEVE command in CCLPGM are deleted when CCLPGM exits to the MAINCLPGM program. When CBLPGM or RPGPGM are called, PGMDEVA is associated with RMTLOCNAME(NEWYORK).

Use the delete override device entry (DLTOVRDEVE) command to delete the effect of the OVRICFDEVE command.

Refer to the *Data Management Guide* for the rules governing the use of override commands.

Mapping Program Device Name to Communications Configurations

The first purpose of the program device entry is to associate a program device name with a device description. This mapping uses the parameters shown in Table 4-4 on the ADDICFDEVE, CHGICFDEVE, and OVRICFDEVE commands.

Table 4-4. Mapping Parameters for All Communications Types

Parameter	Description	APPC	SNUF	BSCEL	Asyn-chronous	Intra-system	Finance	Retail
PGMDEV	Program device name	X	X	X	X	X	X	X
RMTLOCNAME	Remote location name	X	X	X	X	X	X	X
DEV	Communica-tions device description	X	X			X	X	X
LCLLOCNAME	Local location name	X						
MODE	Mode	X						
RMTNETID	Remote network ID	X						

PGMDEV

Specifies the program device name being defined (the name used by the program to do 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. This mapping allows you to have multiple sessions through the same configurations, and to have different device attributes for the same configurations. PGMDEV is a required parameter.

The rest of the parameters are associated with information supplied at various times during configuration. The following descriptions show the relationship of these parameters to the program device definition. (For information on how this information pertains to configuration, refer to the *Communications User's Guide*.)

RMTLOCNAME

Specifies the name of the remote location associated with the program device. The remote location name is the primary mapping to communications configurations. A remote location is associated with any device description that contains the same remote location name (RMTLOCNAME parameter on the Create Device XXXX (CRTDEVXXXX) command). For APPC, intrasystem, and SNUF communications, there can be a one-to-many relationship between the remote location name and the device description. For asynchronous, BSCEL, finance, and retail communications, there is a one-to-one relationship.

The remote location name is used by the system to select the device description. For those communications types that support multiple device descriptions per remote location, each communications type defines the criteria for selecting the best device. For a given remote location name, a list of devices (one or more) may be available for use.

Each communications type has specific rules for defining what constitutes an available device. Because asynchronous, BSCEL, finance, and retail communications have a one-to-one relationship between the device and remote location name, no device selection process is necessary. APPC, intrasystem, and SNUF communications all have a means for selecting the best available device for use.

For APPC, finance, intrasystem, retail, and SNUF communications, if you want to use a specific device description instead of allowing the system to select it for you, you can use the DEV parameter to further qualify the remote location to a specific device description.

Figure 4-4 on page 4-15 shows the relationship of the remote location to the device description.

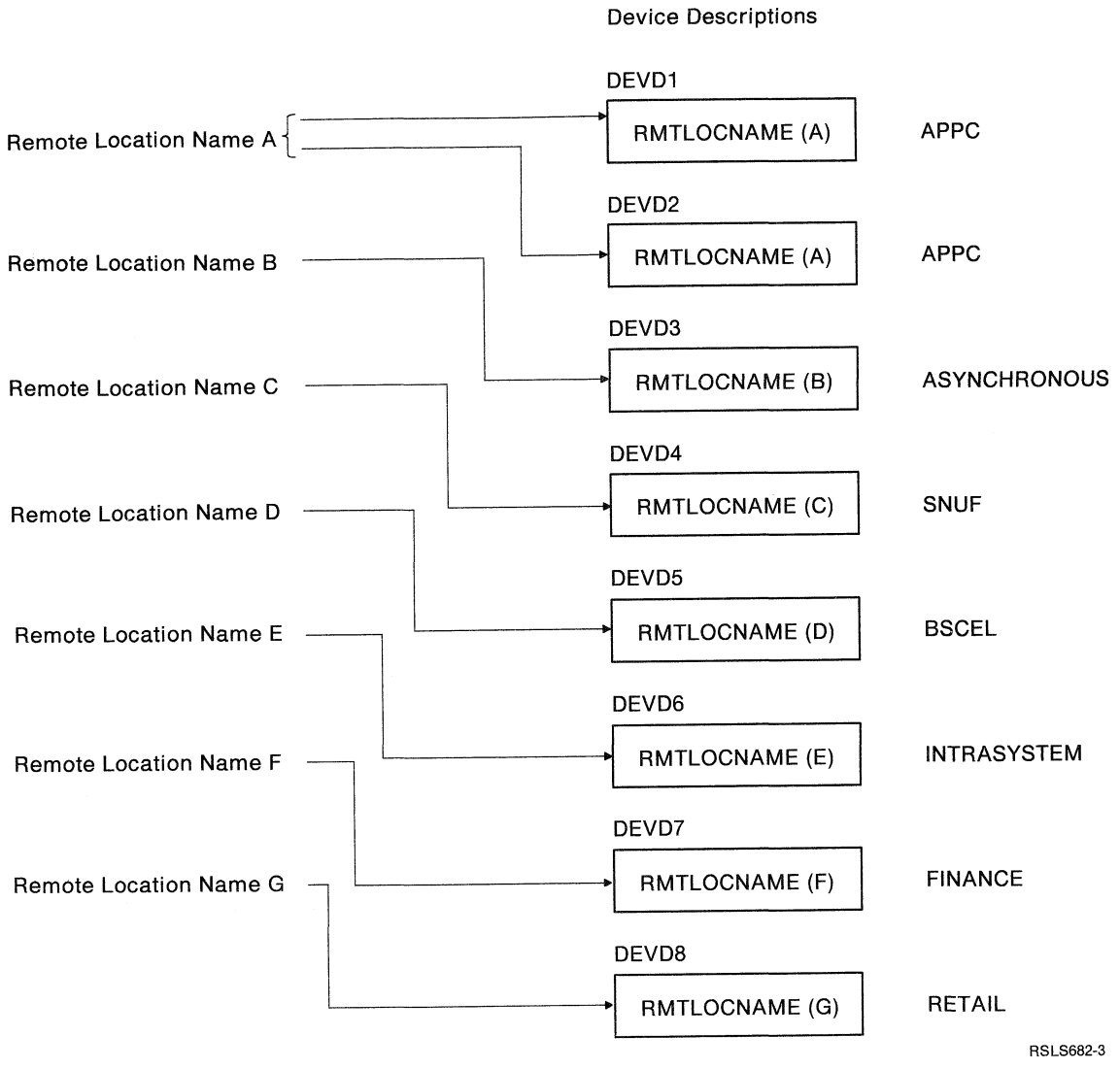


Figure 4-4. Relationship of Remote Location Name to Device Description

Note: Your program selects the communications type and communications link by acquiring a program device associated with a remote location name. The system selects a device description, based on availability (such as varied on and not in use), and other parameters that were specified when the program device entry was defined (such as device description). Note that multiple device descriptions can contain the same remote location name. In Figure 4-4, if your program acquires a program device associated with a remote location of 'A', the system either selects DEVD1 or DEVD2, and the session established uses the APPC communications type.

For additional information on how the system processes the RMTLOCNAME, DEV, LCLLOCNAME, and RMTNETID parameters for APPC, refer to the *APPC and APPN User's Guide*.

The remote location need not exist at the time you define the program device entry. However, the remote location must exist (either as a device description on the system, or as a remote location in the network) when the program device is acquired.

If the communications type you are using allows multiple sessions per remote location, the same remote location can be mapped to different program device names.

If the program device entry being defined is for the requester, the special value of *REQUESTER is used for the RMTLOCNAME parameter. The default for the remaining parameters on the ADDICFDEVE or OVRICFDEVE command in Table 4-4 is *LOC.

RMTLOCNAME is a required parameter.

DEV

Further qualifies the remote location to a specific communications device description.

DEV is an optional parameter. If you do not specify the DEV parameter, and there are several communications device descriptions associated with the remote location, the system determines which device to use. Note that the device used may not be the one you want (for example, the device you want may not be varied on).

LCLLOCNAME

Specifies the local location name of the local system. LCLLOCNAME is an optional parameter.

MODE

Specifies the mode used for the remote location. When you specify the special value *NETATR (the default) the mode in the network attributes is used. BLANK indicates that a mode name consisting of all blanks is used. MODE is an optional parameter.

RMTNETID

Specifies the remote network identifier of the remote location. When you specify the special value *NETATR (the default) the network identifier in the network attributes is used. *NONE indicates that a network identifier consisting of all blanks is used. RMTNETID is an optional parameter.

If any of the information supplied in the RMTLOCNAME, DEV, LCLLOCNAME, MODE, or RMTNETID parameters conflicts, the acquire operation to the program device fails.

Refer to the *CL Reference* for more information on the format and allowable values for these parameters. For more specific information, refer to *Communications User's Guide* and to the appropriate communications programming manual for the communications type you are using.

Communications-Type-Dependent Attributes

The second purpose of a program device entry is to establish the characteristics of the communications session. These communications-type-dependent attributes are specified as parameters on the ADDICFDEVE, CHGICFDEVE, and OVRICFDEVE commands.

Table 4-5 describes the communications-dependent attributes used with an ICF file and the communication types that support each attribute.

Table 4-5 (Page 1 of 2). Communications-Dependent Attributes

Parameter	Description	APPC	SNUF	BSCEL	Asyn-chronous	Intra-system	Finance	Retail
FMTSLT	Record format selection technique.	X	X	X	X	X	X	X
CMNTYPE	Identifies the communications type you are using to select prompting on the command.	X	X	X	X	X	X	X
APPID	VTAM identifier of the Customer Information Control (CICS/VS) or the Information Management System for Virtual Storage (IMS/VS) host system.		X				X	X
BATCH	Specifies whether this session is used for batch activity with IMS/VS host system.		X			X	X	X
HOST	Identifies type of host system with which to communicate.		X					
HDRPROC	Specifies whether received function management headers should be passed to the application.		X					
MSGPTC	Specifies whether message protection should be used.		X					
CNVTYPE	Conversation type.	X						
BLOCK	Specifies whether system or user will block and unblock transmitted records.			X				
RCDLEN	Maximum record length to transmit and receive.		X	X			X	X

Table 4-5 (Page 2 of 2). Communications-Dependent Attributes

Parameter	Description	APPC	SNUF	BSCEL	Asyn-chronous	Intra-system	Finance	Retail
BLKLEN	Maximum block length to transmit and receive.		X	X				
TRNSPY	Specifies whether text transparency is used when sending blocked records.			X				
DTACPR	Specifies whether blanks are compressed when sending and receiving data.			X				
TRUNC	Specifies whether trailing blanks are removed when sending data.			X				
GRPSEP	Specifies separator for groups of data (data sets and documents).			X				
RMTBSCEL	Specifies whether the session supports BSCEL commands and online messages.			X				
INLCNN	Specifies the method of making a connection on the line when a session is established.			X				
SECURE	Specifies whether this program device is secured from previously called override commands.	X	X	X	X	X	X	X

You can specify any parameter on any communications type, but the parameter is ignored if it is not supported by the specified communications type.

Although some attributes, like FMTSLT and CNVTYPE, have system-level defaults, the default for the majority of the parameters is the information supplied during communications configuration.

Parameters not specified on an ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command take on the appropriate system default or specified configuration value. If the same parameter is specified on both an ADDICFDEVE and OVRICFDEVE command, the value specified on the OVRICFDEVE overrides the value declared on the ADDICFDEVE command.

The OVRICFDEVE command follows the general rules for override processing. For information on determining the result when two OVRICFDEVE commands are specified for the same program device entry, refer to the *Data Management Guide*.

Format Selection (FMTSLT)

The FMTSLT parameter specifies the type of record selection used for input operations for the program device specified in the PGMDEV parameter.

Following are the values for the FMTSLT parameter:

***PGM** The record format is determined by the program.
***RECID** The record format is based on the use of the RECID DDS keyword.
***RMTFMT** The remote program determines the record format to use.

These different values have meaning only when used in conjunction with specific DDS keywords. Refer to Chapter 5 for more information on format selection processing.

Communications Type (CMNTYPE)

The CMNTYPE parameter identifies the communications type for which you are defining a program device entry. This identification prompts you for the communications-type-dependent attributes associated with the communications type you are using.

These values are available for the CMNTYPE parameter when *REQUESTER is not specified for the remote location name:

***ALL** Prompt for all possible communications-type-dependent attributes
***APPC** Prompt for all APPC-supported attributes
***SNUF** Prompt for all SNUF-supported attributes
***BSCEL** Prompt for all BSCEL-supported attributes
***ASYNC** Prompt for all asynchronous communications-supported attributes
***INTRA** Prompt for all intrasystem communications-supported attributes
***FINANCE** Prompt for all finance communications-supported attributes
***RETAIL** Prompt for all retail communications-supported attributes

This parameter is valid only if you enter the command interactively.

However, when you specify *REQUESTER for the remote location name (RMTLOCNAME), you are only prompted for selected values based on the CMNTYPE parameter:

***ALL** Prompt for FMTSLT, CNVTYPE, RCDLEN, BLKLEN, and SECURE parameters
***APPC** Prompt for FMTSLT, CNVTYPE, and SECURE parameters
***ASYNC** Prompt for the FMTSLT and SECURE parameters
***SNUF** Prompt for FMTSLT, RCDLEN, BLKLEN, and SECURE parameters
***BSCEL** Prompt for the FMTSLT and SECURE parameters
***INTRA** Prompt for the FMTSLT and SECURE parameters
***FINANCE** Prompt for the FMTSLT and SECURE parameters
***RETAIL** Prompt for the FMTSLT and SECURE parameters

Note: You can still specify values for the parameters that you are not prompted for (when *REQUESTER is specified for the remote location name) by typing those values and parameters on any command line with any of the program device entry commands. However, the parameter values you specify are ignored and no error return codes are issued.

Secure from Override (SECURE)

The SECURE parameter is valid only on the OVRICFDEVE command. This parameter does not apply to the ADDICFDEVE or CHGICFDEVE commands. This parameter is used to restrict the effects of override processing.

Refer to the *Data Management Guide* for information about how the SECURE parameter works. Refer to the *CL Reference* for more information about the format and allowable values.

Other Communications-Type-Dependent Parameters

Each of these parameters has specific meaning and function, depending on the communications type you are using. Also, some of these parameters are ignored if the program device being defined is for a RMTLOCNAME(*REQUESTER). Refer to the appropriate communications programming manual for more information on the use of these parameters.

Refer to the *CL Reference* for more information on the format and allowable values for these parameters.

Intersystem Communications Function Command Summary

Figure 4-5 shows the relationship between ICF commands.

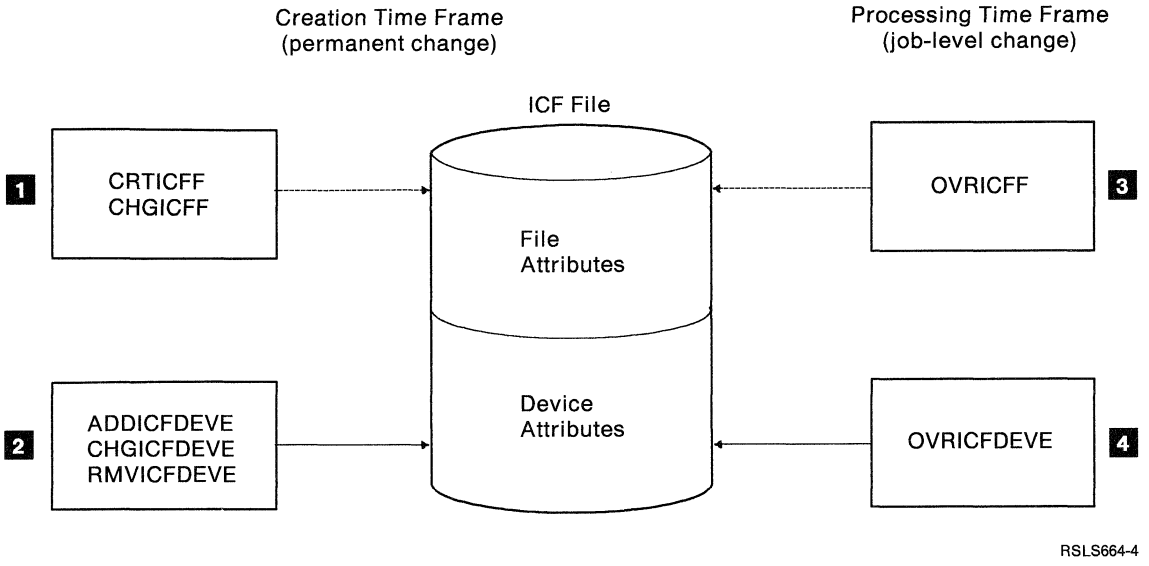


Figure 4-5. Relationship between ICF Commands

1. The CRTICFF and CHGICFF commands are used to create the ICF file and work with file-level attributes. (The DLTF command is used to delete the ICF file.)
2. The ADDICFDEVE, CHGICFDEVE, and RMVICFDEVE commands allow defining program device- or communications-type-dependent information in the ICF file. The ADDICFDEVE command is optional, and is used only to support early binding, and setting of system-wide defaults.
3. The OVRICFF command allows the changing of file-level attributes at the job level only. This command does not cause any permanent change, and it is not system-wide. (The DSPOVR command displays the information entered on the OVRICFF command. The DLTOVR command is used to delete the effects of the OVRICFF command.)
4. The OVRICFDEVE command affects the processing of the ICF file at the job level only. You do not specify the file on the override. Whatever ICF file is active at the time is the file that is overridden. It does not cause any permanent change, and it is not system wide. The OVRICFDEVE command uses a late binding function that ties the program device and the remote location at job time. The OVRICFDEVE command is also used to temporarily change communications-type-dependent attributes. The OVRICFDEVE command is job-wide, and has the characteristics of an override command. (The DLTOVRDEVE command is used to delete the effects of the OVRICFDEVE command.)

Chapter 5. Using an Intersystem Communications Function File

This chapter describes how an application uses an ICF file.

To use an ICF file, identify it as a WORKSTN file in RPG/400 or as a TRANSACTION file in COBOL/400. For C/400, the type of file need not be specified.

C/400, COBOL/400, and RPG/400 support an interface that allows the application to perform the following operations:

- Open the file
- Acquire a program device
- Read from a program device
- Write to a program device
- Release a program device
- Close the file

Read and write operations are done using a record that contains data description specifications (DDS) keywords. These DDS keywords allow more specific communications functions to be done with the read and write operations. ICF also supports system-supplied record formats that can be used in place of user-defined DDS record formats. Refer to Chapter 6 and Chapter 7 for more information about the communications functions that can be performed with the read and write operations.

Sample programs in Chapter 9 through Chapter 11 provide an overview of the language interface that supports these functions. For more information on the language interface, refer to the appropriate language reference manual.

Opening an Intersystem Communications Function File

The processing done by the open operation depends on whether the open is a subsequent open of a shared file. The open is a subsequent open of a shared file if you specify SHARE(*YES) and the file is currently open with SHARE(*YES) specified.

If the open is not a subsequent open of a shared file in the same job, the open operation allocates the file and any other resources needed to support the acquiring of program devices to the file, and allocates the input/output (I/O) buffers.

If the open is a subsequent open of a shared file, the program is simply attached to the already open file. Any program devices acquired by other programs are available for use by this program. The state or attributes of the file do not change during a subsequent open. For example, if the program device specified as the ACQPGMDEV parameter has been released, the subsequent open does not cause it to be acquired.

Refer to the *Data Management Guide* for more information about shared file processing.

After the file and other resources have been allocated, the open operation implicitly acquires the program device specified by the ACQPGMDEV parameter, on the Create Intersystem Communications Function File (CRTICFF), Change Intersystem Communications Function File (CHGICFF), or Override Intersystem Communications

Function File (OVRICFF) command. See "Acquiring a Program Device when the File Is Opened" on page 4-5 for information on how to specify the ACQPGMDEV parameter.

The following is a description of the processing done by the open operation based on the ACQPGMDEV parameter value specified for the file:

- If you did not specify the ACQPGMDEV parameter, or if you specified *NONE, the open operation does not acquire any program devices. A program device must be explicitly acquired for the file before the program tries any I/O operations to the file, or before another program opens the same file if the file is opened with SHARE(*YES) specified.
- If you specify a program device name on the ACQPGMDEV parameter, the open operation acquires the specified program device. See "Acquiring a Program Device" on page 5-2 for information about acquiring a specific program device.

If the open operation is not successful, the only allowable operation is closing the file. See "Closing an Intersystem Communications Function File" on page 5-21 for more information.

Obtaining Information about the Open Intersystem Communications Function File

After the program opens the file, an **open feedback area** is available to the program. This area contains information about the open file such as the file name, library name, and program device information. You can use the information in this area as long as the file is open. See Appendix C for a summary chart of the open feedback fields. Refer to the appropriate language reference manual for information on accessing the fields.

Acquiring a Program Device

Before any input or output operations can be directed to a program device, the program device must be acquired.

Only program devices defined to the file by use of the Add Intersystem Communications Function Device Entry (ADDICFDEVE) or the Override Intersystem Communications Function Device Entry (OVRICFDEVE) command can be acquired. See "Identifying the Devices Used with an Intersystem Communications Function File" on page 4-11 for more information about defining program device entries.

A program device can be acquired in two ways:

- One program device can be implicitly acquired through the open operation. Refer to "Opening an Intersystem Communications Function File" on page 5-1 for more information on an implicit acquire through the open operation.
- A program device can be explicitly acquired through the acquire operation. The acquire operation can be used many times with different program device names.

When a program device is explicitly acquired with an acquire operation, you identify the session to establish by using the same program device name on the acquire as specified on the PGMDEV parameter on the ADDICFDEVE or the OVRICFDEVE command.

The examples in Figure 5-1 show the relationship between the program device entry (defined using an ADDICFDEVE or an OVRICFDEVE command) and a C/400, a COBOL/400, and an RPG/400 operation.

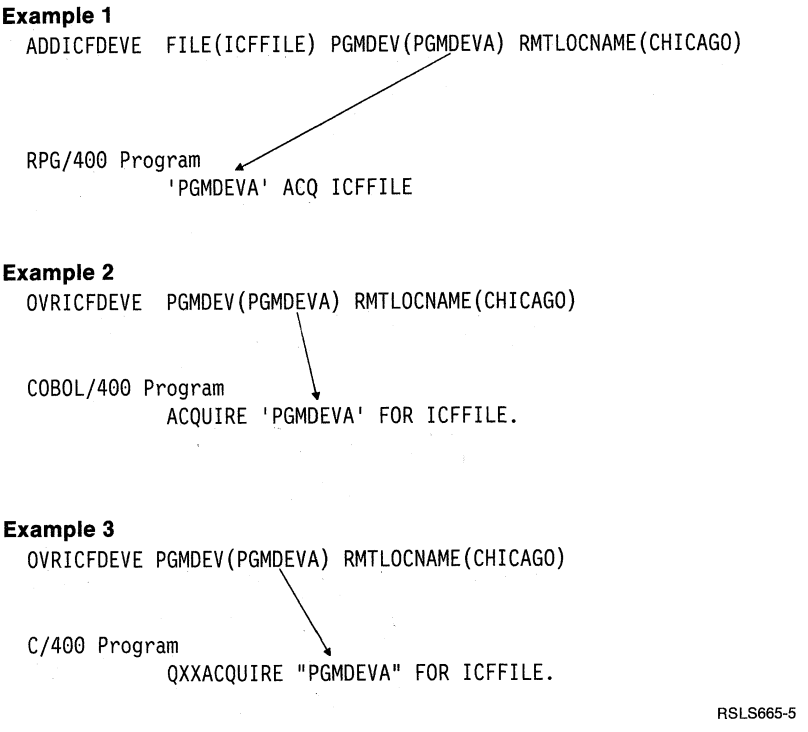


Figure 5-1. Relationship of Program Device Entries to Operations

Acquiring the program device automatically allows any I/O operations valid for that program device to be issued. For example, if the file is opened for input only, the read operation is allowed, but the write operation is not allowed.

The amount of time the system waits for resources to become available to complete the acquire request is specified on the WAITFILE parameter of the CRTICFF, CHGICFF, or OVRICFF command.

The following sections describe some of the functions performed when a program device is acquired.

Acquiring a Program Device — Source Program

The system tries to allocate a new session with the remote location for the job in which the program is running.

Some causes of a failed acquire operation are:

- The device associated with the program device is not varied on.
- The device description for the device associated with the program device is allocated to another job.
- A session is not available for the remote location.

Acquiring a Program Device — Target Program

The system tries to establish a connection with the requesting program device. An acquire operation to the requesting program device does not allocate a new session. It only establishes a logical connection to the session and transaction on which the target program was started.

The acquire operation fails if any of the following occur:

- The session was previously ended by the target program.
- A program not started by an evoke function issues an acquire for a requesting program device.
- The requesting program device is acquired for another file in the job.

Obtaining Information about a Particular Program Device

You can obtain information about a program device in two ways:

- Using the **program device definition list**
- Using the **get-attributes operation**

Program Device Definition List

After a program device is acquired, the program device becomes part of the program device definition list. The program device definition list contains information about the program device, such as the program device class, device type, and invite state.

You can use the information in the program device definition list as long as the program device is acquired. The support provided by the high-level language you use determines whether you can access this information.

See Appendix C for a summary chart of the program device definition list. Refer to the appropriate language reference manual for information on accessing these fields.

Get-Attributes Operation

The get-attributes operation can be used at any time after a file has been opened to determine the status of a particular program device. The program device does not need to be acquired. The operation gets the current status about the session in which your program is communicating. If the program device is not acquired, the information is obtained from the program device entry defined with the ADDICFDEVE or OVRICFDEVE command.

The status information received by the get-attributes operation contains the fields shown in Table 5-1 on page 5-5.

Table 5-1 (Page 1 of 2). Attribute Information Fields

Position	Value	Meaning
1 through 10	Name	Program device name: The name the program used to identify the program device in the file to read and write from.
11 through 20	Name	Device description name: The device description associated with the program device name (specified during configuration and optionally on the ADDICFDEVE or OVRICFDEVE command).
21 through 30	Name	User ID: If the program was started locally, this is the user ID used to sign on to the work station. If the program was started as a result of a program start request, this is the user ID used to start the target program.
31	I D U	The device is an ICF device type. The device is a display device. Unknown.
32 through 37	APPC SNUF BSCEL ASYNC INTRA FNC RETAIL	APPC communications type. SNUF communications type. BSCEL communications type. Asynchronous communications type. Intrasystem communications type. Finance communications type. Retail communications type.
38	Y N	This is a requesting program device. This is a program device acquired by a source program.
39	Y N	Program device has been acquired. Program device has not been acquired.
40	Y N	Input is invited for this program device. Input is not invited for this program device.
41	Y N	Invited input is available for this program device. Invited input is not available for this program device.
42 through 50	Reserved	Not applicable to communications.
51	Y N	Session has an active transaction. Session does not have an active transaction.
52[1]	0 1	Synchronization level is NONE. Synchronization level is CONFIRM.
53[1]	M B	Mapped conversation. Basic conversation.
54 through 61	Name	Remote location name: This is the remote location associated with the program device name (specified during configuration and on the ADDICFDEVE or OVRICFDEVE command).
62 through 69[1]	Name	Local logical unit (LU) name.
70 through 77[1]	Name	Local network ID.
78 through 85[1]	Name	Remote LU name.
86 through 93[1]	Name	Remote network ID.

Table 5-1 (Page 2 of 2). Attribute Information Fields

Position	Value	Meaning
94 through 101[1]	Name	Mode: This is the mode associated with the program device name (specified during configuration and optionally on the ADDICFDEVE or OVRICFDEVE command).
102 through 144	Reserved	

1 This information is valid only for some of the communications types. These fields will be blank if the information does not pertain to the communications type you are using.

Sending and Receiving Data

Data is sent between systems by using output (or write) and input (or read) operations. The read and write operations are done using a record format. The results of read and write operations are communicated to the program with ICF messages, major/minor return codes, high-level language status values, and an **I/O feedback area**.

The I/O feedback area is updated for every read/write operation. The I/O feedback area consists of two sections:

- The **common I/O feedback area** contains information relevant to all communications types.
- The **file-dependent I/O feedback area** contains information that can apply to one or more of the communications types.

Common I/O Feedback Area

The common I/O feedback area contains information in the following fields:

- **Output operation count.** A count of the number of successful output operations. This count is updated only when an output operation completes successfully.
- **Input operation count.** A count of the number of successful input operations. This count is updated only when an input operation completes successfully and data is received.
- **Output then input operation count.** A count of the number of successful output then input operations.
- **Count of other operations.** A count of the number of successful operations other than output and input operations (such as acquire and release operations).
- **Current operation.** A hex value representing the current (last requested) operation, sent as follows:

  Hex 01 Input

  Hex 05 Output

  Hex 06 Output then Input

  Hex 11 Release

  Hex 12 Acquire

- **Record format name.** Name of the record format just processed. The record format is either specified on the I/O request or determined by the specified format selection processing option.
- **Device class and type.** A hex code representing a device class for ICF and the communications type used as follows:

  Hex 0B0E APPC

  Hex 0B20 SNUF

  Hex 0B0A BSCEL

  Hex 0B1F Asynchronous

  Hex 0B1E Intrasystem

  Hex 0B42 Finance

  Hex 0B43 Retail
- **Program device name.** The program device name to which the last operation was issued.
- **Record length.** The record length of the last I/O operation based on the record format processed.
- **Blocked record count.** The number of records sent or received on an I/O operation. For ICF, the value is always 1.

File-Dependent I/O Feedback Area

The file-dependent I/O feedback area contains information in the following fields:

- **Actual record length.** On input, this is the actual length of user data received from the remote system or device. When the data received is longer than the data requested (all the received data cannot be contained in the record format used), the length of data is provided, if known. If the actual length cannot be determined, the field is set to hex FFFFFFFF. When a partial record is received (the remainder of the record is never sent), the length of the data received is provided. If the input operation completes with an error (other than partial record or truncated record), the contents of the field are undetermined.

  On output, the actual length is the length of data moved from the user's buffer to the output buffer to send to the remote system. If the output operation completes with an error, the contents of the field are undetermined.
- **ICF major/minor return code.** A 4-character code (2 characters representing the major code, 2 characters representing the minor code) indicating the results of each operation.
- **Negative response error data.** For some return codes, this field contains more detailed information about the reason for the error. Refer to the *APPC and APPN User's Guide*, the *Finance Communications Programmer's Guide*, and the *Retail Communications Programmer's Guide* for more information on this field.
- **Request-to-write indication.** This indication tells you if the remote system requested that the application program stop sending data and give permission (by issuing a read or an allow-write request) to the remote system to begin sending.
- **Remote format name.** The remote format name received from the program device on an input operation. This is valid when the FMTSLT option on the

ADDICFDEVE or OVRICFDEVE command is *RMTFMT. See Chapter 6 for more information on the FMTNAME DDS keyword.

- **Mode.** Mode associated with the program device. Mode is for APPC only. Refer to the *APPC and APPN User's Guide* for more information on modes.
- **Safe indicator.** This field shows that an end-of-text (ETX) control character has been received in the buffer, and it is only valid for BSCEL. The safe indicator is not set if BLOCK(*USER) was specified on the ADDICFDEVE or OVRICFDEVE command. Refer to the *BSC Equivalence Link Programmer's Guide* for more information on this indicator.

Refer to Appendix C for a summary of the fields and the communications types to which the information applies. Refer to the appropriate communications programming manual for specific details on pertinent fields.

Like the open feedback area, the support provided by the high-level language you use determines whether you can access this information.

Checking Return Codes

After each operation, an ICF **return code** is returned to your program. Your program checks this return code to determine:

- The status of the operation just completed
- The operation that should be done next

It is recommended that your program check these return codes at the completion of every operation to ensure that the operation completed successfully or that the appropriate recovery action is taken.

A summary of these return codes is described in Appendix B. These codes, or groups of codes, are also converted to language return codes. For example, the ICF codes are converted to RPG *STATUS values or to COBOL/400 file status values. C/400 does not have file status values. These values are shown in a chart in the appropriate language chapter of this manual.

Each ICF return code consists of a 2-digit **major code** and a 2-digit **minor code**. The major code identifies the general condition for a group of return codes, and is usually sufficient to determine the action to be taken. The minor code identifies the specific condition and may indicate the specific action that should be taken next. For example:

```
8233
| |
| Minor code
|
Major code
```

In this example the major code **82** indicates that an acquire or open operation was not successful. The minor code **33** indicates that the operation failed because an ADDICFDEVE command or OVRICFDEVE command was not issued for the program device you are trying to acquire.

Usually, your program determines the action to take by checking only the major code or the language status code. However, you may need to check minor codes for specific conditions that occur for your particular application or communications configuration.

For more information about major and minor return codes, refer to Chapter 8.

Writing to a Program Device

Use a write operation to send data to the remote system.

The ICF file supports a set of DDS processing keywords and system-supplied formats, used in conjunction with the write operation, to perform various communications functions. Refer to Chapter 6 and Chapter 7 for more information about specifying the communications function to be used with the write operation.

Note: Data can be written to only one program device for each write operation.

Figure 5-2 shows the use of the write operation when sending data.

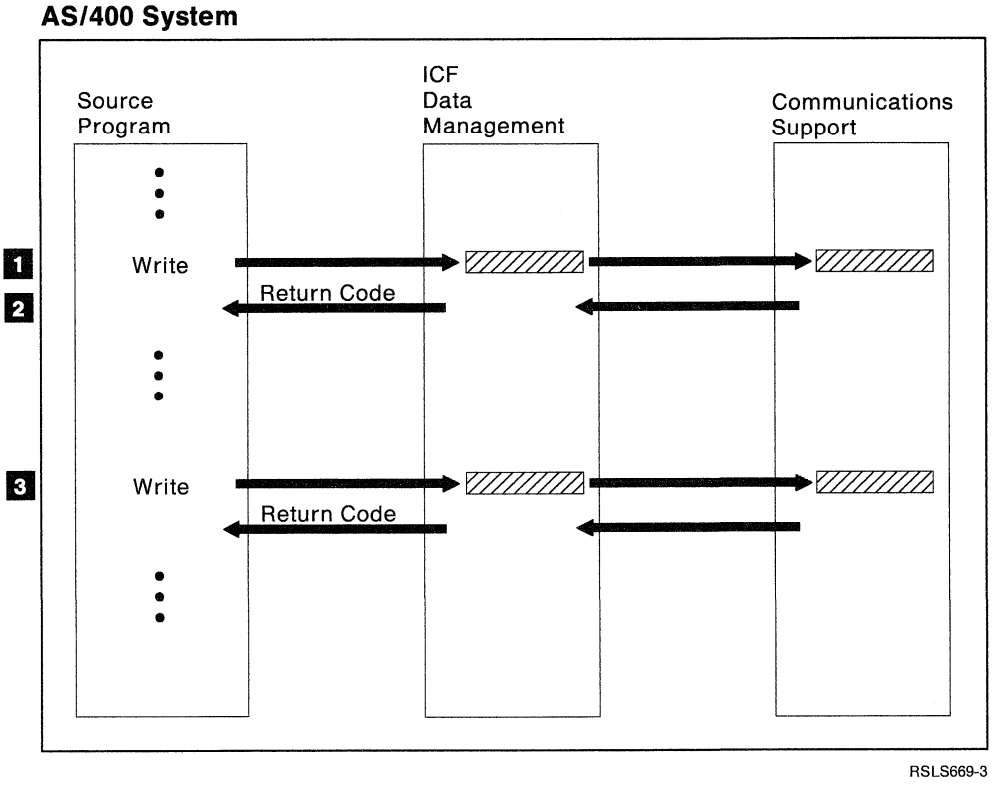


Figure 5-2. Using Write Operation when Sending Data

1. Your program uses a write operation to send data to the remote system.
2. Your program receives a return code indicating the completion status of the operation.
3. If a successful return code is received, your program continues sending several records.

Inviting a Program Device

Your program indicates it wants to start an asynchronous input operation by inviting a program device. The invite function prepares your program to receive data. Your program can continue processing after issuing the invite request, and does not have to wait for the data.

An invite function is specified by:

- Issuing a write operation to a program device using a record format with the INVITE DDS keyword in effect.
- Issuing a write operation to a program device using a system-supplied format whose definition contains the invite function.

For more information about specifying the invite function, see Chapter 6 and Chapter 7.

You can receive the response from an invited program device by doing an input operation. See "Reading from Invited Program Devices" on page 5-13 and "Reading from One Program Device" on page 5-19 for more information.

At least one program device must be invited before a read-from-invited-program-devices operation can return a response from a program device.

A program would invite a program device because:

- Inviting several program devices allows the program to do a read-from-invited-program-devices operation and receive data from one of the invited program devices with a response available. Therefore, using the invite function of the ICF file, the program can handle receiving a record from any one of several invited program devices by reading from a single point in the program. The program issues an input operation for the response.
- A program wants to use the time-out capability of the read-from-invited-program-devices operation.

Format Selection Processing

The format selection (FMTSLT) parameter on the ADDICFDEVE and OVRICFDEVE commands determines how ICF data management selects the record format to use when receiving data with the read and read-from-invited-program-devices operations. The three different methods of selecting a record format are discussed here. The name of the record format selected is placed in the I/O feedback area. Refer to "Determining the Record Format Returned" on page 5-19 for more information about determining the record format selected.

Program Selection (*PGM)

If you specify FMTSLT(*PGM), which is the default, on the ADDICFDEVE or OVRICFDEVE command, your program must specify the record format to use when your program does an input operation. If no record format name is given, ICF data management uses the default record format. The default format is always the first format defined in the file.

The only selection process that is applicable when using the system-supplied QICDMF file is FMTSLT(*PGM). The default record format in this file is a 4096-byte data record called DFTRCD. You should either specify this format on the input oper-

ation or allow the system to default. Your program must then examine the input data to determine what data processing to perform on the fields in the record.

Record-Identifier Selection (*RECID)

Selecting FMTSLT(*RECID) on the ADDICFDEVE or OVRICFDEVE command provides a means of identifying and selecting the record format to use based on the data received. If you specify FMTSLT(*RECID), the file is searched for the RECID keyword on each input operation. The RECID keyword provides a definition for determining which record format to use.

When you specify the RECID keyword, you define a compare value. You must define the beginning position in the record format and the compare value to use. When data is received, the corresponding positions in the record are compared to the defined RECID values. When a match is found, that record format is used to process the received data. If no match is found, or if no data is received, the default record format is used.

When the FMTSLT(*RECID) is specified, the default format for an ICF file is one of the following:

- The first format in the file without the RECID keyword specified
- The first format in the file if all formats have the RECID keyword specified (applies only when no data is received)

Notes:

1. An ICF return code of 81E9 is returned to your program if the default format has the RECID keyword specified and no match is found for the received data. Refer to Appendix B for a complete list of return codes.
2. If a read with a record format specified is issued, the format specified must match the name determined by the RECID keyword selection process. If not, return code 3441 is returned to your program.

Refer to Chapter 6 for more information about the RECID keyword.

Remote Format Selection (*RMTFMT)

Remote format selection is supported by APPC and intrasystem communications.

If you specify FMTSLT(*RMTFMT) on the ADDICFDEVE or OVRICFDEVE command, your program does not need to enter a format name when it does an input operation. Instead, the format name passed with the data from the remote program is used. If the remote system is an AS/400 system, the remote program must specify the FMTNAME keyword in the record used to send the data to ensure the format name is sent.

If the remote system does not send a format name (for example, a record is sent without a FMTNAME keyword specified) and a format name is specified with the input operation in your program, that name is used to process the data received. If no format name is specified on the input operation, the default record format in the ICF file is used. The default record format is the first record format in the file.

If the remote program sends a format name and your program specifies a format name, the names must match. If they do not match, return code 3441 is returned to your program.

If the remote program sends a format name and FMTSLT(*RMTFMT) was not specified on the ADDICFDEVE or OVRICFDEVE commands, the remote format name sent is ignored by ICF.

The record format name received from the remote system on a successful input operation is put in the file-dependent section of the I/O feedback area. The high-level language may access this area to determine the remote record format name received from the remote system. Refer to the appropriate language reference manual for more information about accessing the I/O feedback area.

Summary of Format Selection Processing

Table 5-2 summarizes the record format that is selected based on the format selection option specified on the FMTSLT parameter and the record format name specified on the input operation (if one was specified). This chart also shows what return codes can result from the format selection process on an input operation.

Table 5-2 (Page 1 of 2). Format Selection Options

FMTSLT Option	Input Data	Record Format Name Specified on Input Operation	Record Format Name Not Specified on Input Operation
*PGM	Does not apply to format selection.	If specified format name is defined in ICF file, specified format selected. Otherwise, return code 83E0 returned to program.	Default format[1] selected.
*RECID	Data received matches record format in file with RECID keyword.	If matched format is same as specified format, matched format selected. Otherwise, return code 3441 returned to program.	Matched format selected.
	Data received does not match any record in file with RECID keyword.	If default format[2] does not have RECID keyword, default format selected. Otherwise, return code 81E9 returned to program. If format selected (default[2]) is not same as specified format, error return code 3441 returned to program.	If default format[2] does not have RECID keyword, default format selected. Otherwise, return code 81E9 returned to program.
	No data received.	If default format[2] is same as specified format, default format selected. Otherwise, return code 3441 returned to program.	Default format[2] selected.

Table 5-2 (Page 2 of 2). Format Selection Options

FMTSLT Option	Input Data	Record Format Name Specified on Input Operation	Record Format Name Not Specified on Input Operation
*RMTFMT	Record format name (remote format) received from the remote system	If remote format name is defined in the ICF file, remote format selected. Otherwise, default [1] format selected. If format selected is not same as specified format, return code 3441 returned to program.	If remote format is defined in the ICF file, remote format selected. Otherwise, default format[1] selected.
	Record format name (remote format) not received from the remote system	If specified format name is defined in the ICF file, specified format selected. Otherwise, error return code 83E0 returned to program.	Default format[1] selected.

1 The default record format for FMTSLT(*PGM) and FMTSLT(*RMTFMT) is the first record format in the ICF file.

2 The default format for FMTSLT(*RECID) is the first record format in the ICF file that does not have a *RECID keyword specified, or the first record format if all record formats have the *RECID keyword specified.

Reading from Invited Program Devices

The primary purpose of the read-from-invited-program-devices operation is to provide a single point in the program at which the program can wait for and receive a record from one of several program devices. The read-from-invited-program-devices operation can wait for and return a record to the program from one of the invited program devices with an available record.

The read-from-invited-program-devices operation is also used to check whether the timer that was set by the timer function has ended. Refer to Chapter 6 and Chapter 7 for more information on the timer function.

The read-from-invited-program-devices operation can complete when:

- A communications failure is detected on one of the invited program devices.
- The job is being canceled (controlled).
- The time specified by either the timer function or the WAITRCD parameter on the CRTICFF command is reached.

The read-from-invited-program-devices operation is only valid if the high-level language you are using considers the ICF file to be a multiple device file. See the appropriate language reference manual for more information.

Figure 5-3 on page 5-14 shows how you can use the invite function and read-from-invited-program-devices operation to receive data from two different program devices.

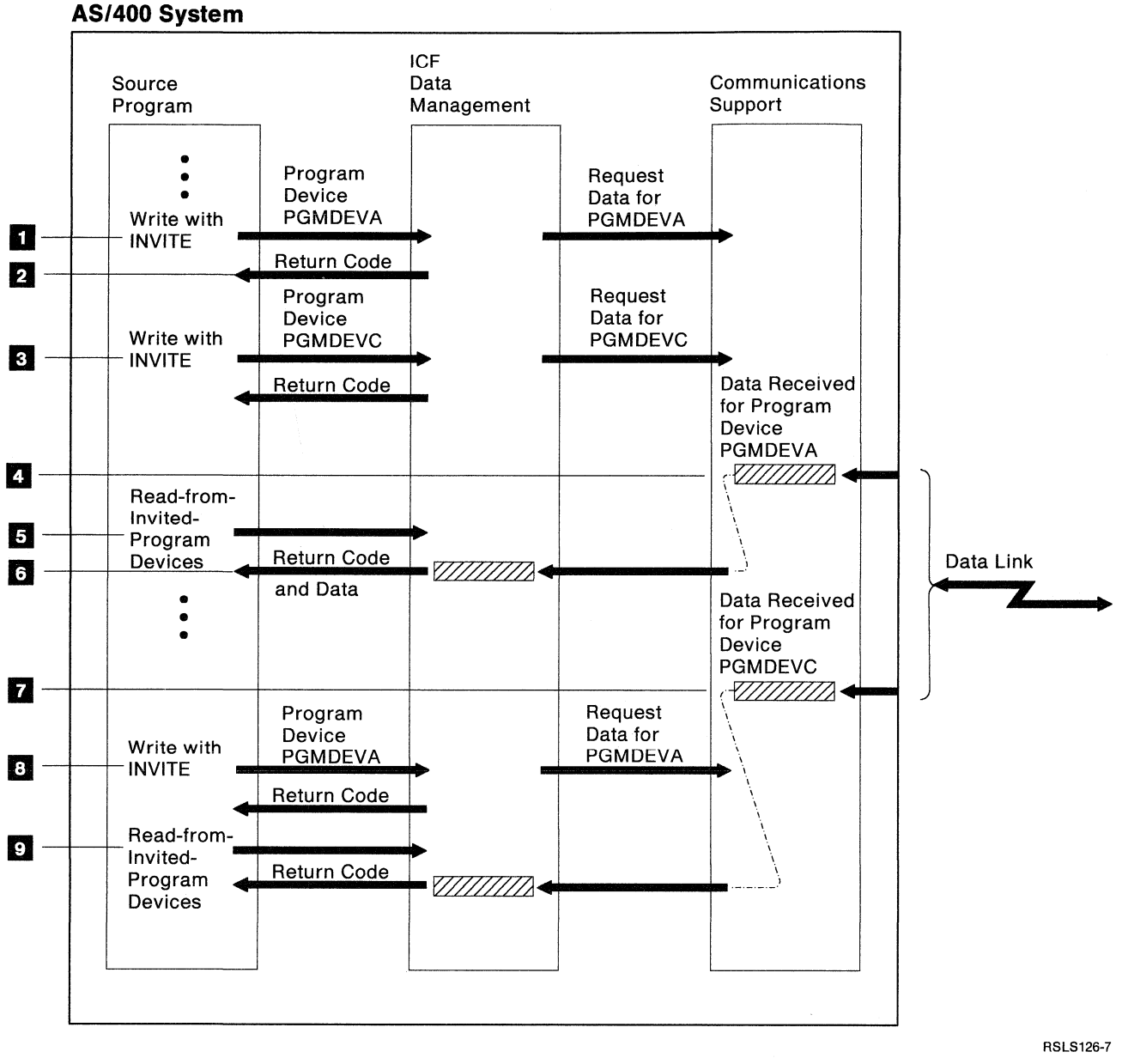


Figure 5-3. Using the Invite Function and Read-from-Invited-Program-Devices Operation to Receive Data

1. Your program uses the invite function to ask the remote system to send data for program device PGMDEVA.
2. A successful completion return code tells your program that ICF data management received the operation and is asking the remote system to send data. No data has yet been received.
3. Your program issues another invite function. This invite is for program device PGMDEVC. Data has not yet been received for the first invite for program device PGMDEVA.
4. The system receives data for program device PGMDEVA. (Data is not necessarily received in the order in which the invite functions were issued. For example, data can be received for program device PGMDEVC before data is received for PGMDEVA. Your program must check the program device name to determine for which program device the data is received.)

5 A read-from-invited-program-devices operation is used to receive the data sent.

6 This time a successful completion return code tells your program that data has been received and is in your program buffer.

7 Data is received for program device PGMDEVC.

8 Another invite function is used to ask for program device PGMDEVA data.

9 Another read-from-invited-program-devices operation is used to receive the data for program device PGMDEVC.

Specifying Maximum Wait Interval

You can specify the maximum amount of time your program will wait for a read-from-invited-program-devices operation to complete.

The time interval can be specified by:

- Specifying the WAITRCD parameter on the CRTICFF, CHGICFF, and OVRICFF commands
- Issuing the timer function

The WAITRCD parameter establishes the maximum time interval used for all read-from-invited-program-devices operations issued against the file.

The timer function is used to specify the maximum time interval used for read-from-invited-program-devices operations until either the timer ends or a new interval is set using the timer function. When the interval for the timer operation is in effect, the value specified on the WAITRCD parameter is ignored.

If a response is not received from the invited program devices within the specified amount of time, the program is notified with an ICF return code (0310) indicating that the timer interval has ended.

Figure 5-4 on page 5-16 shows the relationship between the timer function and the read-from-invited-program-devices operation.

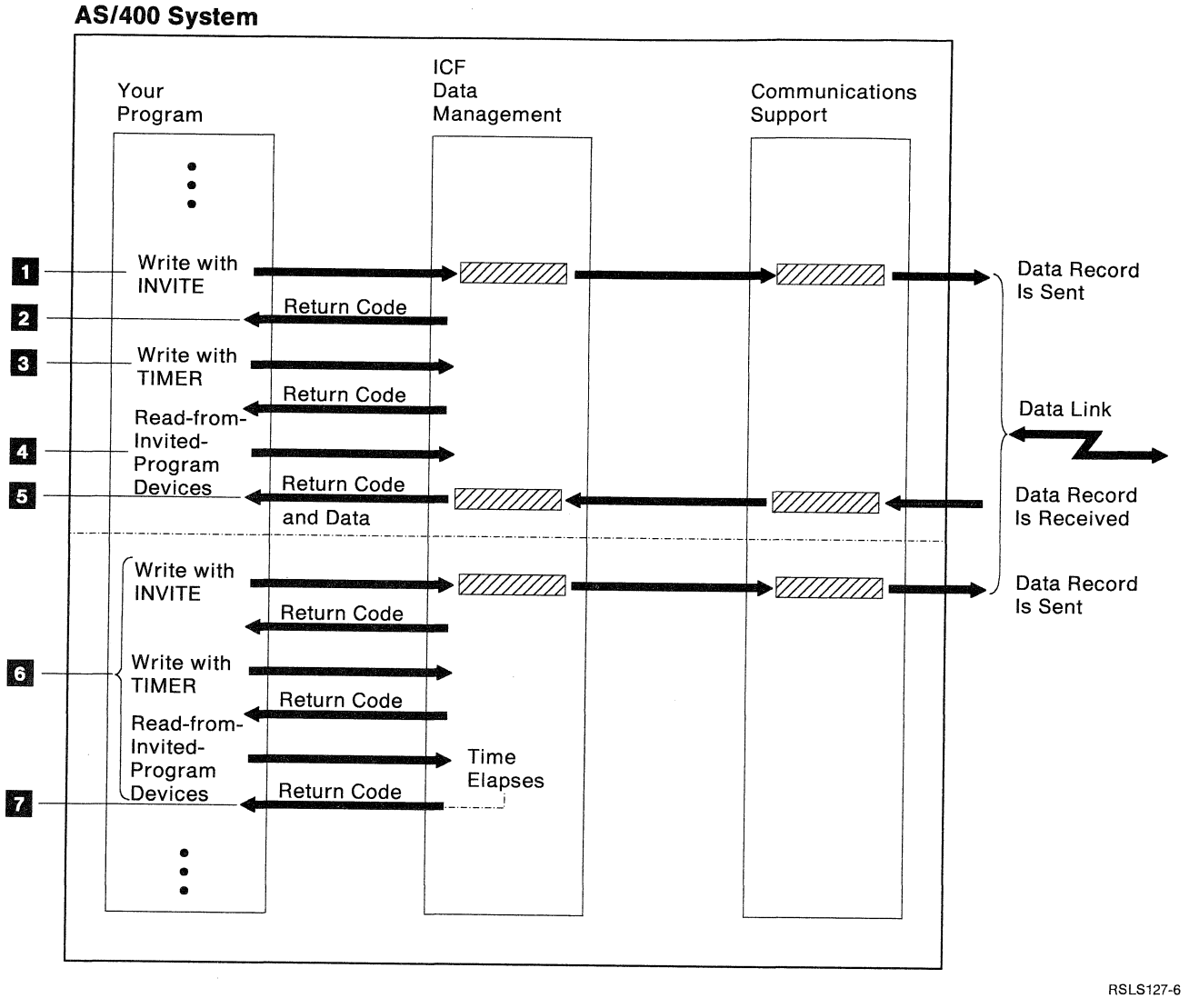


Figure 5-4. Relationship between Timer and Read-from-Invited-Program-Devices Operations

1. Your program issues an invite function.
2. A successful completion return code tells your program that ICF data management accepted the request and that the remote system is expected to respond. No data has yet been received.
3. Your program uses the timer function to set the maximum length of time to wait for a response.
4. A read-from-invited-program-devices operation is used to read the data or, if the data is not received within the length of time specified, the return code indicates that the time you set has elapsed.
5. The data is received before the time elapses. Therefore, a successful completion return code is passed to the program with the response from the remote system.
6. Your program again uses the invite function to send data and ask for a response. The time interval is again set, and another read-from-invited-program-devices operation is issued.

7 This time the read-from-invited-program-devices operation results in a return code that tells your program the time elapsed before a response was received from the remote system. When this happens, you may want to send a message to the operator, continue processing, or both.

COBOL/400 provides a means of calling the read-from-invited-program-devices operation as if the WAITRCD(*IMMED) is specified. Refer to the *COBOL/400 Reference Manual* for information about the NODATA phrase. See "Determining the Wait-for-Record Value" on page 4-7 for information about specifying the WAITRCD parameter.

Refer to Chapter 6 and Chapter 7 for information on specifying the timer function.

Responses

A program can receive one of many responses from the read-from-invited-program-devices operation. These responses are communicated to your program through an ICF return code.

The ICF file also supports a set of DDS response indicators that can be used in conjunction with the read-from-invited-program-devices operation to indicate status information about the operation. Refer to Chapter 6 for more information about response indicators.

The following sections describe possible responses from the read-from-invited-program-devices operation and the conditions under which the program receives the response.

Data from One of the Invited Program Devices with Data Available: If at least one invited program device has data available and the job has not been ended (controlled), the read-from-invited-program-devices operation returns data from one of the invited program devices.

After the read-from-invited-program-devices operation completes, the program can examine feedback that allows it to identify the program device that returned the data and the record format of the data returned. See "Determining Which Invited Program Device Had Data Available" on page 5-18 for information about how to determine which program device responded. See "Determining the Record Format Returned" on page 5-19 for information about determining the format of the returned record.

The program device returning the data is no longer in the invited state. The program device must be invited before it can return any more data using the read-from-invited-program-devices operation. Other invited program devices remain invited.

Job Ended (Controlled): The read-from-invited-program-devices operation returns this response if the job is ended (controlled) before or during the wait for data to become available from an invited program device.

Receiving the job ended response does not cancel the invite. All invited program devices remain invited.

If any program in the job is notified that the job is being ended (controlled), that program should notify all other programs in the job. The system notifies only one program regardless of how many ICF files are used in the job.

When a program receives a job ended (controlled) indication, the program should complete operations and end before the system changes the job ended (controlled) to job ended (immediate) and forces all processing to stop. This action is important if a program needs to complete some processing before it ends.

No Invited Program Devices Have Data Available: The read-from-invited-program-devices operation returns this response when the job is not being ended and:

- At least one program device is invited.
- No data is available from any of the invited program devices.
- The WAITRCD(*IMMED) parameter is specified.

All invited program devices remain invited.

Time-Out on Wait for Data from Invited Program Devices: The read-from-invited-program-devices operation returns this response when:

- No data is available from any of the invited program devices in the amount of time specified as the WAITRCD parameter or the timer function.
- The job is not being ended.

All invited program devices remain invited.

No Program Devices Invited: If no program devices are in the invited state and the timer function is not in effect, the program is notified that no program devices were invited.

Error from One of the Invited Program Devices: The read-from-invited-program-devices operation can return an error condition instead of data from one of the invited program devices if:

- The job is not being ended.
- At least one invited program device has a response available.

If an invited program device detects an error while it does the input operation, the error (like the data) is held until the program device is read using a read-from-invited-program-devices or read operation.

Determining Which Invited Program Device Had Data Available

After a read-from-invited-program-devices operation returns data from an invited program device, the program may need to identify the name of the program device from which the data was returned. This identification is necessary if the program wants to handle one program device differently from other program devices.

The program can determine the name of the program device that returned the data from a field in the I/O feedback area. Refer to the appropriate language reference manual to learn about other ways to get this information, and about how to access the I/O feedback area.

If the program needs the name of the program device that returned the data, the program must get that information before doing any other I/O operations to the file.

If the read-from-invited-program-devices operation did not return a data available response (some other response, like job ended (controlled) was returned to the program), the field containing the name of the program device to which I/O was last directed in the I/O feedback area is set to *N (not applicable).

Determining the Record Format Returned

Because a read-from-invited-program-devices operation returns a record from one of the invited program devices with an available record, regardless of that record's format, you cannot specify a record format on the read-from-invited-program-devices operation.

The system uses the FMTSLT parameter on the ADDICFDEVE, CHGICFDEVE, and OVRICFDEVE commands to determine the record format. Therefore, the program may have to determine the format of the record returned before handling the record (if a program device can return a record in one of several record formats).

The program can determine the record format of the data returned from a field in the I/O feedback area that indicates the name of the last record format used for I/O. Refer to the appropriate language reference manual to learn about other ways of getting this information and about how to access the I/O feedback area.

Note that if the program needs the name of the record format used to receive the data, the program must get that information before doing any other I/O operations to the file.

Reading from One Program Device

A read operation waits for and receives data from one program device. There is no time limit on a read operation (the WAITRCD parameter and the interval specified on a timer function are ignored). The program waits until data is available from that program device. The read operation differs from the read-from-invited-program-devices operation in that a read operation is directed to a specific program device, whereas the read-from-invited-program-devices operation receives data from any program device that was previously invited.

The program can indicate to the system that a read operation must be done in three ways:

- Explicitly, by specifying the name of a program device on the read operation
- Implicitly, by specifying the name of a record format on the read operation
- Implicitly, by specifying neither a record format or program device name, and the high-level language considers the ICF file a single device file

Because you cannot specify a record format on a read-from-invited-program-devices operation, the system interprets a read with a record format specified as a read operation. See the appropriate language reference manual for information about calling the read operation explicitly or implicitly and about which program device is used if the read operation is implicitly called.

Figure 5-5 on page 5-20 shows how to use the read operation.

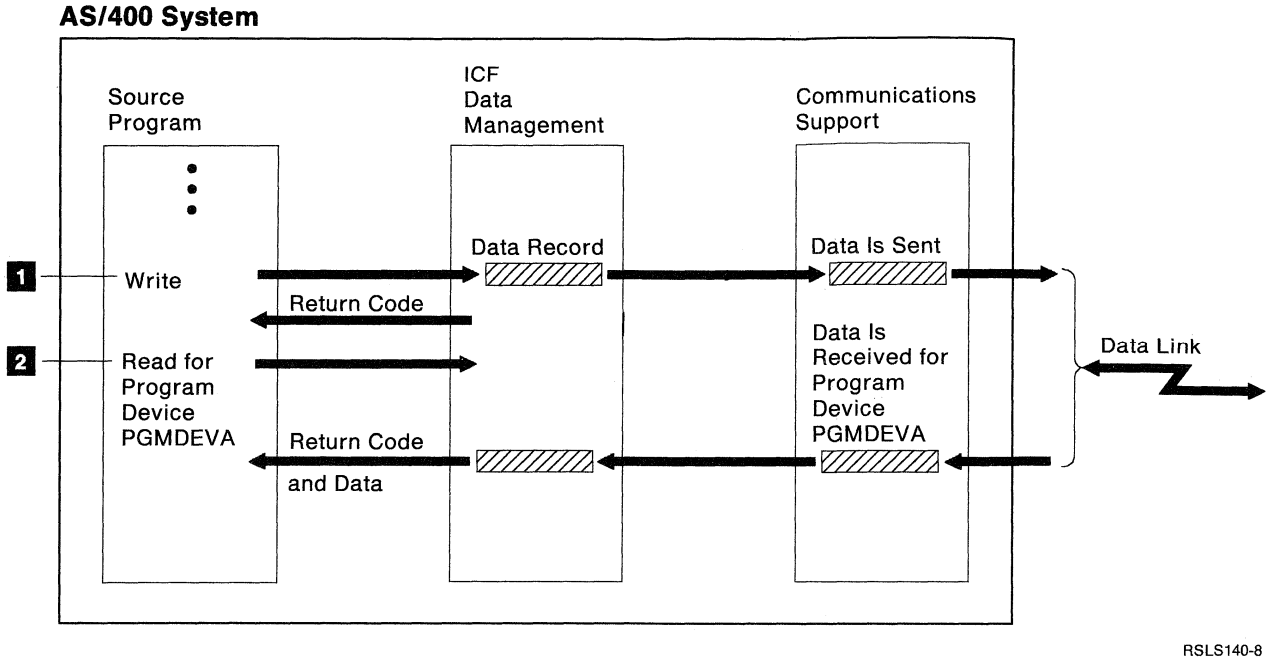


Figure 5-5. Using the Read Operation

1. Your program uses a write operation to send data to the remote system.
2. A read operation is then issued to receive data from PGMDEVA. The program waits to receive the data before continuing.

If the program device specified on the read operation has been invited, the invite is satisfied and the input operation started when the program device was invited is completed before control is returned to the program.

The ICF file supports a set of DDS response indicators that can be used in conjunction with the read operation to indicate status information about the operation. Refer to Chapter 6 for more information about response indicators.

Writing and Then Reading from One Program Device

Some high-level languages support an interface to send a single I/O operation that does a write operation followed by a read operation to a program device.

The same record format is used on both the write and read operation.

Canceling an Invite of a Program Device

If a program device is invited, it is possible to cancel the invite:

- Explicitly, by issuing a cancel-invite function to the program device
- Implicitly, by issuing a write operation to the program device

Refer to Chapter 6 and Chapter 7 for information about specifying the cancel-invite function.

Releasing a Program Device

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 reacquire it before you can use it again for I/O operations.

The release operation ends the session only when certain criteria are met. The end-of-session function always ends the session. Refer to Chapter 6 and Chapter 7 for more information about specifying an end-of-session function.

The following processing is done by the release operation:

- Source program
  - If the program device is invited, the release operation fails.
  - If a transaction is still active on the session, the release operation fails.
  - If a transaction is not active on the session, the session ends.
  - If the device description associated with the program device is allocated when the program device is acquired, it is deallocated when the program device is released.
  - If an error occurs due to the hardware or an SNA protocol violation, the release operation fails.
- Target program
  - The release operation severs the logical connection between the application and the requesting program device. The session is not ended.
  - The program (or another program in the same job structure) can reestablish the connection to the same session by acquiring the requesting program device. The communications session, including the state of the session, remains intact.

Closing an Intersystem Communications Function File

The processing done by the close operation depends on whether the file is shared. If the file is not shared, the following processing is done:

- All sessions associated with the source program are ended.
- All sessions associated with the target program are released.
- The file resources allocated by the open operation are deallocated and returned to the system.

If the file is shared, the program cannot do I/O operations to the file. Other programs that have the file open can still use the file.

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

Summary

Figure 5-6 shows the relationship between the program, the ICF file, and the communications configuration on a local and remote AS/400 system.

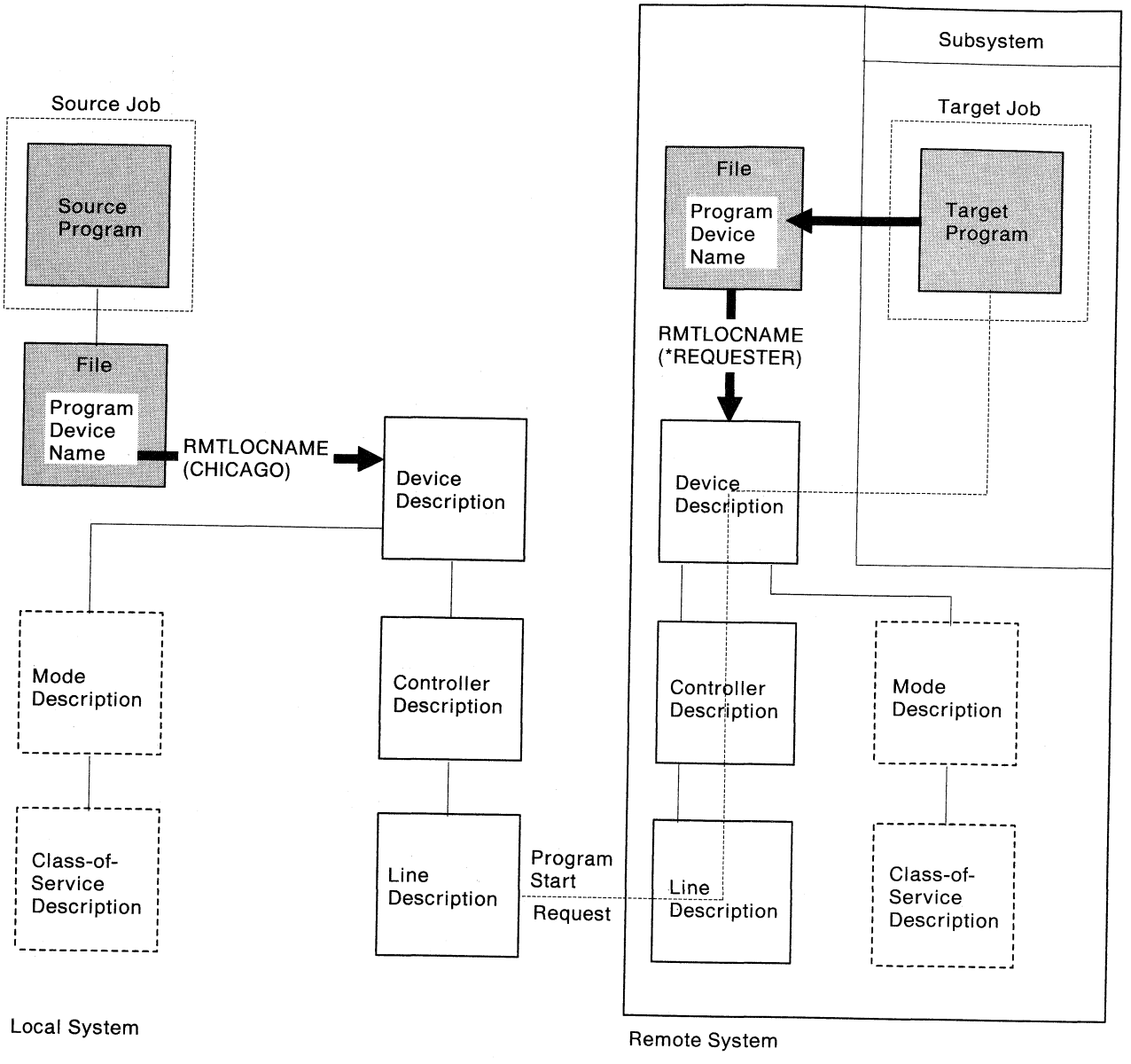


Figure 5-6. Relationship of Program, File, and Configuration

Note: The mode description and class-of-service description apply to APPC only.

Local System

The source program is your C/400, COBOL/400, or RPG/400 application communicating with the target program through an ICF file. You can create this file using the CRTICFF command, and change it using the CHGICFF or OVRICFF command.

The source program:

- Opens the file
- Acquires one or more program devices
- Reads and writes to program devices in the file to receive and send data
- Releases the acquired program devices
- Closes the file

The device entries defined in the file with the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command provide:

- Mapping to the communications configurations
- Communications-type-dependent definitions of program device attributes

The local configurations selected by the program device entry define the connection to the remote system.

Remote System

An incoming program start request from the local system starts a target job.

The target program is your C/400, COBOL/400, or RPG/400 application, which is communicating with the source program through an ICF file. You can create this file using the CRTICFF command and change it using the CHGICFF or OVRICFF command.

The target program:

- Opens the file
- Acquires the requesting program device
- Reads and writes to the requesting program device for the file to receive and send data
- Releases the requesting program device
- Closes the file

The device entries defined in the file with the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command provide:

- A relationship to the requesting program device
- Communications-type-dependent definitions of program device attributes

Refer to "Remote Program Start Considerations" on page 8-6 for more information on subsystems.

Figure 5-7 on page 5-24 shows how the program, file, and configuration names are mapped to each other in C/400.

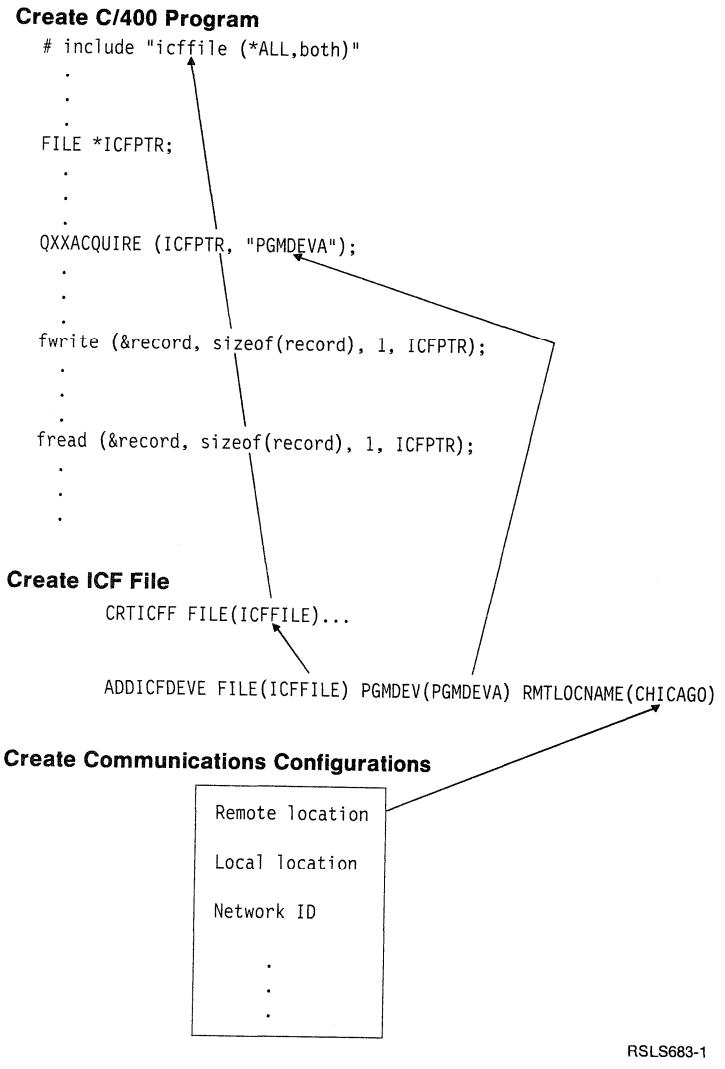


Figure 5-7. C/400 Program, File, and Configuration Mapping

You create various configuration objects when you use the communications configuration function. The remote location name provides the primary mapping between the program device and the communications configurations. The specified remote location name is used to select the device description.

Figure 5-8 on page 5-25 shows how the program, file, and configuration names are mapped to each other in COBOL/400.

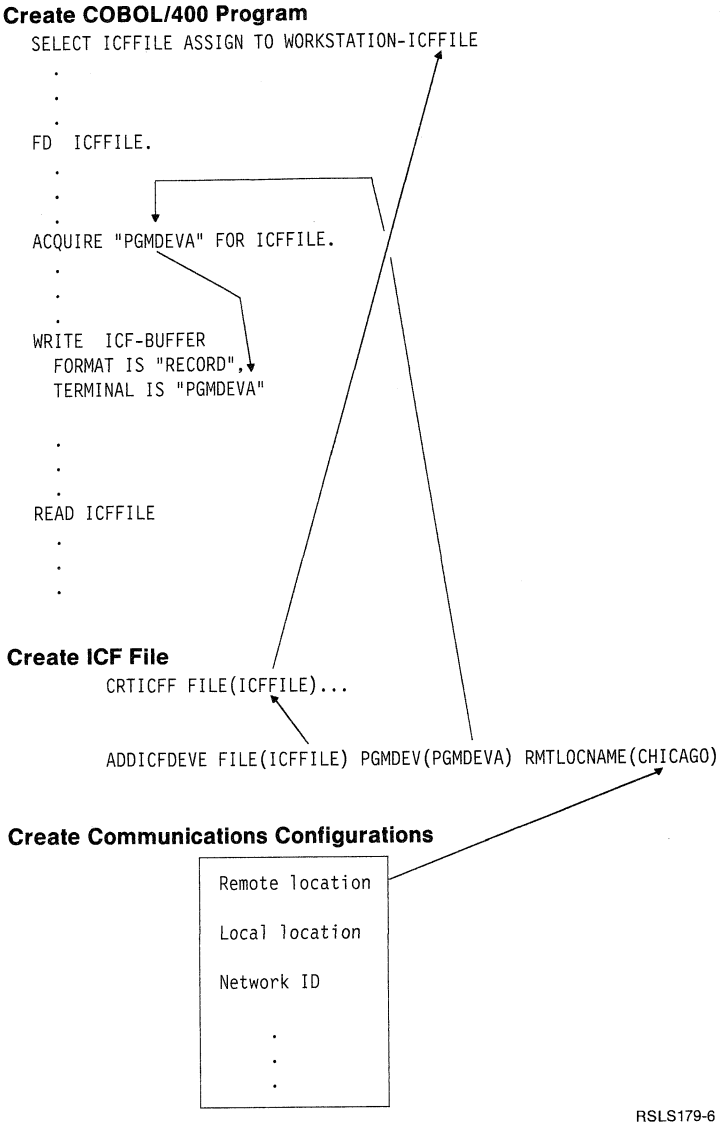


Figure 5-8. COBOL/400 Program, File, and Configuration Mapping

You create various configuration objects when you use the communications configuration function. The remote location name provides the primary mapping between the program device and the communications configurations. The specified remote location name is used to select the device description.

Figure 5-9 on page 5-26 shows how the program, file and configuration names are mapped to each other in RPG/400.

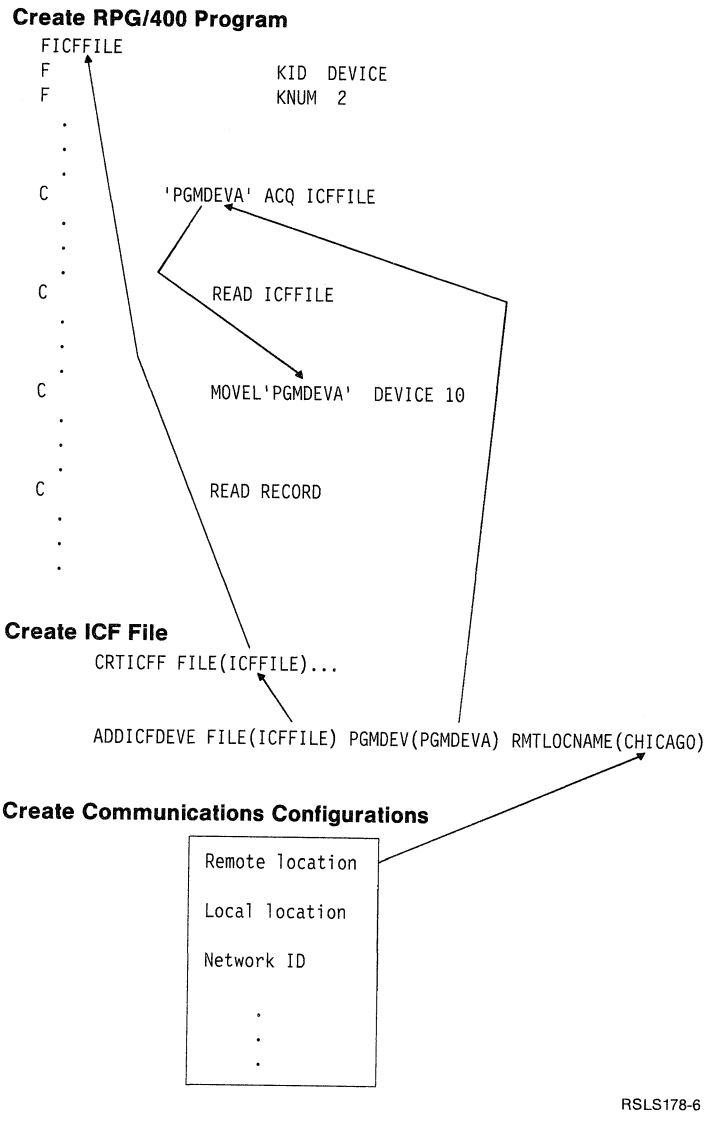


Figure 5-9. RPG/400 Program, File, and Configuration Mapping

You create various configuration objects when you use the communications configuration function. The remote location name provides the primary mapping between the program device and the communications configurations. The specified remote location name is used to select the device description.

Chapter 6. Using Communications DDS Keywords

This chapter explains how to use data description specifications (DDS) keywords on input and output operations to perform communications functions with the remote system. These DDS keywords are associated with the defined record format used on the read or write operation. The record formats associated with the DDS source for your ICF file are referred to as user-defined formats. This is in contrast to the system-supplied formats discussed in Chapter 7. It is assumed that you have opened your file and established your session as described in Chapter 5.

The information and illustrations provided describe the function of each of the processing keywords supported by the ICF file. Although all of the parameters supported by each keyword are described, the information on coding the keywords is found in Chapter 6 of the *DDS Reference*. The *DDS Reference* also contains general information on defining record formats.

You can use several DDS keywords and combinations of keywords on a single input/output (I/O) operation. Figure 6-15 on page 6-28 shows the processing sequence when multiple DDS keywords are specified together.

All the keywords described in this chapter may not be supported by the communications type you are using. Furthermore, some keywords may operate differently depending on the communications type. Table 6-1 on page 6-26 and Table 6-2 on page 6-27 summarize the support provided by each communications type. Refer to the appropriate communications programming manual for the communications type you are using for more detail about supported keywords.

Several DDS keywords that do processing-control, referencing, and text-definition functions that are valid in ICF files and other types of files are not discussed in this manual. These keywords are ALIAS, FLTPCN, INDARA, INDTXT, REF, REFFLD, and TEXT. Refer to the *DDS Reference* for more information on how to code and use these keywords. These keywords are supported by all communications types.

Examples of source DDS and the commands used to create and use an ICF file are found at the end of this chapter.

Refer to Chapter 10 for complete program examples that use DDS keyword processing.

You can use system-supplied communications formats instead of DDS keywords to do communications-specific functions. Refer to Chapter 7 for more information on system-supplied formats.

Starting a Program on the Remote System

Your program must specify the target program it will communicate with before it can send or receive data. The target program is started by specifying an output operation with the EVOKE keyword in effect. Generally, the necessary parameters to identify the target program you want to start must be specified. However, for some communications types, these parameters are not required.

These parameters include items such as the program name, the name of the remote library where the program is stored, and security information (when required). You

may also include data with the evoke function, which will be sent to the target program when the evoke function is done. When your program issues the evoke function, a program start request is sent to the remote system.

Use the EVOKE, SECURITY, and SYNLVL keywords to start a program at the remote system.

Evoke (EVOKE, SECURITY, and SYNLVL)

The EVOKE keyword allows your program to start a program on the remote system. EVOKE is valid only when the source program is not already communicating with the target program on the same transaction.

The format of the EVOKE keyword is:

```
EVOKE([library-name/]program-name [parameter-1... [parameter-255]])
```

The program-name parameter is required on the EVOKE DDS keyword to identify the program to be started on the remote system. However, some communications types do not require the program name. In these cases, blanks should be used for the program name instead. Refer to the appropriate communications manual to determine if the communications type you are using requires a program name on the EVOKE DDS keyword.

The optional library-name parameter specifies the library where the program is stored on the remote system. If the remote system is an AS/400 system and a library name is not given, the library list for the subsystem that is handling the request on the remote system is searched for the program name. The library list for the subsystem consists of the values from the QSYSLIBL and QUSRLIBL system values at the time the subsystem was started.

In addition to passing the program-name and library-name to the remote system, you can also use the EVOKE DDS keyword to send up to 255 user-defined parameters to the remote system. (Some communications types do not support 255 parameters. Refer to the appropriate communications programming manual for any additional restrictions.) The target program defines the number and format of the parameters. If the remote system is another AS/400 system, the following apply:

- The parameters are passed to the program as if they were passed from a Call a Program (CALL) command.
- If the parameters contain embedded commas, the remote AS/400 system considers these to be multiple parameters rather than a single parameter.

Any transaction status information sent by the source program is received on the first read operation of the target program. For example, if the target AS/400 system program is started from an AS/400 system with an evoke-with-invite function using advanced program-to-program communications (APPC), the first read operation on the target program completes with an 0300 (change direction received) major/minor return code.

You can use the SECURITY DDS keyword to include security information with the evoke request. The SECURITY keyword is only valid in conjunction with the EVOKE keyword. All security specifications must satisfy the requirements of the remote system.

The format of the SECURITY keyword is:

```
SECURITY(n reserved-word|'literal'|field-name-1|&field-name-1[.3.])
```

The n parameter required by the remote system identifies the security subfield being described. The n parameter can be specified as:

- 1 for a profile ID
- 2 for a password
- 3 for a user ID

You can specify the following values for the security fields:

reserved-word. This value can be specified as one of the following:

- *USER. Specifies that the user's profile name on the local AS/400 system is used as the security field.
- *NONE. Specifies that the security field is not supplied.

'literal'. A literal value of up to 10 characters that contains the needed security information.

field-name (or &field-name). The name of a field in the record format that contains the needed security information. If you want to send blanks as the security field, you must specify this as a literal value or use a field name.

If you do not explicitly define the security values on the SECURITY keyword for an evoke request, no security values are sent.

Refer to Chapter 8 for information about remote program start considerations on the AS/400 system.

Use the SYNLVL DDS keyword to specify the level of synchronization supported on this transaction. The SYNLVL keyword is valid only in conjunction with the EVOKE keyword.

The format of the SYNLVL keyword is:

```
SYNLVL[(*NONE|*CONFIRM)]
```

You can specify the following optional values for the SYNLVL keyword:

*NONE. Specifies that confirmation of the receipt of data is not allowed on this transaction. For example, on the AS/400 system the CONFIRM keyword is not allowed.

*CONFIRM. Specifies that the sending program can request that the receiving program responds to receipt of the data. The receiving program can send a positive response, or the receiving program or system can send a negative response. For example, on the AS/400 system the CONFIRM keyword is allowed on write operations.

Refer to the keyword descriptions for "Confirm (CONFIRM)" on page 6-5, "Receive-Confirm" on page 6-21, and "Respond-to-Confirm (RSPCONFIRM)" on page 6-14 for additional information on CONFIRM processing.

Figure 6-1 on page 6-4 shows how to start a target program on the remote system.

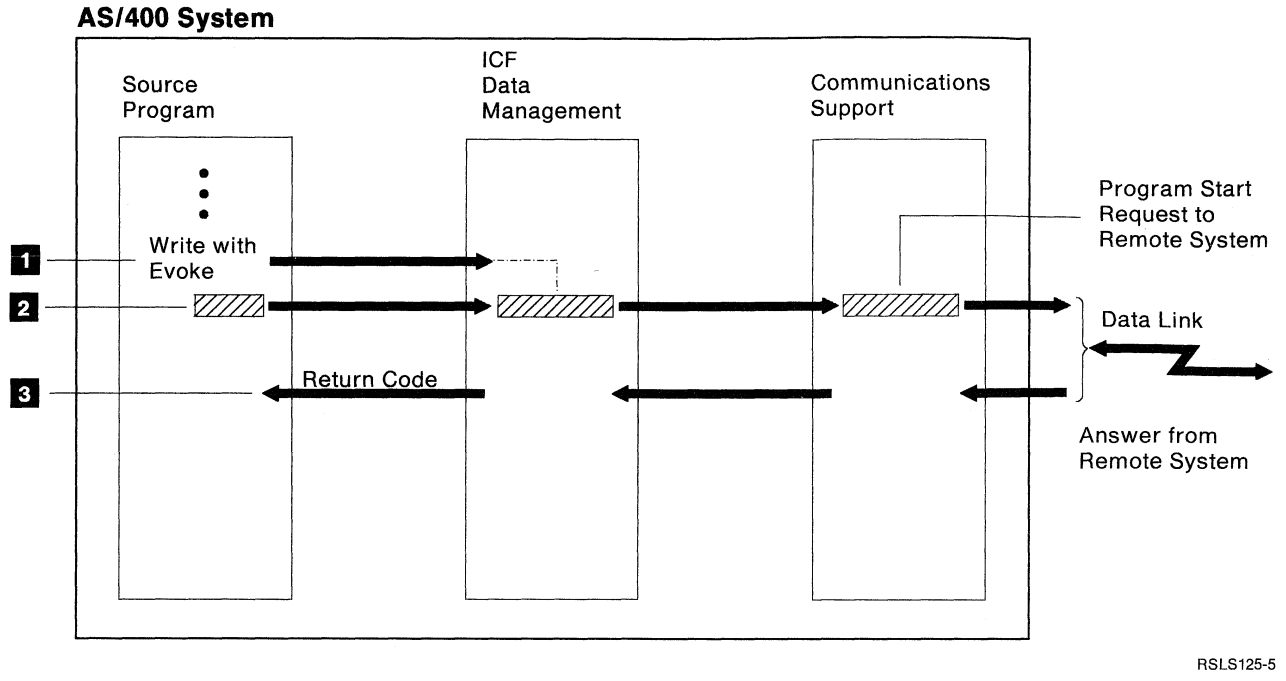


Figure 6-1. Starting a Target Program

1. The source program issues an evoke request to start the program at the remote system.
2. The evoke parameters, including program name, library name, and security information are sent to the remote system. Program initialization parameters can also be sent with the program start request (optional).
3. A successful completion return code tells the source program that the evoke request was accepted and a program start request was sent to the remote system. If the program start request is successful, both the program at the remote system and the communications transaction are started.

Sending Data

You may begin sending and receiving data when both systems are communicating with each other. This section discusses sending data. See "Receiving Data" on page 6-8 for a discussion on receiving data.

You can use several DDS keywords and combinations of keywords in conjunction with sending data. These keywords provide additional information about how to process the data being sent to the remote program.

The following are valid functions that can be done when sending data. The DDS keywords associated with these functions are valid only with output operations.

Variable-Length Data (VARLEN)

The length of an output operation is determined by the record format specified. The record format length is determined by the record definition in DDS. You can use the VARLEN keyword to change the length of the data record sent with each write operation, while using the same record format.

The format of the VARLEN keyword is:

```
VARLEN(&field-name)
```

The field-name parameter specifies the length of the record sent on a write operation. The length cannot be greater than the length of the data field defined for this record format. The length you specify with the VARLEN keyword overrides any length specified elsewhere in your write operation.

Force-Data (FRCDTA)

Use the FRCDTA keyword on a write request to cause the communications support to immediately send any data currently held in the output buffer. The communications support does not wait for the buffer to fill. Any data specified on the same operation as the force-data request is also sent. No operation is done if there is no data in the buffer to send.

Note: This causes the data to be sent to the other system, but not necessarily to the remote program.

Confirm (CONFIRM)

Use the CONFIRM keyword to request that the remote program respond when it has received the data you sent. An output operation with the CONFIRM keyword specified forces any data in the output buffer to be sent. The CONFIRM keyword also asks the remote program to respond when the data is received. The operation does not complete, and your program does not continue, until a response is received. The remote program must respond with either a positive or negative reply as to whether the data was successfully received.

Note: Refer to the RCVCONFIRM keyword described in "Using Response Indicators" on page 6-21 and the RSPCONFIRM keyword described in "Additional Keywords" on page 6-14 for information on how to receive and respond to a confirm request.

If a positive response is received, the output operation completes normally. If a negative response is received, the major/minor return code and ICF message indicate the reason.

CONFIRM is valid only on a transaction with a synchronization level of confirm.

Format-Name (FMTNAME)

Use the FMTNAME keyword to pass the name of the record format used for this output operation to the remote system. If the remote system is an AS/400 system, ICF uses this name to find the record format to use when receiving the data at the remote system.

Note: If you use the FMTNAME keyword while sending data to another AS/400 system, you should specify *RMTFMT for the format selection (FMTSLT) parameter on the Add Intersystem Communications Function Device Entry (ADDICFDEVE), Change Intersystem Communications Function Device Entry

(CHGICFDEVE), or Override Intersystem Communications Function Device Entry (OVRICFDEVE) command at the system at which the data is received.

Subdevice-Selection (SUBDEV)

Use the SUBDEV keyword to specify the remote system device (such as a printer or diskette) to which you are sending data. The receiving controller then directs output from your program to the appropriate device. The subdevice selection is designed primarily to support specific hardware devices, such as 3776, 3777, and 3780.

The format of the SUBDEV keyword is:

```
SUBDEV(type)
```

The type parameter values, *DC1, *DC2, *DC3, and *DC4, are required to specify the device control character used by the receiving controller so that output can be directed to the appropriate device.

End-of-Group (ENDGRP)

Use the ENDGRP keyword to indicate to the remote system the end of a user-defined group of records. The communications type you are using determines the type of indication sent to the remote system to indicate the end of a group of records.

Note: Refer to the RCVENDGRP keyword described in "Using Response Indicators" on page 6-21 for information on how to handle receiving an end-of-group indication.

Function-Management-Header (FMH)

Use the function-management-header (FMH) keyword to send control information about the data that follows to the remote system. A function-management-header is valid only with the first record of a group.

Note: Refer to the RCVFMH keyword described in "Using Response Indicators" on page 6-21 for information on how to handle receiving a function-management-header.

Examples of Sending Data

Figure 6-2 on page 6-7 shows how to use the CONFIRM, FRCDTA, and FMTNAME keywords when sending data.

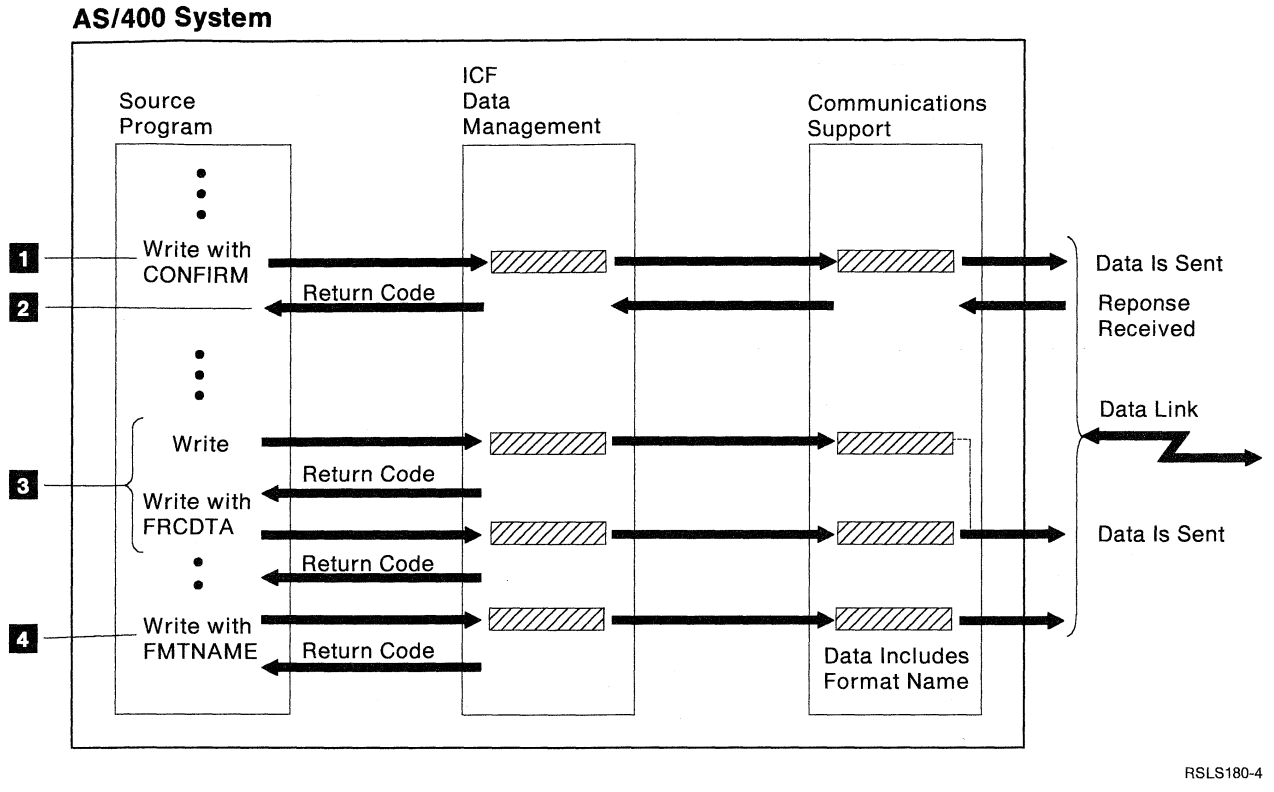


Figure 6-2. Using the CONFIRM, FRCDTA, and FMTNAME Keywords to Send Data

1. Your program issues a write-with-confirm operation to send data to the remote system and asks the remote system for a response.
2. Your program cannot continue processing until a response is received from the remote system. Your program checks the return code to determine if the remote system issued a positive or negative response.
3. If a successful return code is received, your program continues sending several records. On the last record, your program also specifies the force-data function. The FRCDTA causes all buffered data to be sent. The return code indicates the data is successfully sent. The force-data request does not wait for a response from the remote system.
4. Your program sends a record. The format-name function indicates that the record format name used on this write is also sent to the remote system. The remote system uses this record format when receiving the data.

Figure 6-3 on page 6-8 shows how to use the ENDGRP and FMH keywords when sending data.

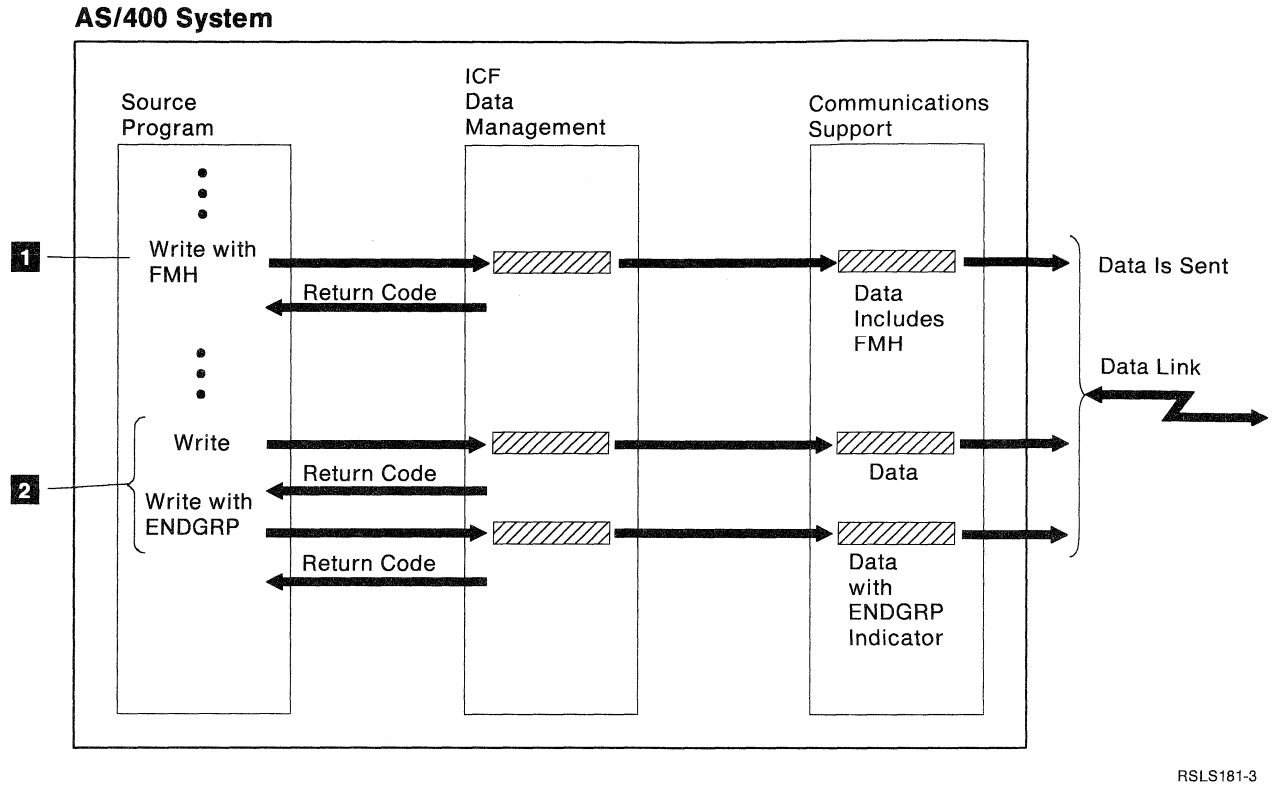


Figure 6-3. Using the ENDGRP and FMH Keywords to Send Data

1 Your program sends a record to the remote system and, with the FMH keyword, indicates that the first part of the data is function-management-header data, which contains information about the user data that follows.

2 Your program continues sending data records to the remote system. Your program uses the end-of-group function on the last record to indicate it is the last in this group of records.

Receiving Data

You can use two operations to receive data: read and read-from-invited-program-devices. In addition, you can use the invite, timer, and record-identification functions with the preceding operations to provide additional functions when receiving data.

The read operation receives data from the program device you specify. This operation differs from the read-from-invited-program-devices operation, which receives data from any program device with a previously issued invite request.

Invite (INVITE)

The INVITE keyword prepares your program to receive data. You must do an output operation with the INVITE keyword specified to issue an invite function. You can combine additional output keywords 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.

The read-from invited-program-devices operation is a companion to the invite function. After issuing an invite function, you use the read-from-invited-program-devices operation to receive the data from the remote system.

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

Refer to Chapter 5 for additional information about the read and read-from-invited-program-devices operations and their relationship to the invite function.

Timer (TIMER)

Your program can use the timer function before performing some specified function, such as a read-from-invited-program-devices operation. The timer function specifies an interval of time (in hours, minutes, and seconds) to wait before your program receives a timer-expired (0310) return code.

Use the TIMER keyword to set the timer for the specified interval of time. The TIMER keyword is issued on an output operation.

The format of the TIMER keyword is:

```
TIMER(HHMMSS | &field-name)
```

The parameter specified with the TIMER keyword can be one of the following:

HHMMSS A literal value where HH is the number of hours, MM is minutes, and SS is seconds.

&field-name A value where the field contains the TIMER value in the same HHMMSS format.

Your program continues to run, and all operations and functions are valid during the time interval. Your program must issue a read-from-invited-program-devices operation some time after it has issued the timer function, so it can accept the return code indicating that the timer interval has ended.

Only one time interval can be maintained for your program. If a previous timer function has been issued and the timer has not yet ended, the old time interval is replaced by the new interval.

The timer function can be used to vary the maximum amount of time that a read-from-invited-program-devices operation will wait for a response. When the time interval set by the TIMER keyword is in effect, the value specified for the WAITRCD parameter on the CRTICFF command is ignored.

You can use the timer function to retry other operations that may not be successful, possibly because of a temporary lack of resources (for example, during an acquire operation). To do this, issue the timer function, and then perform read-from-invited-program-devices operations until the timer interval ends. (The read-from-invited-program-devices operation allows the program to continue receiving input from other invited program devices while waiting for the timer.)

Refer to Chapter 5 for additional information on the read-from-invited-program-devices operation and its relationship to the timer function.

Record-Identification (RECID)

The RECID keyword identifies and selects the record format to use with an input operation based on the data received from the remote program. This keyword is applicable only if you specify FMTSLT(*RECID) on the ADDICFDEVE or OVRICFDEVE command.

The format of the RECID keyword is:

```
RECID(starting-position compare-value)
```

Specify the starting-position parameter as either nnnnn or *POSnnnnn, where nnnnn defines the beginning position of the compare value within the record format. The first position in the record is position 1.

Specify the compare-value parameter as:

**ZERO*. The data character in the position specified must be 0 (hex F0) to match the record identifier.

**BLANK*. The data character in the position specified must be a blank (hex 40) to match the record identifier.

'literal'. The data received, beginning with the position defined by the starting-position value, must match the literal specified here.

If the length of the record received is less than the number of positions examined for RECID value, the positions past the end of the record are treated as if they contained blanks. If the RECID keyword compare value specifies blanks for those positions, the data is considered a match.

For example, if your program receives both header and detail records from the remote program, you can specify the following in your ICF file:

```
RECID(1 'H')
RECID(1 'D')
```

Your program issues input operations to the file without specifying a record format name. You do not specify a record format name because the correct record format is not known until the data is received. Your program receives the records (either headers or detail) in the order they are sent by the remote program. For this example, the sending and receiving programs provide for an explicit code (an H for header records and a D for detail records) to identify the type of record being sent and received. The RECID keyword identifies the input buffer location where the H or D appears, and specifies the value (starting in the position specified) that identifies the record type.

The remote program must identify the type of record (either header or detail) by placing H or D in the first position of the data buffer.

For each input operation, the value specified in the first position of the buffer is compared with the value specified on the RECID keyword. If the value in a record is H, the format associated with the RECID(1 'H') specification is selected. Duplicate RECID keyword compare values are not checked. The first format with a compare value that matches the received value is used.

Be careful how you specify more than one RECID keyword within a file if more than one compare value begins in the same record position. For example, the following compare values begin in the same position:

```
RECID(1 'A')
RECID(1 'AB')
RECID(1 'ABC')
```

The first format is always selected if the data starts with an A, because any received records matching the last two compare values also match the first. Specify the longest value first to prevent confusion.

```
RECID(1 'ABC')
RECID(1 'AB')
RECID(1 'A')
```

You can use the RECID DDS keyword to eliminate processing of alphameric data in fields that should contain only numeric data. Refer to "Input Considerations" on page 8-4 for more information on eliminating data decimal errors.

Problem Notification

Use the fail, cancel, and negative-response functions to inform the application program of an error that has occurred in the data being sent or received. The DDS keywords associated with these functions are specified on an output operation.

Fail (FAIL)

Use the FAIL keyword to indicate that an error has occurred when sending and receiving data.

If a program that is sending data issues a fail, it may indicate that the data just sent was in error. Your program can continue to send data and is usually responsible for the first error recovery. The communications type you are using determines whether data that is in the output buffer before issuing the fail function is sent to the remote system with the fail indication. The communications type determines the type of notification sent.

You can also use a fail function if your program receives data and detects an error in the received data. Figure 6-4 on page 6-12 shows how to use the fail function when your program is receiving data and detects an error.

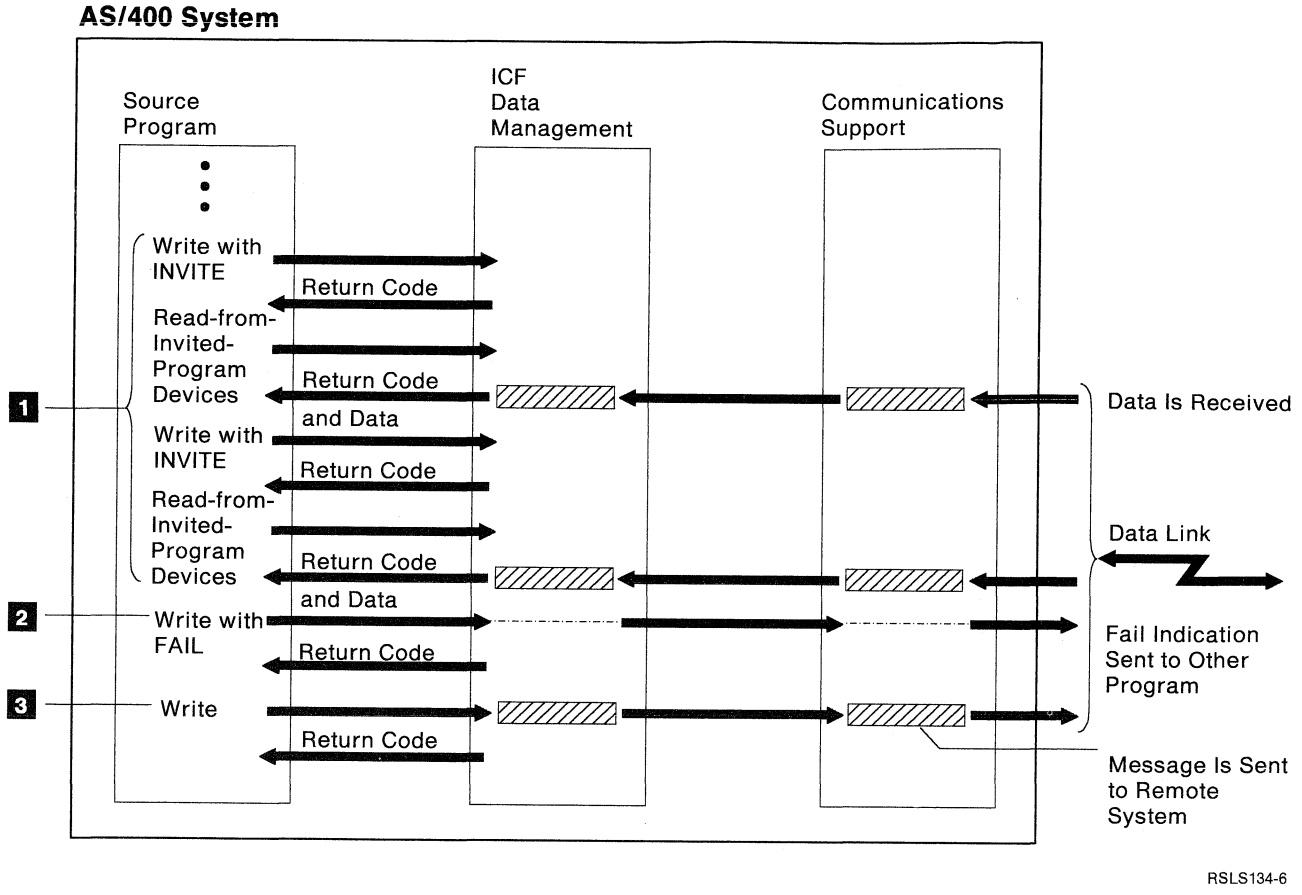


Figure 6-4. Using the FAIL Keyword to Send an Error Indication

1. Your program is receiving data from the remote program.
2. While receiving data, your program determines that it must send a fail indication to the other program.
3. A message or data is then sent (write operation) to tell the other program why you sent the fail indication.

Refer to the RCVFAIL keyword described in "Using Response Indicators" on page 6-21 for information on handling receipt of a fail indication.

Cancel (CANCEL)

Use the CANCEL keyword to tell the remote system to cancel the group of records you are currently sending. Your program can use the cancel function only when sending data (similar to issuing a fail when your program is sending data). Figure 6-5 on page 6-13 shows how to use the cancel function when a program is sending data and detects an error.

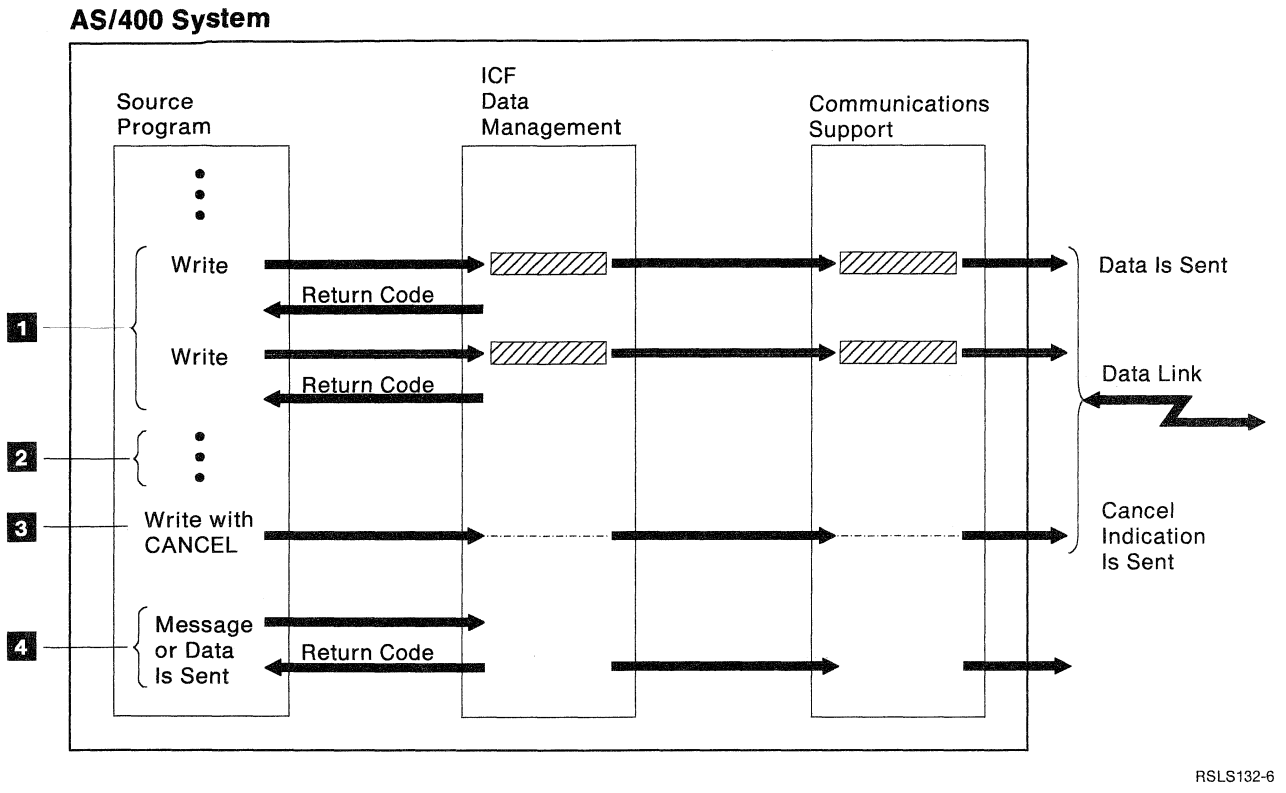


Figure 6-5. Using the CANCEL Keyword to Send an Error Indication

1. Your program is sending data to the remote system.
2. Your program checks the data and determines that something is wrong with it.
3. Your program uses a cancel function to tell the remote system to discard the data you have sent.
4. Your program can send a message indicating the problem, send the data again, receive more data, or end the transaction.

Refer to the RCVCANCEL keyword described in "Using Response Indicators" on page 6-21 for information on handling receipt of a cancel indication.

Negative-Response (NEGRSP)

Use the NEGRSP keyword to tell the remote system that the data just received is not correct. The format of the NEGRSP keyword is:

```
NEGRSP[(&field-name)]
```

The optional parameter on the NEGRSP keyword specifies the name of the field that contains sense data to be sent to the remote program with the negative response.

Issuing a negative-response function is similar to issuing a fail function while receiving data, except that you can also include 8 characters of sense data with the negative-response function. The sense data tells the remote system what is wrong with the data you received. The first 4 characters of the sense data must begin with 10XX, 08XX, or 0000. The last 4 characters are user-defined. Refer to the appropriate communications programming manual for the communications type you are using for more information about the allowed sense values.

The sense data is sent in the normal output buffer. No other data is allowed to be sent with a negative-response function.

Figure 6-6 shows how to send a negative response with a sense code to the remote system.

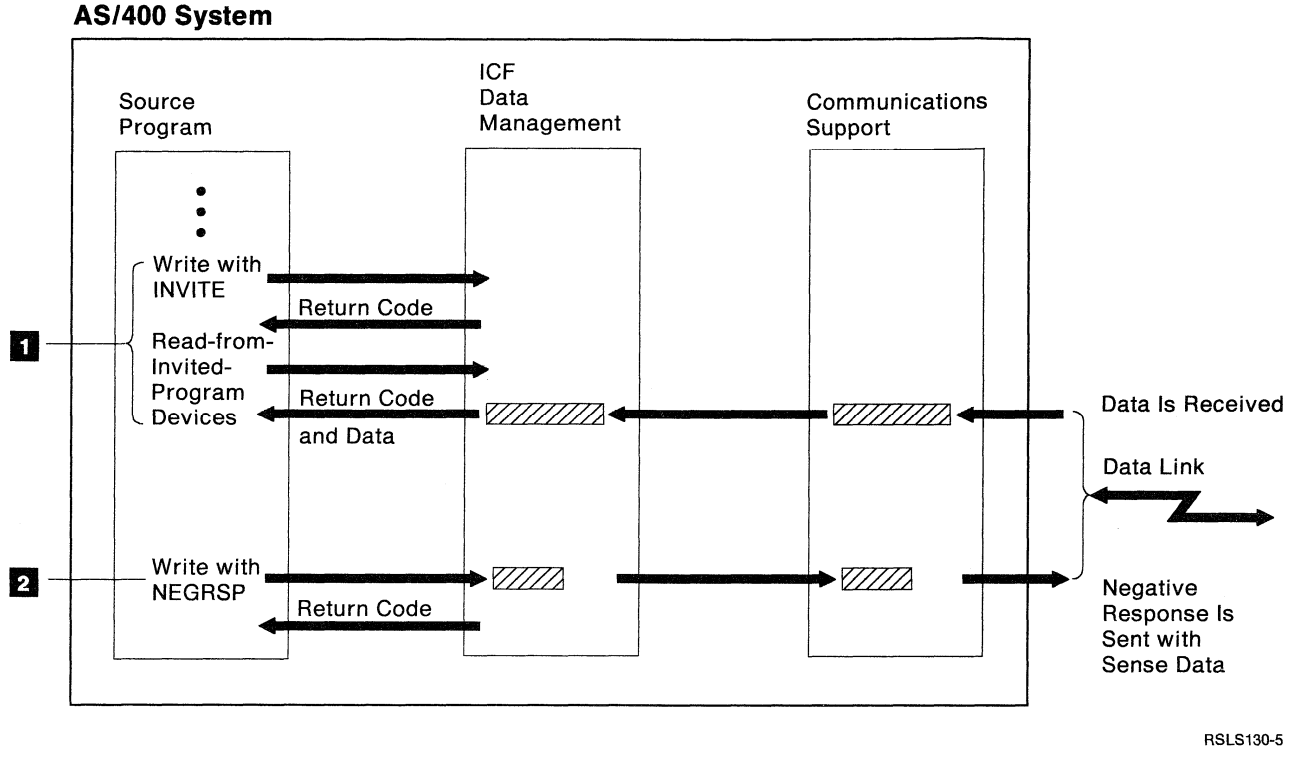


Figure 6-6. Sending a Negative Response with Sense Code to Remote System

1. Your program finds that the data it is receiving is not correct.
2. The program sends a negative response to the remote system, including the sense data 08110000. The negative response tells the remote system that the data received is wrong, and the sense data 08110000 asks the remote system to cancel the current group of data records.

Refer to the RCVNEGRSP keyword described in "Using Response Indicators" on page 6-21 for information on handling receipt of a negative response indication.

Additional Keywords

You can use the respond-to-confirm, request-to-write, allow-write, and cancel-invite functions to perform additional functions.

Respond-to-Confirm (RSPCONFIRM)

Use the RSPCONFIRM keyword to send a positive response to a received confirm request. The respond-to-confirm function can be used only when a confirm request is outstanding. You can check the major/minor return codes or use the RCVCONFIRM indicator to determine when to issue a respond-to-confirm function. After sending the response, your program can continue processing as indicated by any other information received.

Figure 6-7 shows how to use the respond-to-confirm function.

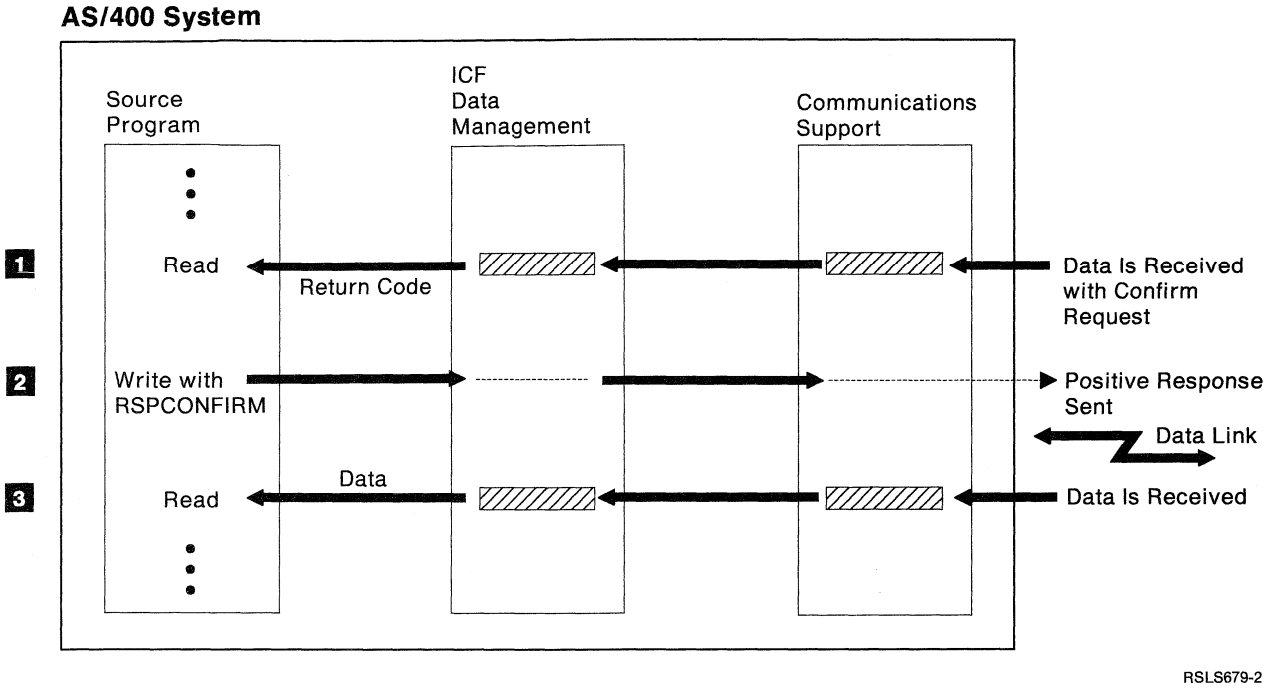


Figure 6-7. Using the Respond-to-Confirm Function

1. Your program is receiving data from the remote system. The return code indicates data and a confirm request. The read could have also been done with the RCVCONFIRM keyword to indicate that a confirm request was received.
2. Your program issues a write operation with the RSPCONFIRM keyword in effect to acknowledge the receipt of data.
3. Your program continues to receive because the remote program is still in send state.

Request-to-Write (RQSWRT)

Use the RQSWRT keyword, while your program is receiving data, to ask the remote system to stop sending so your program can send. The request-to-write function tells the remote system you want to change the direction of data transmission. If the remote system allows the change, your program can send either data or a message, or both, to the remote system. After issuing the request-to-write, your program must continue receiving data until the remote system sends a notification indicating it is ready to receive.

Figure 6-8 on page 6-16 shows how to use the request-to-write function.

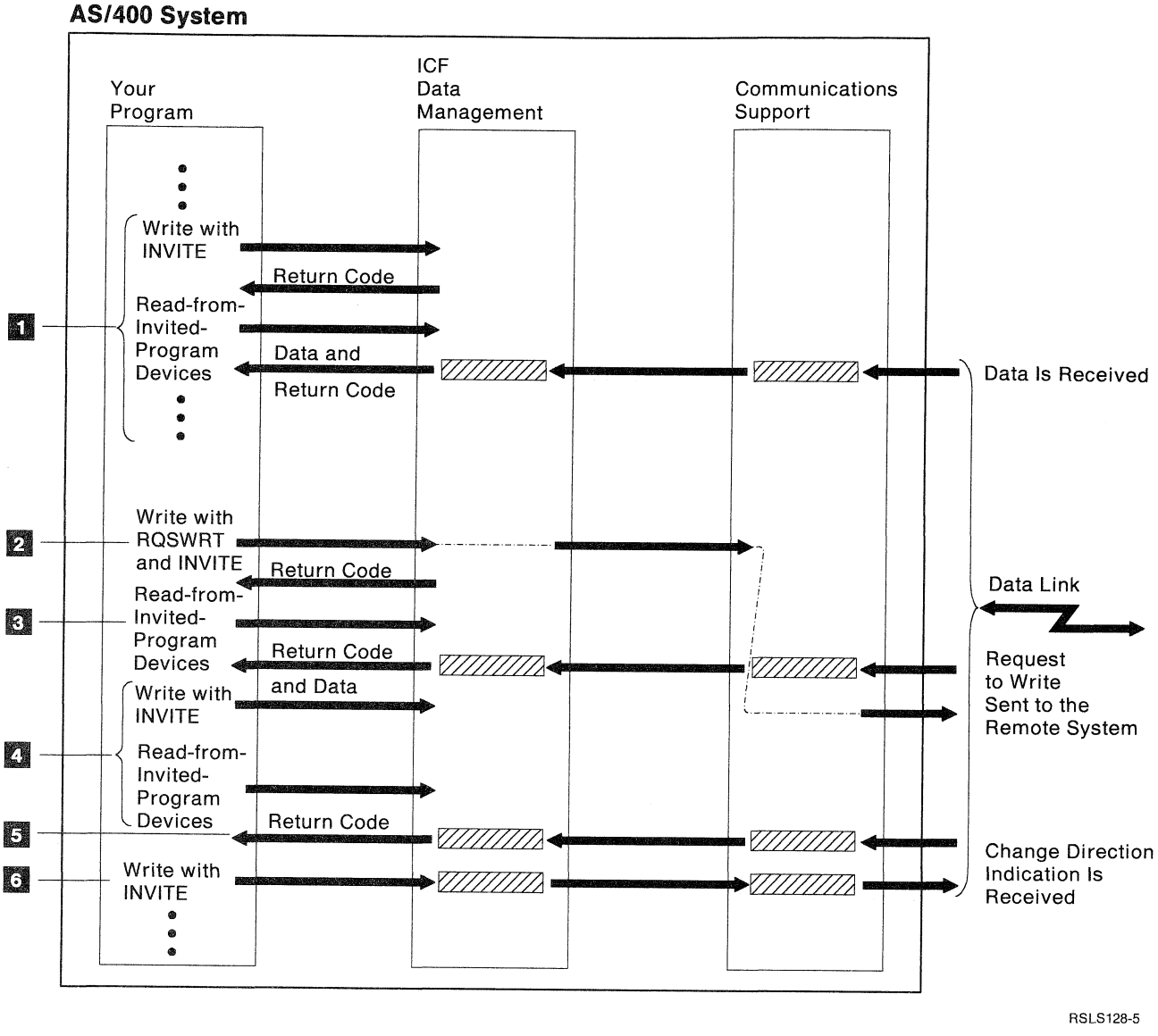


Figure 6-8. Using the Request-to-Write Function

1. Your program is receiving data from the remote system. The program processes the data received, then receives data again.
2. At some time while data is being received, your program determines that it needs to send a message to the remote system. Your program issues a write operation, with the RQSWRT and INVITE keywords in effect, to ask the remote system to stop sending so your program can send the message. The request-write indication is sent to the remote system at the first available opportunity. Since the session is in receive state, the indication may be held until the next data record is received.
3. After issuing the request-to-write function, your program must continue receiving data until it gets a return code indicating that the remote system is ready to receive. To continue receiving, another read-from-invited-program-devices operation is used.
4. Another invite and read-from-invited-program-devices operation is issued to continue receiving data.

5 When the remote system is ready to receive, it sends one more data record with a change-direction indication. The record says the remote system is now ready to receive data or, as in this example, a message.

6 A write operation with the INVITE keyword in effect is used to send the message to the remote system and ask the remote system to continue sending data.

When your program receives a request-to-write request from the remote system, a code is set in the I/O feedback file-dependent section. Refer to Table C-5 in Appendix C for more information about where this field is in the I/O feedback area.

The code indicates the following conditions:

0 Continue sending as normal.

1 A request-to-write has been received.

Refer to the appropriate communications programmer's manual for the communications type you are using for more specific information on what this code means for the communications type you are using.

Allow-Write (ALWWRT)

Use the ALWWRT keyword to explicitly inform the remote system that your program is done sending. The allow-write function clears the buffers, forcing any data to be sent. The same function occurs automatically if you issue an input operation after a write operation. In that case, the ALWWRT DDS keyword is not required. After issuing an allow-write, your application program can issue an input operation to receive data from the remote system.

Figure 6-9 shows how to use the allow-write function.

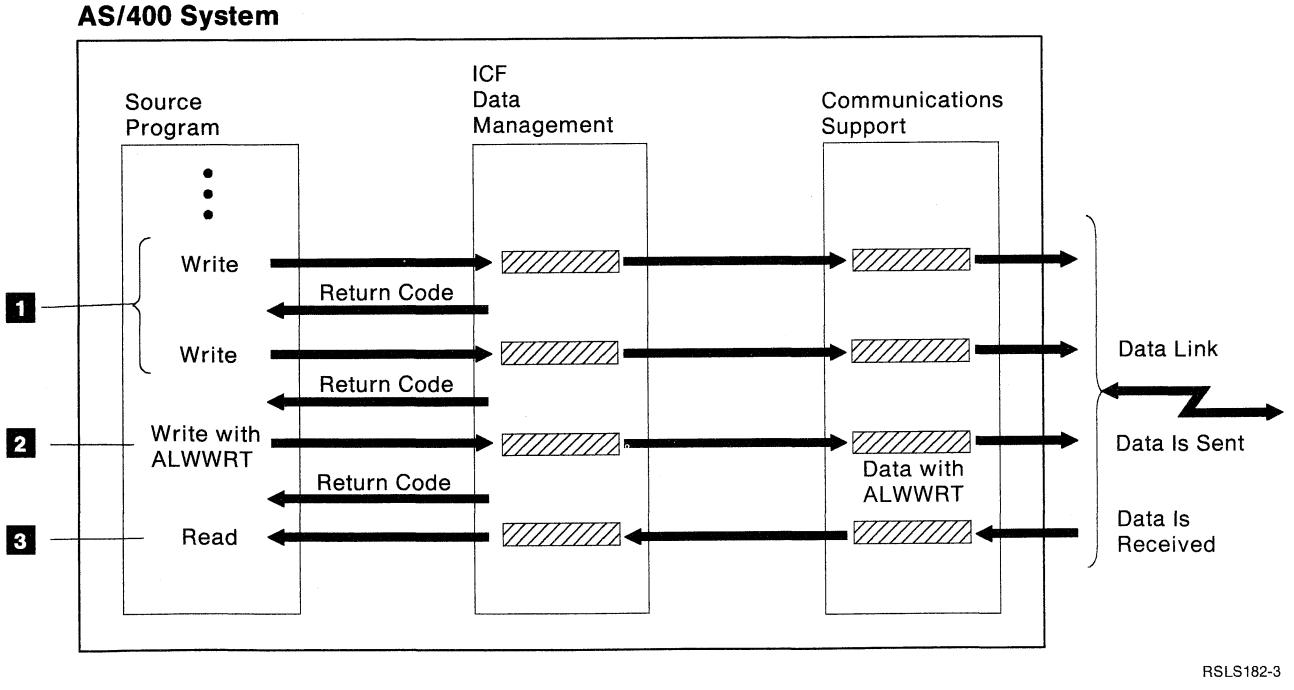


Figure 6-9. Using the Allow-Write Function

1. Your program sends several data records to the remote system.
2. You use an allow-write with the last record to inform the remote system you are done sending.
3. The remote system can now send data, and your program must begin receiving.

Refer to "Receive-Turnaround" on page 6-22 for information on handling receipt of an allow-write indication.

Cancel-Invite (CNLINVITE)

Use the CNLINVITE keyword to cancel a valid invite for which no data has yet been received from an invited program device. Your program can continue to send data.

Figure 6-10 shows how to use the CNLINVITE keyword.

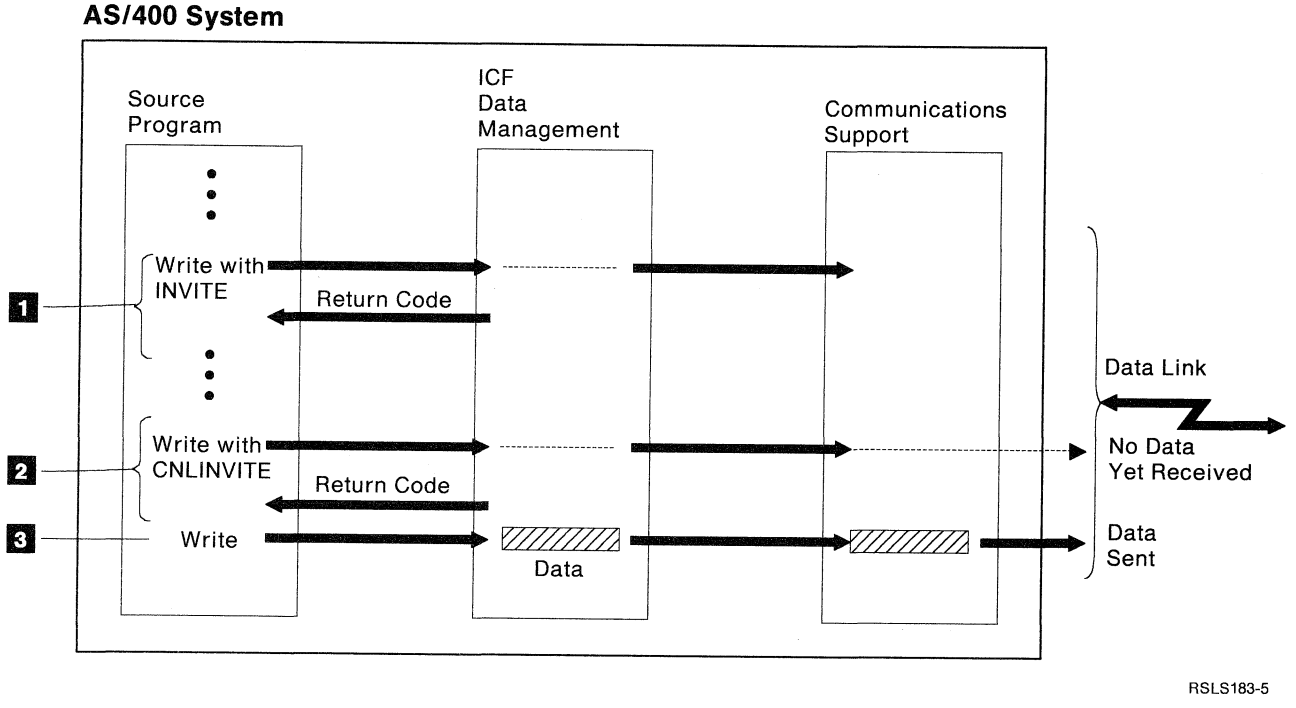


Figure 6-10. Using the Cancel-Invite Function

1. Your program issues an invite operation to receive data from the remote program, then continues processing.
2. Your program uses the cancel-invite function to cancel the previous invite operation. Your program must check the return code it receives to determine if any data has already been received from the remote system.
3. Your program can continue to send data if data was not received.

Ending a Communications Transaction

A communications transaction can be ended by your program or by the program at the remote system. Your job and the remote system that your system is communicating with determine the program that ends the transaction.

Communications with the remote program end when your program ends the transaction. However, the session may still exist if your program started the session. If the session still exists, you can end the session or you may be able to start another program at the remote system.

Detach (DETACH)

Use the DETACH DDS keyword to end the transaction. The detach explicitly informs the remote program that your program is done sending and has ended the transaction.

Figure 6-11 shows how your program can end a communications transaction.

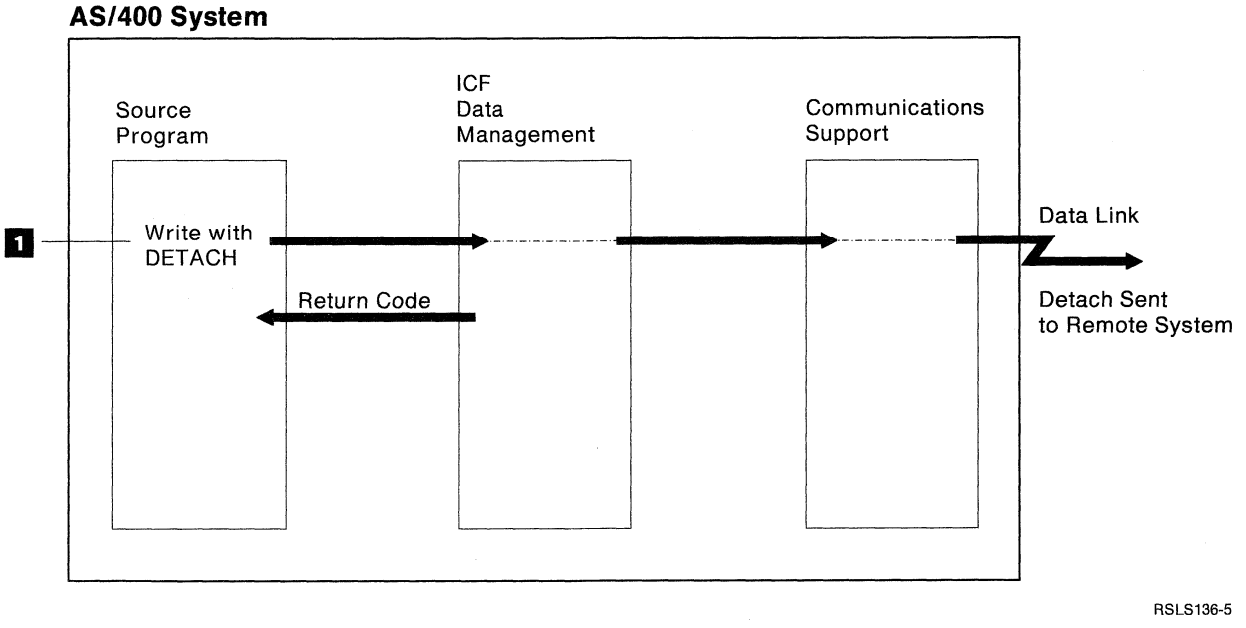


Figure 6-11. Ending a Communications Transaction

1 Your program issues the detach to tell the remote system that your program ended the communications transaction.

Refer to the RCVDETACH keyword described in "Using Response Indicators" on page 6-21 for information on handling receipt of a detach indication.

If a detach function is issued by the target program, the session is ended as well as the transaction.

Ending the Communications Session

How the communications session is ended depends on whether your program or the remote system started the session.

If your program started the session (source program), your program must end the session using either the release operation or the end-of-session function. You should primarily use the release operation. Use the end-of-session only when you want to force the session to end.

The release operation ends the session only if all processing is complete. The end-of-session operation *always* ends the session.

End-of-Session (EOS)

Use the EOS DDS keyword to issue an end-of-session function. The only possible return codes from end-of-session are 0000 or 830B (program device not acquired).

If the remote system started the session (target program), your program must issue an end-of-session or go to end of job in order to end the session. Figure 6-12 shows how you can end the session using the release operation and the end-of-session function.

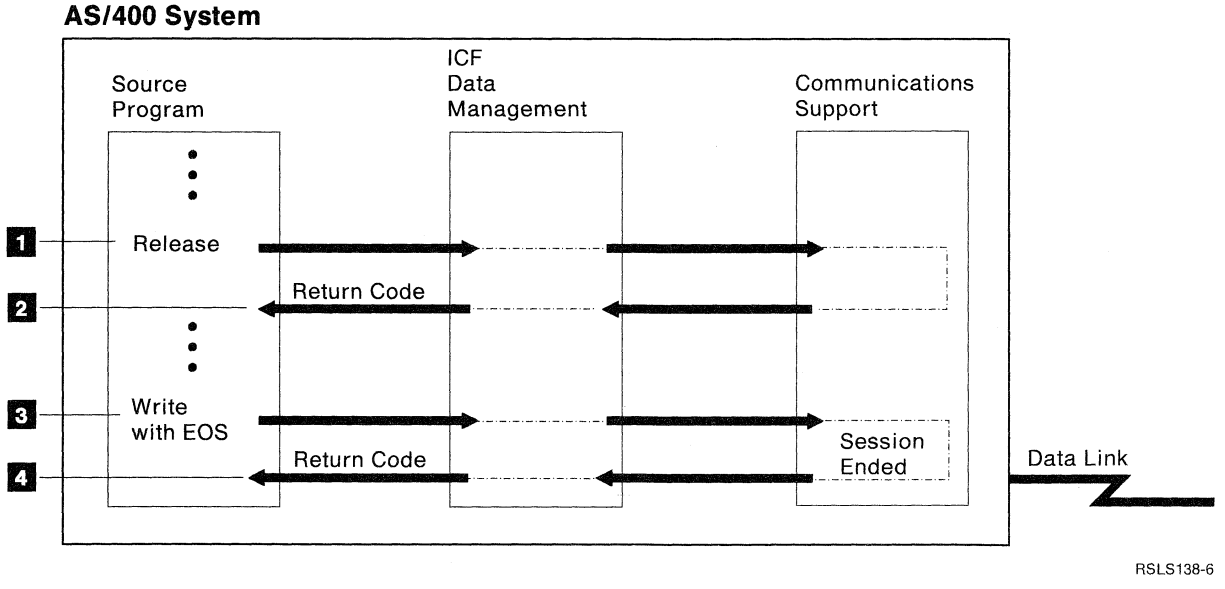


Figure 6-12. Using the Release and End-of-Session Functions

1. Your program issues the release operation to end the current communications session.
2. The return code tells your program whether the session was ended, or if an error occurred while trying to end the session. If, for example, all transactions have not ended when the release operation is issued, an error occurs and the session is not ended.
3. If an error occurs and normal recovery is not possible, your program can use the end-of-session function to end the session.
4. The end-of-session function always ends the session.

Using Response Indicators

Response keywords provide information to your program about the data record being received or the actions taken by the remote program. Check which response indicators are set when your program does an input operation to determine:

- What the remote program sent
- What the remote program expects from your program
- What your program's next operation should be

Response keywords are only effective for input operations or a combined output then input. They have no effect on an output operation. Multiple response keywords can be used on a single input operation.

Receive-Confirm

Use the RCVCONFIRM keyword to request that a response indicator be set on if the record received from the remote system contains a confirm request. A received confirm request indicates that the remote program expects your program to do a specific action to synchronize the programs. The action can be a write with the RSPCONFIRM keyword (positive response), or a write with the FAIL keyword (negative response). If the session is abnormally ended (end-of-session or job end before ending the transaction), a negative response is sent.

This same type of information can be determined by checking the major/minor return code returned in the I/O feedback area at the completion of each operation.

The program receiving the confirm indication is responsible for making sure that the response (positive or negative) is returned to the program requesting the confirmation.

If you want to return a positive response to the remote system, issue a write with the RSPCONFIRM DDS keyword in effect. If you want a negative response returned, either issue a write with the FAIL DDS keyword in effect, or abnormally end the session (end-of-session or job end before ending the transaction).

Receive-End-of-Group

Use the RCVENDGRP keyword to request that a response indicator be set on if the record received from the remote system contains an end-of-group indicator. The response indicator is set if the last record received in the input buffer was the end of a user-defined group of records.

Receive-Function-Management-Header

Use the RCVFMH keyword to request that a response indicator be set on if the record received from the remote system contains a function-management-header indication. The response indicator is set if the data received in the input buffer is function-management-header data. Asynchronous, finance, intrasystem, and retail communications give the user data along with the function-management-header indication in one operation. If you are using SNUF, however, you must do an additional input operation to get the remaining user data that accompanied the function-management-header. (APPC and BSCEL do not support the function-management-header).

Receive-Fail

Use the RCVFAIL keyword to request that a response indicator be set on if the record received from the remote system contains a fail indication. The remote program informs your program that it found something wrong while sending or receiving data. Your program should issue an input operation after receiving a fail indication. The program sending the fail indication must start the error recovery.

Receive-Cancel

Use the RCVCANCEL keyword to request that a response indicator be set on if the record received from the remote system contains a cancel indication. The remote system informs your program that the current chain of data is not correct. Your program should discard the data, then continue to receive or end the job.

Receive-Negative-Response

Use the RCVNEGRSP keyword to request that a response indicator be set on if the data received from the remote system contains a negative-response indication. The remote system informs your program that an error was detected in the data it just received. You may receive an 8-byte sense code with the negative response signal.

Receive-Turnaround

Use the RCVTRNRND keyword to request that a response indicator be set on if the data received from the remote system contains a change-in-transmission-direction indication. The remote system informs your program that it is finished sending data, and is ready to receive data. Your program can begin sending data.

Receive-Detach

Use the RCVDETACH keyword to request that a response indicator be set on if the record received from the remote system contains a detach indication. The remote system informs your program that it is ending this communications transaction with your program. Your program can no longer communicate with the remote program. The session with the remote system may still exist if your program started the session. If the remote system started the session, communications with the remote system are ended.

Example DDS Files for Creating an Intersystem Communications Function File

The following examples are DDS source files that can be used to create an ICF file. Files created using this source DDS are used in the application program examples in Chapter 9 through Chapter 11.

```
A*************************************************************
A*                                                           *
A*                        ICF FILE                           *
A*              USED IN BATCH DATA TRANSFER PROGRAM          *
A*                                                           *
A*************************************************************
A*
A*  FILE LEVEL INDICATORS:
A*
A                                       INDARA
A*
A                                       RCVTRNRND(15 'END OF DATA')
A*
A  30                                   DETACH
A*
A                                       INDTXT(30 '30->DETACH TAR-
A                                       GET PROGRAM.')
A*
A                                       RCVDETACH(35 'RECEIVED -
A                                       DETACH.')
A*
A*
A*************************************************************
A*                   ICF RECORD FORMATS                      *
A*************************************************************
          R RCVDATA
            RCVFLD        80A
          R SNDDATA
            SNDFLD        80A
          R EVOKPGM
A  50                                   EVOKE(&LIB/&PGMID)
A           PGMID         10A  P
A           LIB           10A  P
A         R ENDREC
A         R INVITE
A  45                                   INVITE
```

Figure 6-13. DDS Source File for a Batch Data Transfer Program

```
300      A*****************************************************************
400      A*                                                               *
500      A*                          ICF FILE                             *
600      A*         USED IN SOURCE MULTIPLE SESSION PROGRAM               *
700      A*                                                               *
800      A*****************************************************************
900      A                                      INDARA
1000     A          R ITMRSP
1100     A                                      RECID(1 'I')
1200     A            RECITM         1
1300     A            ITEMNO         6  0
1400     A            DESC          30
1500     A            QTYLST         7  0
1600     A            QTYOH          7  0
1700     A            QTYOO          7  0
1800     A            QTYBO          7  0
1900     A            UNITQ          2
2000     A            PR01           7  2
2100     A            PR05           7  0
2200     A            UFRT           5  2
2300     A            SLSTM          9  2
2400     A            SLSTY         11  2
2500     A            CSTTM          9  2
2600     A            CSTTY         11  2
2700     A            PRO            5  2
2800     A            LOS            9  2
2900     A            FILL1         56
3000     A          R DTLRSP
3100     A                                      RECID(1 'C')
3200     A                                      RCVTRNRND(90)
3300     A            RECCUS         1
3400     A            CUSTNO         6  0
3500     A            DNAME         30
3600     A            DLSTOR         6  0
3700     A            DSLSTM         9  0
3800     A            DSPM01         9  0
3900     A            DSPM02         9  0
4000     A            DSPM03         9  0
4100     A            DSTTYD        11  0
4200     A            IDEPT          3  0
4300     A            FILL2         57
4400     A          R DETACH
4500     A                                      DETACH
4600     A          R EOS
4700     A                                      EOS
4800     A          R EVKREQ
4900     A                                      EVOKE(&LIB/&PGMID)
5000     A            PGMID         10A  P
5100     A            LIB           10A  P
5200     A          R ITMREQ
5300     A                                      INVITE
5400     A            ITEMNO         6  0
5500     A          R DTLREQ
5600     A                                      INVITE
5700     A            CUSTNO         6  0
                  **** END OF SOURCE ****
```

```
10/14/87
10/14/87
10/14/87
10/14/87
10/14/87
10/14/87
10/07/87
10/06/87
10/06/87
10/06/87
10/06/87
10/13/87
10/06/87
10/06/87
10/06/87
10/06/87
10/06/87
10/06/87
10/06/87
10/06/87
10/06/87
10/06/87
10/06/87
10/06/87
10/06/87
10/06/87
10/06/87
10/06/87
10/06/87
10/13/87
10/06/87
10/06/87
10/06/87
10/06/87
10/06/87
10/06/87
10/06/87
10/06/87
10/06/87
10/06/87
10/06/87
10/06/87
10/06/87
10/12/87
10/06/87
10/06/87
10/06/87
10/06/87
10/13/87
10/06/87
10/06/87
10/13/87
```

Figure 6-14. DDS Source File for a Multiple Session Program

The following is an example of a Create Intersystem Communications Function (CRTICFF) command used to create an ICF file from a DDS source file:

```
CRTICFF  FILE(ICFLIB/ICFFILE)  SRCFILE(ICFLIB/QDDSSRC)  SRCMBR(*FILE)
   ACQPGMDEV(*NONE)  MAXPGMDEV(10)  TEXT('ICF FILE EXAMPLE')
```

The created file has the following attributes:

- The file name is ICFFILE and it is stored in library ICFLIB, as specified on the FILE parameter.
- The SRCFILE parameter indicates that the DDS source file, from which this ICF file is created, is a member in file QDDSSRC in library ICFLIB.

- The SRCMBR parameter indicates the source file has the same name as the file you are creating.
- The ACQPGMDEV parameter indicates that no program device is automatically acquired when the file is opened. Your program must explicitly issue the acquire.
- The MAXPGMDEV parameter indicates that up to 10 program devices can be acquired and active with this file.
- The TEXT parameter describes the file.

The remaining parameters not specified on the CRTICFF command are assigned default values. Refer to Chapter 4 and the *CL Reference* for more information on these parameters and their default values.

The following is an example of an ADDICFDEVE command used to add a program device entry to the ICF file just created:

```
ADDICFDEVE  FILE(ICFLIB/ICFFILE)  PGMDEV(PGMDEVA)  RMTLOCNAME(CHICAGO)
   FMTSLT(*PGM)
```

The file now has a program device entry with the following attributes:

- The PGMDEV parameter indicates that PGMDEVA is the program device name added to the file. This is the program device name used in your program.
- The RMTLOCNAME parameter indicates that CHICAGO is the name of the remote location associated with PGMDEVA. CHICAGO is the remote location name specified on the device description when you configured your system for communications.
- The FMTSLT parameter indicates that *PGM is the format selection option used on input operations. For more information on this parameter, refer to "Format Selection Processing" on page 5-10.

The remaining parameters not specified on the ADDICFDEVE command have assigned default values. Refer to Chapter 4 and the *CL Reference* for more information on these parameters and their default values.

Keyword Processing Charts

Table 6-1 and Table 6-2 on page 6-27 summarize the DDS keywords discussed in this chapter. Use these charts for a quick reference when defining and creating an ICF file, and when writing application programs.

Table 6-1 lists the DDS keywords defined in this chapter that are supported by the various communications types for output operations.

Table 6-1. Output DDS Processing Keyword Support

DDS Keyword	APPC	SNUF	BSCEL	Asyn-chronous	Intra-system	Finance	Retail
ALWWRT	X	X	X		X		
CANCEL		X			X	X[2]	X
CNLINVITE		X	X	X	X	X	X
CONFIRM	X				X		
DETACH	X	X	X	X[1]	X		X
ENDGRP		X	X		X	X	X
EOS	X	X	X	X	X	X	X
EVOKE	X	X	X	X	X		X
FAIL	X	X	X	X	X	X	
FMH		X		X[1]	X	X[3]	X
FMTNAME	X				X		
FRCDTA	X				X	X	X
INVITE	X	X	X	X	X	X	X
NEGRSP		X			X	X	X
RQSWRT	X	X	X		X		
RSPCONFIRM	X				X		
SECURITY	X	X	X	X	X		
SUBDEV			X		X		
SYNLVL	X				X		
TIMER	X	X	X	X	X	X	X
VARLEN	X	X	X	X	X	X	X

1 Use of these keywords are restricted. Refer to the *Asynchronous Communications Programmer's Guide* for more details.

2 This keyword is not valid for the 3694 controller. Refer to the *Finance Communications Programmer's Guide* for more details.

3 This keyword is valid for the 3694 controller only. Refer to the *Finance Communications Programmer's Guide* for more details.

Table 6-2 lists the DDS keywords defined in this chapter that are supported by the various communications types for input operations.

Table 6-2. Input DDS Processing Keyword Support

DDS Keyword	APPC	SNUF	BSCEL	Asyn-chronous	Intra-system	Finance	Retail
RCVCANCEL		X			X	X[1]	X
RCVCONFIRM	X				X		
RCVDETACH	X	X	X		X		X
RCVENDGRP		X	X		X	X	X
RCVFAIL	X			X	X		
RCVFMH		X			X	X[2]	X
RCVNEGRSP		X			X	X	X
RCVTRNRND	X	X	X		X		
RECID	X	X	X	X	X	X	X

1 This keyword is not valid for the 3694 controller. Refer to the *Finance Communications Programmer's Guide* for more details.

2 This keyword is valid for the 3694 controller only. Refer to the *Finance Communications Programmer's Guide* for more details.

The Figure 6-15 on page 6-28 shows the priority sequence used by ICF in processing these DDS keywords and data during output operations.

The following chart shows the priority sequence used in processing output DDS keywords and data when you specify more than one keyword on a single record format:

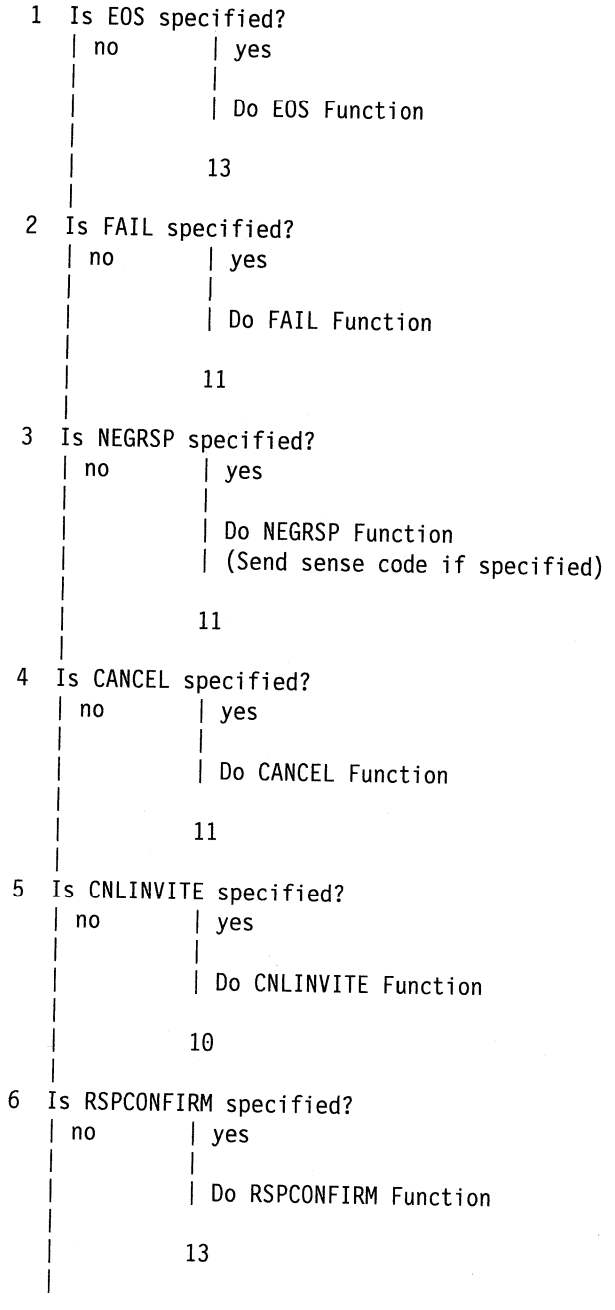


Figure 6-15 (Part 1 of 3). Keyword Processing Chart

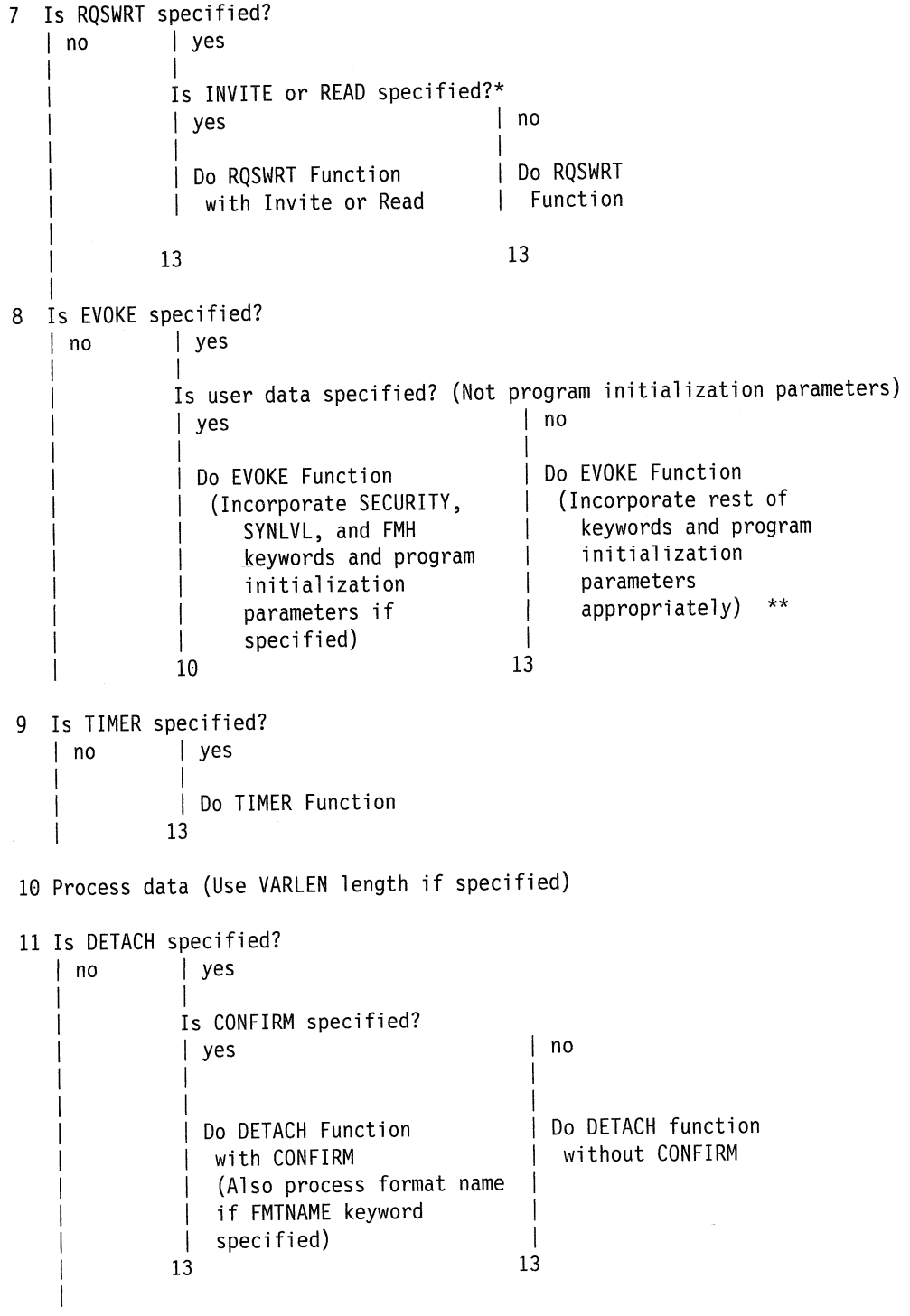


Figure 6-15 (Part 2 of 3). Keyword Processing Chart

```
12 Is ALWWRT, ENDGRP, FMH, SUBDEV, FMTNAME, FRCDTA, CONFIRM, INVITE,
   | no       | yes                          or READ specified?
   |          |
   |          | Do ALWWRT Function if specified
   |          | Do ENDGRP Function if specified
   |          | Do FMH Function if specified
   |          | Do SUBDEV Function if specified
   |          | Do FMTNAME Function if specified
   |          | Do FRCDTA Function if specified
   |          | Do CONFIRM Function if specified
   |          | Do INVITE or READ Function if specified*
   |          |
   |
   |         13
   |
13 END
*  READ and INVITE are mutually exclusive.  If both are specified,
   an error occurs.

** The rest of the keywords: DETACH, CONFIRM, INVITE, ALWWRT, ENDGRP,
   FMH, SUBDEV, FMTNAME, and FRCDTA, are processed in the same
   priority sequence as step 11 and step 12 above.  SUBDEV is not
   allowed with EVOKE if there is no user data.
```

Figure 6-15 (Part 3 of 3). Keyword Processing Chart

Chapter 7. Using System-Supplied Communications Formats

This chapter defines the system-supplied communications formats you can use in your program to control data communications with the remote system. These system-supplied formats are used in place of user-defined data description specifications (DDS) record formats on the write operation. This chapter also maps these system-supplied communications formats to their DDS keyword counterparts in Table 7-5 on page 7-29.

Programming examples are included to show you how these system-supplied formats are used. These examples are program segments only. You can find complete C/400, COBOL/400, and RPG/400 program examples in Chapter 9 through Chapter 11.

All system-supplied formats described in this chapter may not be supported by the communications type you are using. Table 7-4 on page 7-28 summarizes the support provided by each communications type. For more detail, refer to the appropriate communications programming manual for the communications type you are using.

General Description

You can use system-supplied formats for communications only when using an ICF file. You can either create your own file or use the default file provided by ICF for communications when using system-supplied formats. This file, QICDMF, is in library QSYS. You must still perform the override commands for QICDMF to define your program device names.

The QICDMF file was created with the following characteristics:

- The INDARA keyword is used in this file; therefore, a separate indicator area must be specified in your program when this file is used.
- *NONE was specified for the ACQPGMDEV parameter. Therefore, no program device is acquired when the file is opened.
- The maximum record length for the file is 4096 bytes. The maximum record length is used in allocating I/O buffers. If your program does not need this large a record, you may want to override this value by using the Override Intersystem Communications Function File (OVRICFF) command, specifying the MAXRCDLEN parameter.
- The maximum number of program devices that can be acquired with this file is five. If your program uses more than five program devices, you will need to change this file by using the Change Intersystem Communications Function File (CHGICFF) command and specifying a larger value for the MAXPGMDEV parameter.
- 30 SECONDS was specified for the WAITFILE parameter.
- *NOMAX was specified for the WAITRCD parameter.
- *NO was specified for the SHARE parameter.
- *USE was specified for the AUT parameter. Refer to the *Security Concepts and Planning* for information on what rights this characteristic provides.

Do not change this file with the CHGICFF command unless you need to change the maximum number of program devices or want to provide different default characteristics system-wide than those provided at file creation. Use the Override Intersystem Communications Function File (OVRICFF) command to temporarily override any characteristics needed by a particular application.

Do not add any program device entries to the file using the Add Intersystem Communications Function Device Entry (ADDICFDEVE) command. Define program device entries using the Override Intersystem Communications Function Device Entry (OVRICFDEVE) command.

The primary communications functions you can perform using system-supplied formats are:

- Evoke functions (starting remote programs)
- Output functions (sending data)
- Detach functions (ending communications transactions)
- End-of-session functions (ending the session)

These functions are described on the following pages.

All of the system-supplied formats are specified on output operations. The system-supplied formats that allow you to perform the invite and timer functions do, however, affect input processing.

Note: This chapter discusses only how you can use system-supplied formats to do specific communications functions, such as starting and stopping a communications transaction and sending data. Your program will, of course, need to perform additional operations such as starting a session and receiving data. Refer to the appropriate sections in Chapter 5 for information on these operations.

Starting a Program on the Remote System

The target program must be started before communications can begin between your program and a target program. To start a target program and to start a communications transaction, your program must issue an evoke function.

Evoke

You can use one of the following three system-supplied formats to perform an evoke function:

- **Evoke ($$EVOKNI).** Starts the specified program on the remote system. Your program remains in send state, so it can send data to the target program.
- **Evoke with Invite ($$EVOK).** Starts the specified program on the remote system and invites that program to send data.
- **Evoke with Detach ($$EVOKET).** Starts the specified program on the remote system and ends the communications transaction, without allowing the target program to communicate in return. Refer to "Ending a Communications Transaction" on page 7-23 for more information on the detach function.

Figure 7-1 on page 7-3 shows how to start a target program on the remote system.

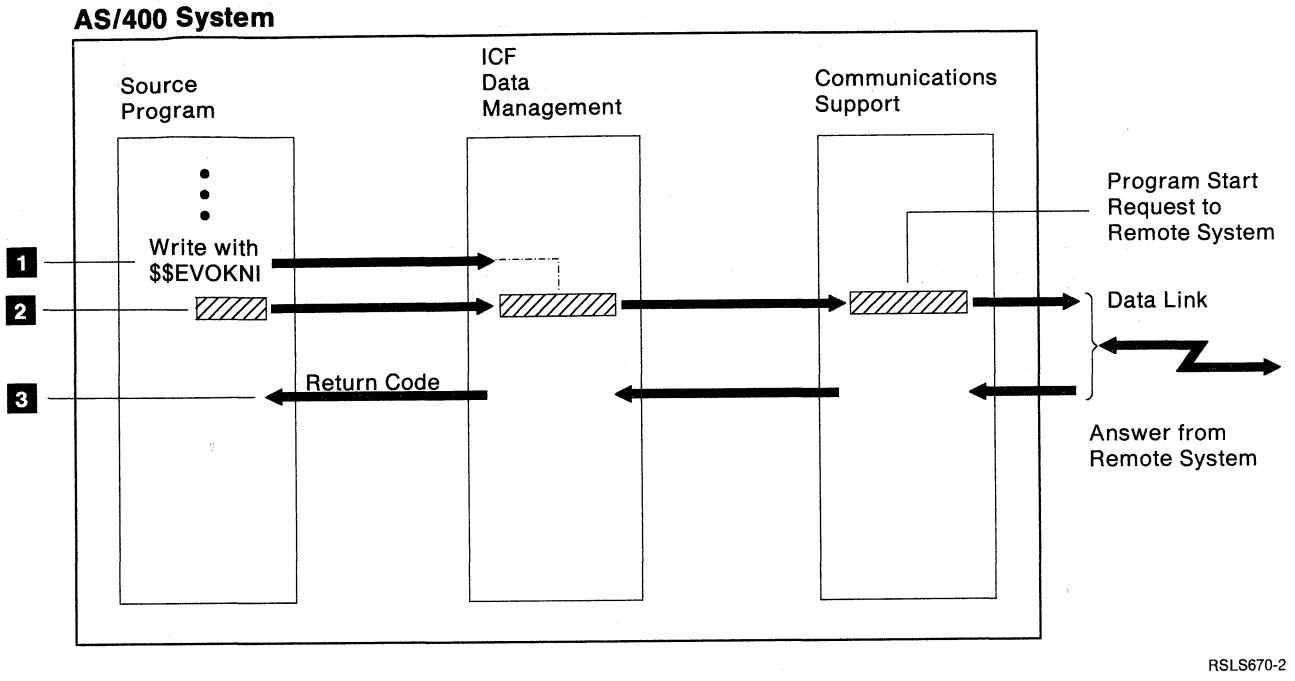


Figure 7-1. Starting a Target Program

1. The source program uses an evoke function to start the program at the remote system.
2. The evoke parameters, including program name, library name, and security information, are sent to the remote system.
3. A successful completion return code tells the source program that the evoke function was accepted and a program start request was sent to the remote system. If the program start request is successful, both the program at the remote system and the communications transaction are started.

You must specify an **evoke parameter list** in the output buffer with an evoke function. The evoke parameter list contains information for the remote system, such as what program to start on the remote system. Specify the field parameters in that list using the format shown in Table 7-1 on page 7-4.

Table 7-1. Evoke Parameter List

Positions	Field Description
1 through 8	The name of the program to be evoked (left-adjusted)
9 through 16	The password you use to sign on the remote system (left-adjusted)
17 through 24	The user identifier you use to sign on the remote system (left-adjusted)
25 through 32	The name of the remote system library that contains the program to be evoked (left-adjusted)
33 through 52	Reserved
53 through 56	The length of data (program parameters)
57 through xxxx	Program initialization parameters

If a field is not used, enter the correct number of blanks for the unused field.

If multiple program initialization parameters are used, the program is responsible for using the proper separation characters for the remote system. For example, if the remote system is an AS/400 system, multiple parameters must be separated by a comma.

If the remote system is another AS/400 system, the program parameters are passed to the target program as if they were passed from a Call a Program (CALL) command. Data sent with an evoke function are parameters used by the target program.

System-supplied formats do not allow a synchronization level of CONFIRM and always revert to the default synchronization level of NONE.

Figure 7-2 is an example of a C/400 write statement that can be used to issue an evoke.

```
struct {
    char program_name??(8??);
    char password??(8??);
    char user_id??(8??);
    char library_name??(8??);
    char filler??(20??);
    char data_length??(4??);
    char data??(1000??);
} evoke_rec;

:
FILE *icffptr;                      /* Pointer to the ICF file */

:
icffptr = fopen("ICFFILE","ab+ type=record indicators=y");

:
QXXFORMAT(icffptr, "$$EVOKNI  ");   /* Set evoke w/no invite format */
QXXPGMDEV(icffptr, "CM1       ");   /* Set default device to CM1 */

fwrite(&evoke_rec, sizeof(evoke_rec), 1, icffptr);   /* Do the evoke */
```

Figure 7-2. Evoke C/400 Write Statement

Figure 7-3 on page 7-5 is an example of a COBOL/400 WRITE statement which can be used to issue an evoke.

```
01  DATA-RECORD.
    03  PROGRAM-NAME      PIC X(8).
    03  PASSWORD          PIC X(8).
    03  USER-ID           PIC X(8).
    03  LIBRARY-NAME      PIC X(8).
    03  FILLER            PIC X(20).
    03  DATA-LENGTH       PIC 9(4).
    03  THE-DATA          PIC X(256).
     •
     •
     •
    WRITE TRANSACTION-RECORD FROM DATA-RECORD,
       FORMAT IS '$$EVOKNI', TERMINAL IS ICF-PGMDEV.
```

Figure 7-3. Evoke COBOL/400 WRITE Statement

Figure 7-4 is an example of an RPG/400 output specification used to issue an evoke function.

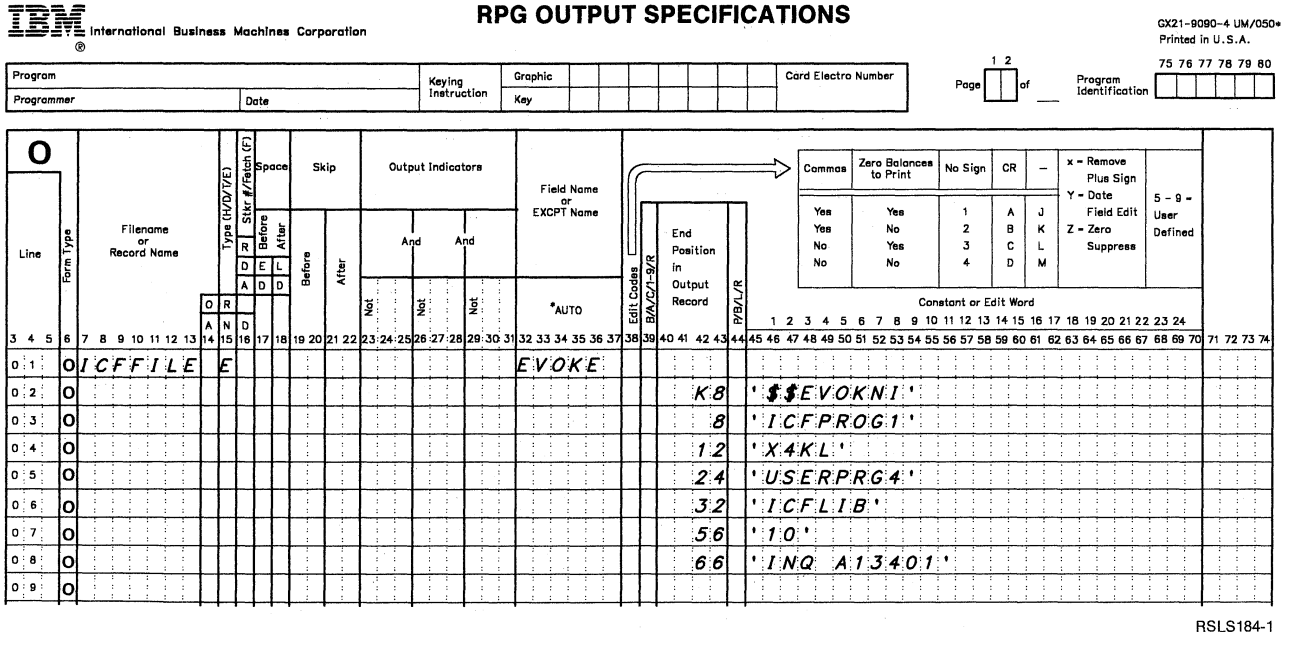


Figure 7-4. Evoke RPG/400 Output Specification

Sending Data

A data record can be sent from your program to the remote program. The following list describes each of the system-supplied send formats that can be used to send data:

- **Send ($$SENDNI)**. Sends one data record to the remote program.
- **Send with Invite ($$SEND)**. Sends one data record to the remote program and issues an invite to the remote program. Your program must use an input operation to receive the data sent from the remote system.

- **Send with Function-Management-Header ($$SENDNF)**. Sends a data record that includes a function-management-header to the remote program. Function-management-header data contains control information that tells the remote system about the data being sent.
- **Send with Function-Management-Header and Invite ($$SENDFM)**. Sends a data record that includes a function-management-header and an invite to the remote program.
- **Send with End-of-Group ($$SENDE)**. Sends a data record to the remote program and tells the remote program that the record is the last in a group or chain of records.
- **Send with Detach ($$SENDET)**. Sends a data record to the remote program and tells the remote program that your program is ending this communications transaction. Communications between the two programs have ended. Refer to "Ending a Communications Transaction" on page 7-23 for more information on the detach function.

Note: Except for $$SENDFM and $$SENDNF, you can specify a length of zero and perform any of the preceding functions without sending any data.

Figure 7-5 shows how to use system-supplied formats to send data.

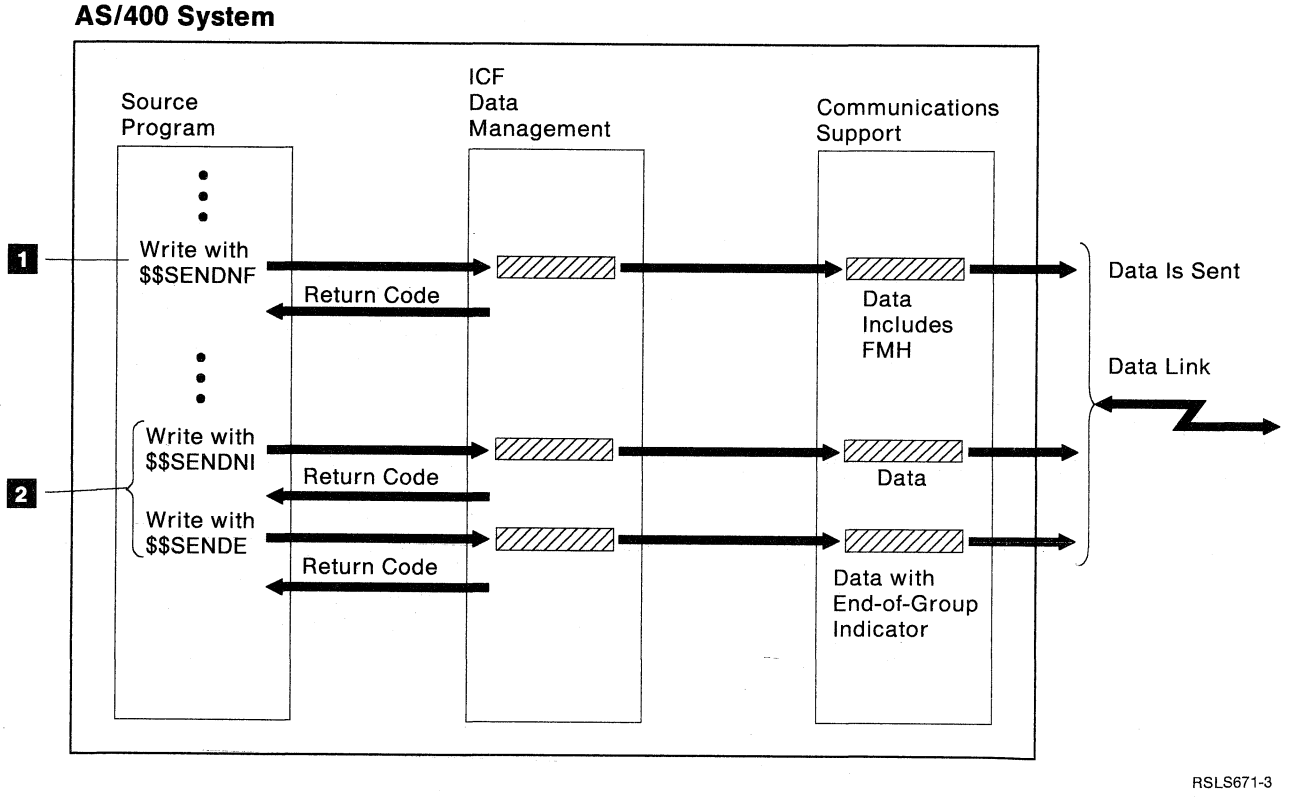


Figure 7-5. Using $$SENDNF, $$SENDNI, and $$SENDE to Send Data

1 Your program sends a record to the remote system and, with the $$SENDNF communications format, indicates that the first part of the data is function-management-header data. The function-management-header data contains information about the user data that follows.

2 Your program continues sending data records to the remote system. Your program uses the $$SENDE communications format on the last record to indicate it is the last in this group of records.

Each of the preceding functions requires the fields shown in Table 7-2 in the output buffer.

Table 7-2. Required Output Fields

Positions	Description
1 through 4	Length of user data (in decimal)
5 through xxxx	The user data to be sent

Figure 7-6 is an example of a C/400 write statement that sends one data record.

```
struct {
    char record_length??(4??);
    char data??(80??);
} data_rec;

:
FILE *icffptr;                      /* Pointer to the ICF file */

:
icffptr = fopen("ICFFILE","ab+ type=record indicators=y");

:
QXXFORMAT(icffptr, "$$SENDNI  ");   /* Set write w/no invite format */
QXXPGMDEV(icffptr, "CM1       ");   /* Set default device to CM1 */

strncpy(data_rec.data_length, "0080", 4);   /*Set record length*/
fwrite(&data_rec, sizeof(data_rec), 1, icffptr);  /* Do the write */
```

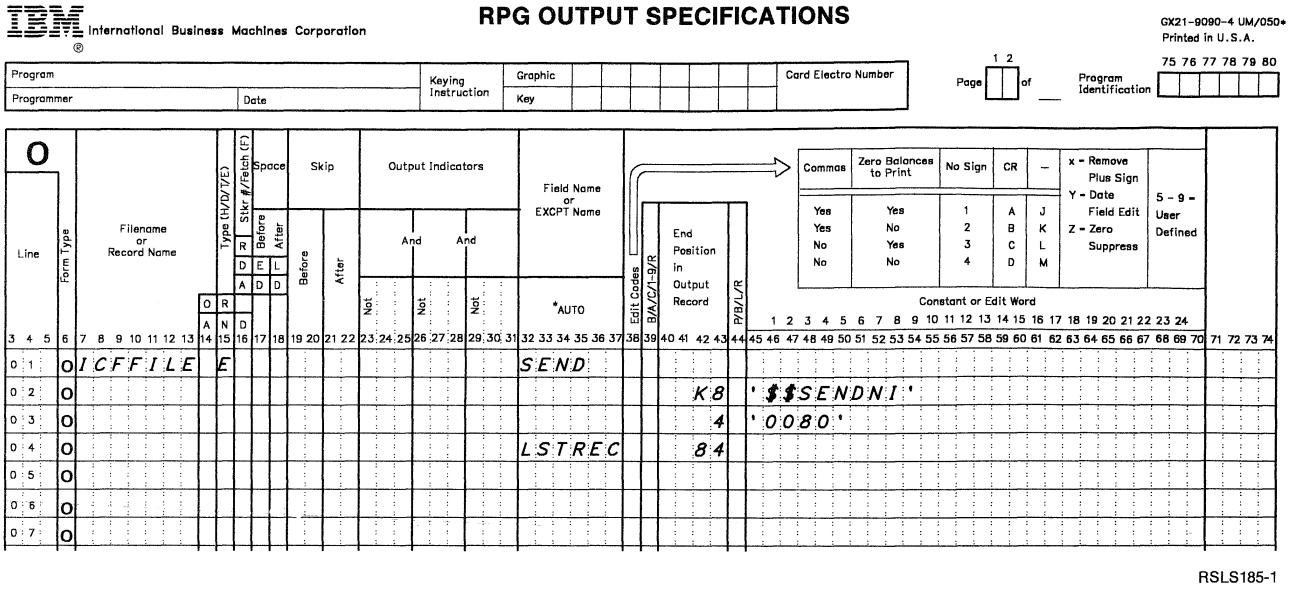
Figure 7-6. Send C/400 Write Statement

Figure 7-7 is an example of a COBOL/400 WRITE statement that sends one data record.

```
01  DATA-RECORD.
    03  RECORD-LENGTH    PIC 9(4).
    03  THE-RECORD       PIC X(256).
     •
     •
     •
    WRITE TRANSACTION-RECORD FROM DATA-RECORD,
       FORMAT IS '$$SENDNI', TERMINAL IS ICF-PGMDEV.
```

Figure 7-7. Send COBOL/400 WRITE Statement

Figure 7-8 on page 7-8 is an example of an RPG/400 output specification to send one data record.



RSLS185-1

Figure 7-8. Send RPG/400 Output Specification

Receiving Data

You can use two operations to receive data: read and read-from-invited-program-devices. In addition, you can use the invite and timer functions to provide additional functions when receiving data.

The read operation receives data from the program device you specify. This operation differs from the read-from-invited-program-devices operation, which receives data from any program device with a previously issued invite.

Invite

The invite function prepares your program to receive data. You can use one of the following system-supplied formats (in combination with other functions) to perform an invite function:

- **Evoke with Invite ($$EVOK)**. Starts the specified program on the remote system and invites that program to send data.
- **Send with Invite ($$SEND)**. Sends one data record to the remote program, and issues an invite to the remote program, asking it to send. Your program must issue an input operation to receive the data sent from the remote program.
- **Send with Function-Management-Header and Invite (SENDFM)**. Sends a data record that includes a function-management-header to the remote program, followed by an invite.
- **Cancel with Invite ($$CANL)**. Cancels the current chain of data, then invites the remote program to send its own data. Refer to "Problem Notification" on page 7-11 for more information about the cancel function.
- **Negative Response with Invite ($$NRSP)**. Sends a negative-response indication to the remote program, followed by an invite. Refer to "Problem Notification" on page 7-11 for more information about the negative-response function.

- **Request-to-Write with Invite ($$RCD)**. Sends a request-to-write indication to the remote program, followed by an invite. Refer to "Additional System-Supplied Formats" on page 7-18 for more information about the request-to-write function.

Refer to "Starting a Program on the Remote System" on page 7-2, "Problem Notification" on page 7-11, and "Additional System-Supplied Formats" on page 7-18 for information about the output format of these functions.

Note: You can use the $$SEND communications format with an output length of zero to issue an invite function by itself.

The read-from-invited-program-devices operation is a companion to the invite function. After issuing an invite function, use the read-from-invited-program-devices operation to receive data from the remote system.

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

Refer to Chapter 5 for additional information about the read and read-from-invited-program-devices operations and their relationship to the invite function.

Timer

Your program can use the timer function to set a timer before performing a specified function, such as a read-from-invited-program-devices operation. The timer function specifies an interval of time (in hours, minutes, and seconds) to wait before your program receives a timer-expired (0310) return code.

Use the $$TIMER system-supplied format to issue the timer function. The output field for the timer request must be in the following format:

```
hhmmss
```

where hh is hours, mm is minutes, and ss is seconds.

Figure 7-9 on page 7-10 is a C/400 example that shows how to use $$TIMER and set the timer to 30 seconds.

```
FILE *icffptr;                          /* Pointer to the ICF file */

:
icffptr = fopen("ICFFILE","ab+ type=record indicators=y");

:
QXXFORMAT(icffptr, "$$TIMER   ");   /* Set timer format */
QXXPGMDEV(icffptr, "CM1       ");   /* Set default device to CM1 */

fwrite("000030", 6, 1, icffptr);    /* Issue timer function */

      /* See if the timer ended by checking 0310 return code */

if (strncmp(_Maj_Min_rc.major_rc, "03", 2) == 0 &&
    strncmp(_Maj_Min_rc.minor_rc, "10", 2) == 0)

      timer_exp();        /* Timer ended, call timer_exp */
                          /* routine to handle the time out */
```

Figure 7-9. Timer C/400 Statement

7-10 is a COBOL/400 example that shows how to use $$TIMER and set the timer to 30 seconds. A read-from-invited-program-devices operation is used to receive the data. The return code must be checked for the timer-expired return code.

```
01  TIMER                          PIC X(6) VALUE '000030'.
     •
     •
     •
    WRITE TRANSACTION-RECORD FROM TIMER,
       FORMAT IS '$$TIMER', TERMINAL IS ICF-PGMDEV.
     •
     •
     •
    READ TRANSACTION-FILE,
       IF RETURN-CODE EQUAL '0310',
          THEN
             GO TO TIMER-EXPIRED.
```

Figure 7-10. Timer COBOL/400 Statement

Your program continues to run, and all operations and functions are valid during the time interval. Your program must issue a read-from-invited-program-devices operation some time after it has issued the timer function, so it can accept the timer-expired return code.

Only one time interval can be maintained for your program. If a previous timer function has been issued and the timer has not yet ended, the old time interval is replaced by the new interval.

The timer function can be used to vary the maximum amount of time that a read-from-invited-program-devices operation will wait for a response. When the time interval set by the TIMER keyword is in effect, the value specified for the WAITRCD parameter on the CRTICFF command is ignored.

You can use the timer function to retry other operations that may not be successful, possibly because of a temporary lack of resources (for example, during an acquire operation). To do this, issue the timer function, and then perform read-from-invited-program-devices operations until the timer ends. (The read-from-invited-program-

devices operation allows the program to continue receiving input from other invited program devices while waiting for the timer).

Refer to Chapter 5 for additional information on the read-from-invited-program-devices operation and its relationship to the timer function.

Figure 7-11 is an example of using RPG/400 to enter the value on the output specifications to set the timer for 30 seconds.

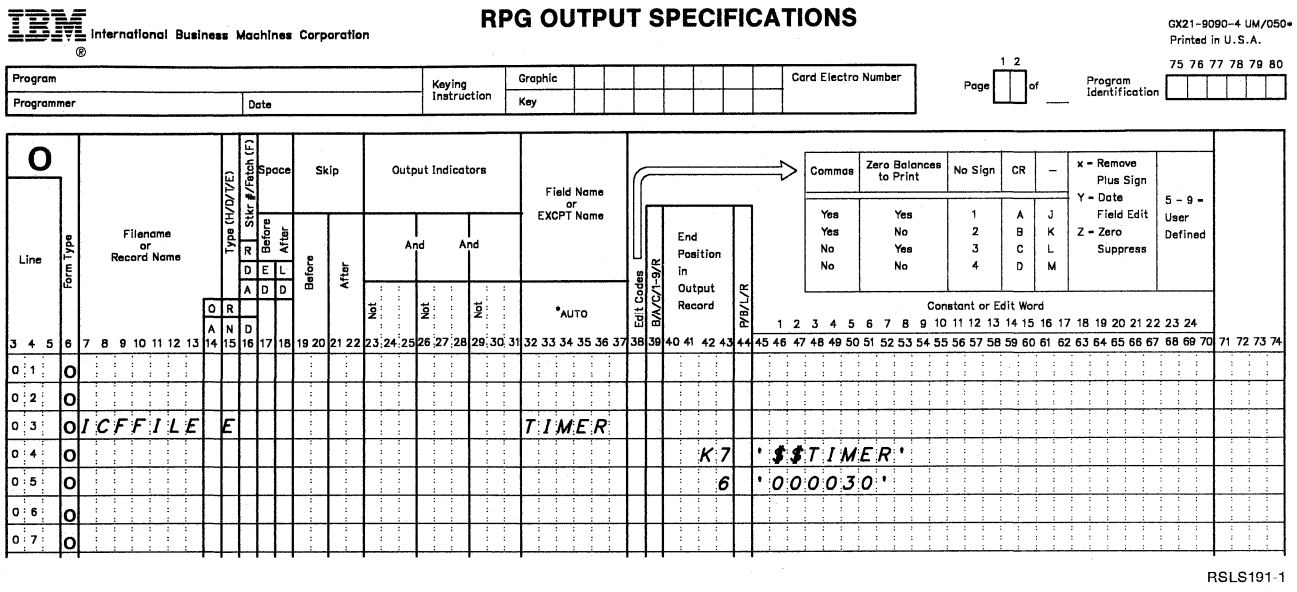


Figure 7-11. Timer RPG/400 Output Specification

Problem Notification

The following are the system-supplied formats that may be used to indicate that some type of error has occurred.

Fail

Use the fail to indicate that your program detected an abnormal condition while sending or receiving data. Use the $$FAIL system-supplied format to issue a fail. No data can be sent with the fail.

Figure 7-12 on page 7-12 shows how to use the $$FAIL communications format when your program is receiving data and detects an error.

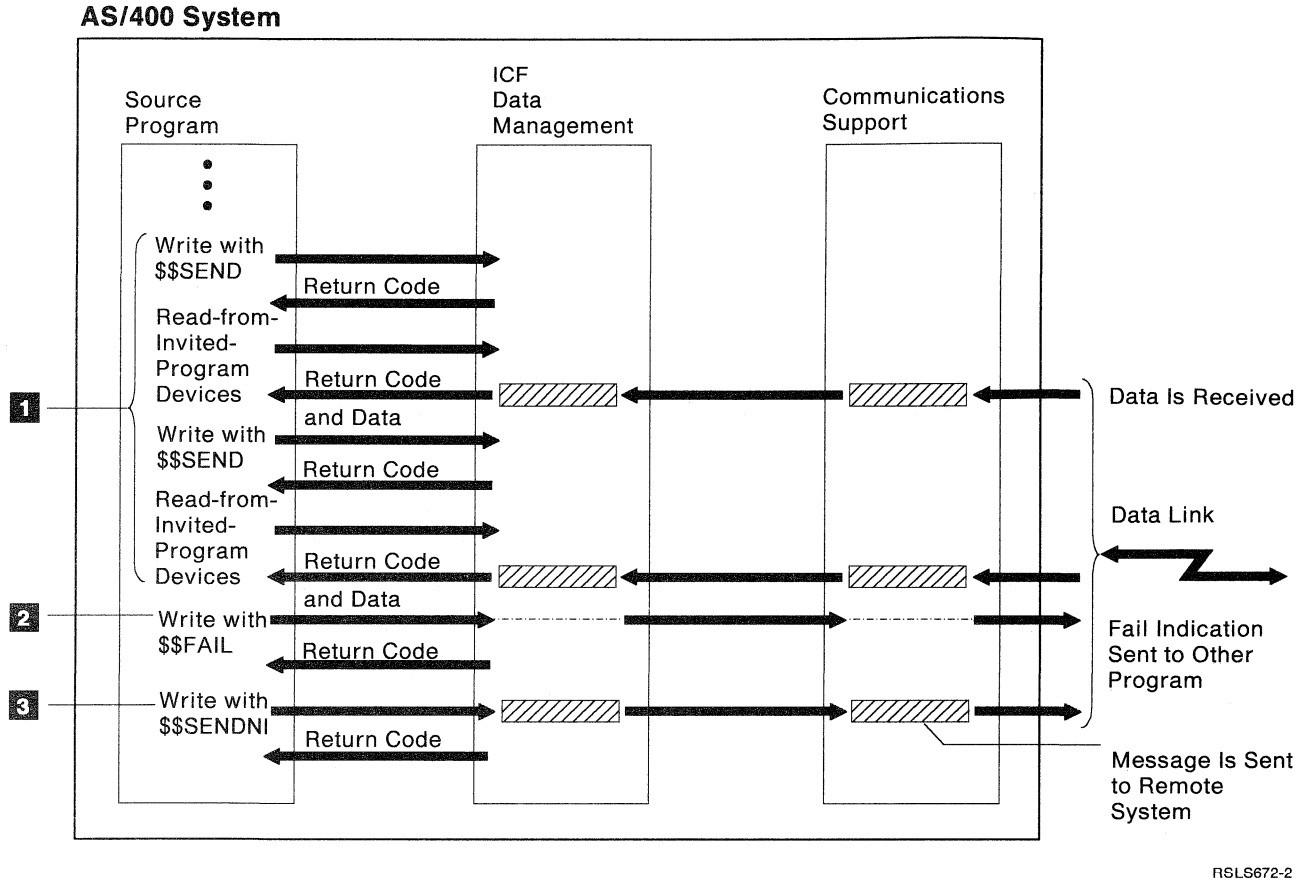


Figure 7-12. Using $$FAIL to Send an Error Signal

1. Your program is receiving data from the remote program.
2. While receiving data, your program determines that it must send a fail indication to the other program. A write with the $$FAIL communications format is used to send the fail indication.
3. A message or data is then sent (write operation) to tell the other program why you sent the fail.

No output fields are associated with the $$FAIL communications format.

Figure 7-13 is a C/400 write statement example that sends a fail indication.

```
FILE *icffptr;                          /* Pointer to the ICF file */

:
icffptr = fopen("ICFFILE","ab+ type=record indicators=y");

:
QXXFORMAT(icffptr, "$$FAIL     ");   /* Set fail format */
QXXPGMDEV(icffptr, "CM1        ");   /* Set default device to CM1 */

fwrite(NULL, 0, 0, icffptr);           /* Send the fail */
```

Figure 7-13. Fail C/400 Write Statement

Figure 7-14 on page 7-13 is a COBOL/400 WRITE statement example that sends a fail indication.

```
WRITE TRANSACTION-RECORD,
   FORMAT IS '$$FAIL', TERMINAL IS ICF-PGMDEV.
```

Figure 7-14. Fail COBOL/400 WRITE Statement

Figure 7-15 is an example of an RPG/400 output specification to send a fail indication.

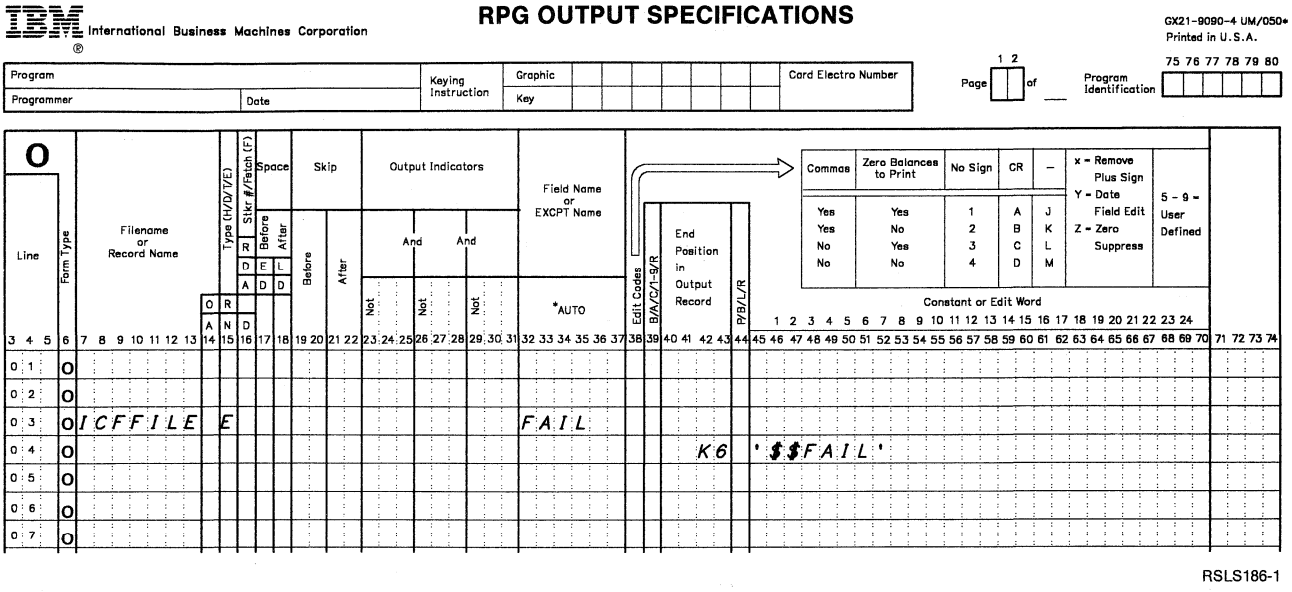


Figure 7-15. Fail RPG/400 Output Specification

Cancel

Use the cancel function to cancel the current chain of data (group of records) that is being sent to the remote program. The receiving program disregards all the records sent in the current chain. You can use two system-supplied formats to perform the cancel function:

- **Cancel ($$CANLNI)**. Cancels the current chain of data.
- **Cancel with Invite ($$CANL)**. Cancels the current chain of data, and then invites the remote program to send its own data.

Figure 7-16 on page 7-14 shows how to use the $$CANLNI communications format when your program is sending data and detects an error.

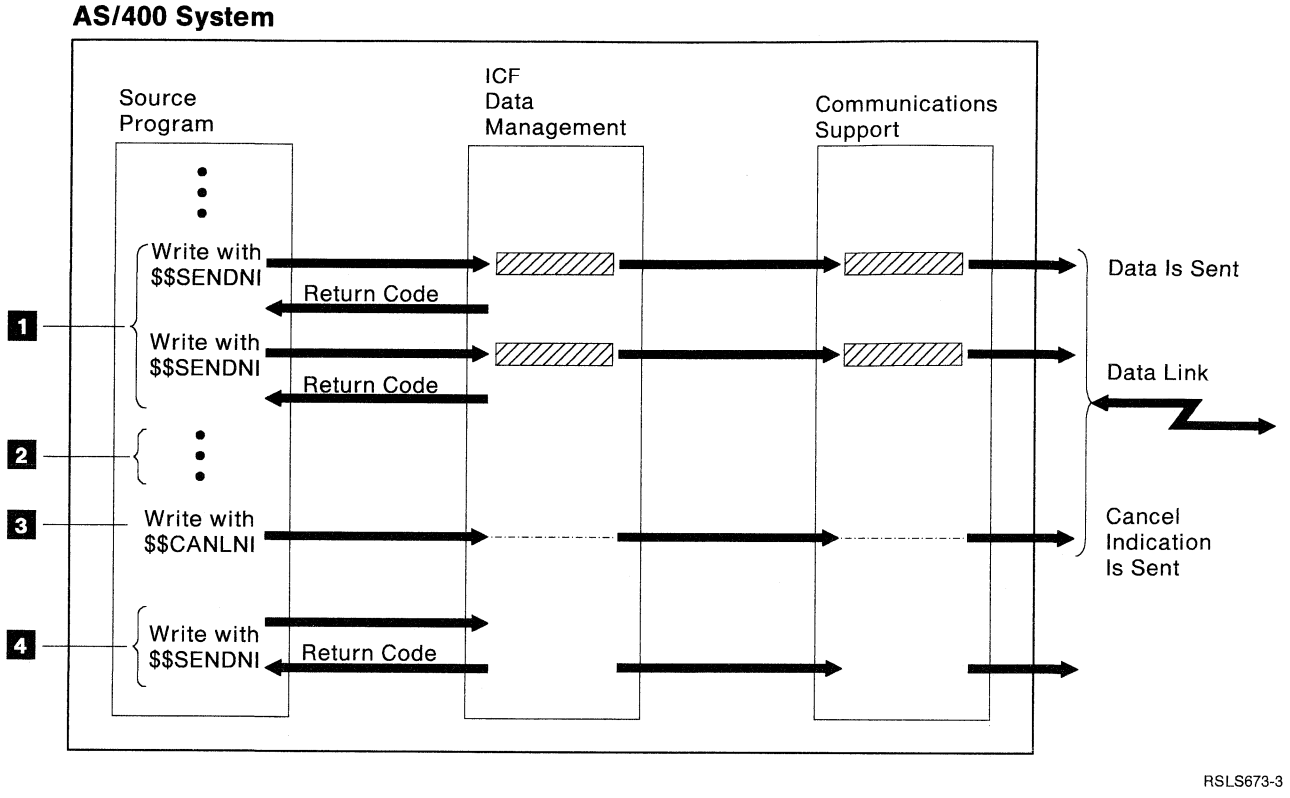


Figure 7-16. Using $$CANL to Send an Error Indication

1. Your program is sending data to the remote system.
2. Your program checks the data and determines that something is wrong with it.
3. A write operation with the $$CANLNI communications format is used to tell the remote system to discard the data you have sent.
4. Your program can send a message indicating the problem, send the data again, receive more data, or end the transaction.

No output fields are associated with the $$CANLNI and $$CANL communications formats.

Figure 7-17 is a C/400 write statement example that cancels the current chain of records.

```
FILE *icffptr;                          /* Pointer to the ICF file */

:
icffptr = fopen("ICFFILE","ab+ type=record indicators=y");

:
QXXFORMAT(icffptr, "$$CANLNI  ");   /* Set cancel w/no invite format */
QXXPGMDEV(icffptr, "CM1       ");   /* Set default device to CM1 */

fwrite(NULL, 0, 0, icffptr);            /* Send the cancel */
```

Figure 7-17. Cancel C/400 Write Statement

Figure 7-18 on page 7-15 is a COBOL/400 WRITE statement example that cancels the current chain of records.

```
WRITE TRANSACTION-RECORD
  FORMAT IS '$$CANLNI', TERMINAL IS ICF-PGMDEV.
```

Figure 7-18. Cancel COBOL/400 WRITE Statement

Figure 7-19 is an example of an RPG/400 output specification to cancel the current chain of records.

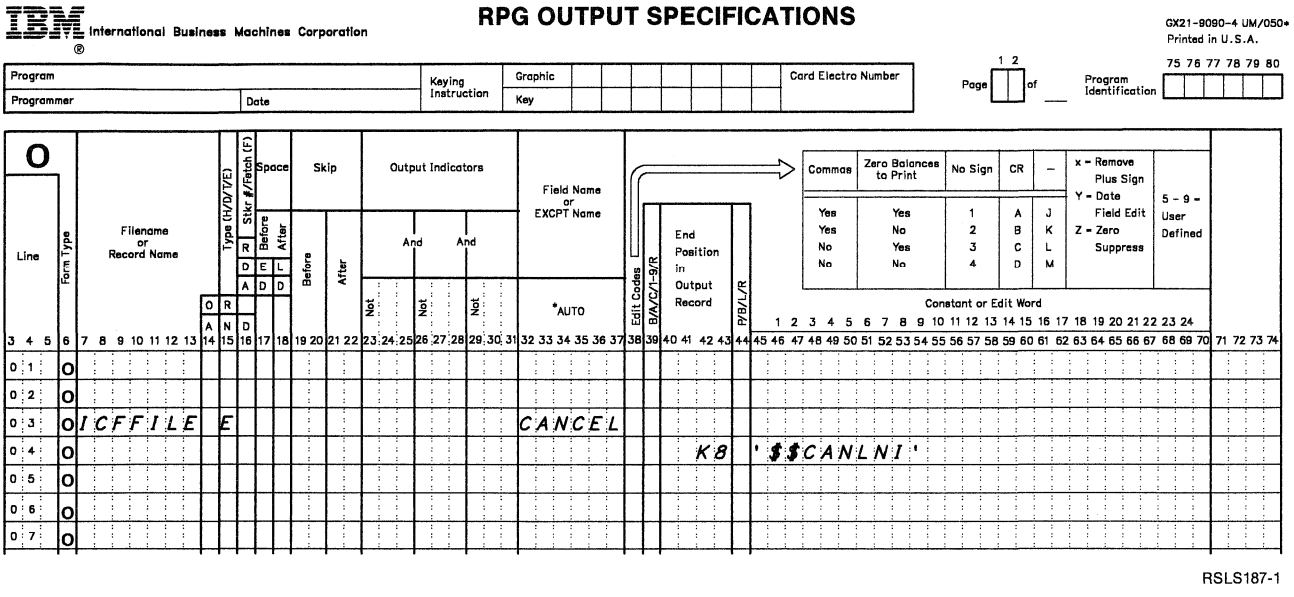


Figure 7-19. Cancel RPG/400 Output Specification

Negative-Response

Your program uses the negative-response function to indicate it detected something wrong with the data it received. You can use two system-supplied formats to issue the negative response function:

- **Negative-Response ($$NRSPNI)**. Sends a negative-response indication to the remote program.
- **Negative-Response with Invite ($$NRSP)**. Sends a negative-response indication to the remote program, followed by an invite.

Your program must use an input operation to determine the action taken by the remote program after issuing a negative response. You can send 8 bytes of user data (sense data) indicating the reason for the negative response with the negative-response indication.

Figure 7-20 on page 7-16 shows how to send a negative response with a sense code to the remote system.

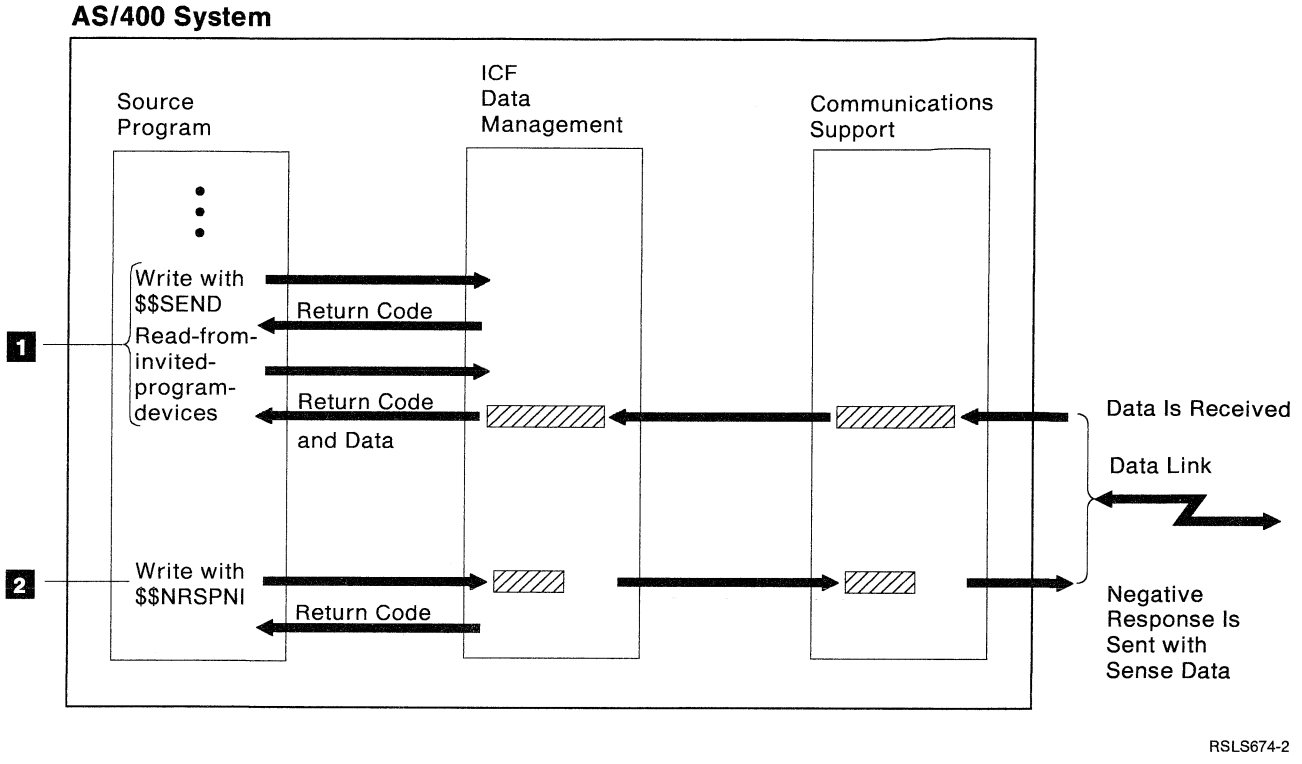


Figure 7-20. Using $$NRSP to Send an Error Condition

1. Your program finds the data it is receiving is not correct.
2. The program sends a negative response, by issuing a write with the $$NRSPNI communications format, to the remote system including the sense data 08110000. The negative response tells the remote system that the data received is wrong, and the sense data 08110000 asks the remote system to cancel the current group of data records.

The negative-response function requires the fields shown in Table 7-3 in the output buffer:

Table 7-3. Sense Data Format

Positions	Description
1	Indicates whether sense data is being sent: 0 or blank indicates that *no* sense data is being sent; 8 indicates that sense data is being sent.
2 through 9	The sense data sent with the negative response. The first four positions of the sense data must begin with 10xx, 08xx, or 0000. The last four positions are user-defined.

For more information about the allowed sense values, refer to the appropriate communications programming manual for the communications type you are using.

Figure 7-21 on page 7-17 shows a C/400 write statement that sends a negative response with invite function that includes the sense data 08110000.

```
struct {
    char length;
    char data??(8??);
} neg_resp_rec = {"8", "08110000"};

:
FILE *icffptr;                          /* Pointer to the ICF file */

:
icffptr = fopen("ICFFILE","ab+ type=record indicators=y");

:
QXXFORMAT(icffptr, "$$NRSP    ");   /* Set negative response format */
QXXPGMDEV(icffptr, "CM1       ");   /* Set default device to CM1 */

     /* Send negative response with sense data */
fwrite(&neg_resp_rec, sizeof(neg_resp_rec), 1, icffptr);
```

Figure 7-21. Negative-Response C/400 Write Statement

Figure 7-22 shows a COBOL/400 WRITE statement that sends a negative response with invite function that includes the sense data 08008000.

```
01  NEG-RESP-REC.
    03  REC-LEN                    PIC X(1) VALUE '8'.
    03  RESP-DATA                  PIC X(08) VALUE '08008000'.
     •
     •
     •
    WRITE TRANSACTION-RECORD FROM NEG-RESP-REC,
       FORMAT IS '$$NRSP', TERMINAL IS ICF-PGMDEV.
```

Figure 7-22. Negative-Response COBOL/400 WRITE Statement

Figure 7-23 shows the RPG/400 output specifications you can use to send a negative response that includes the sense data 08110000.

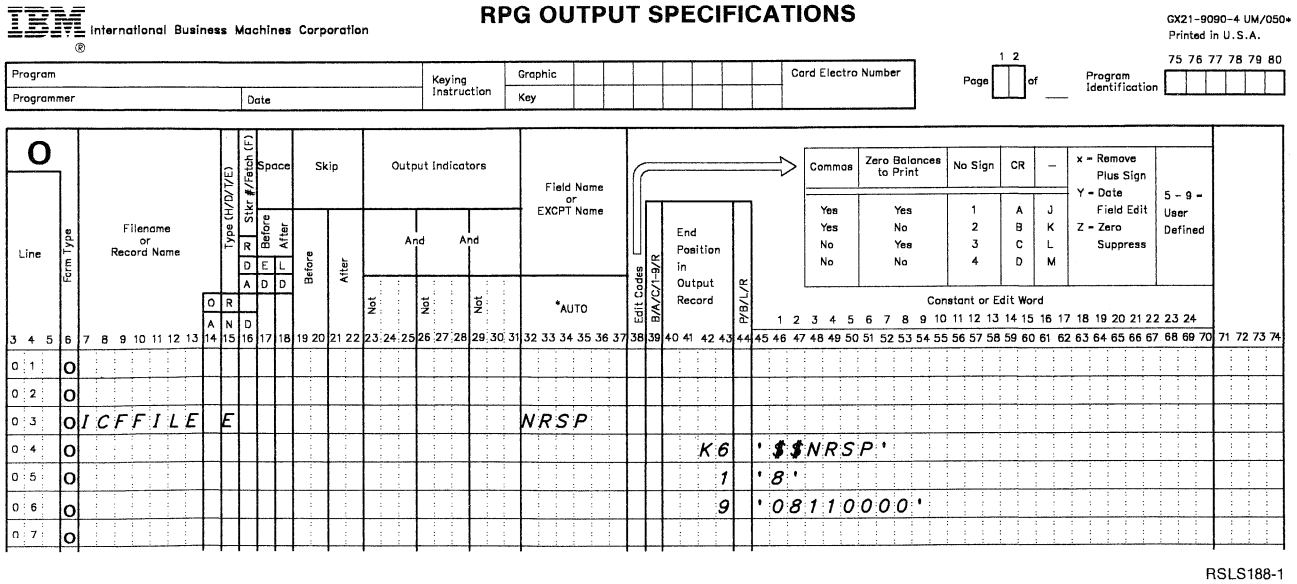


Figure 7-23. Negative-Response RPG/400 Output Specifications

Additional System-Supplied Formats

You can use the following additional system-supplied formats to perform specific tasks when communicating with the remote system.

Request-to-Write

Your program uses the request-to-write function to indicate that it wants to send something to the remote program rather than to continue receiving data.

Use the $$RCD system-supplied format to issue the request-to-write function. This format issues the request-to-write with an invite. After you issue the request-to-write, your program must continue to receive data until it receives a return code indicating that the remote system is ready to begin receiving data. The request-to-write function has no additional associated parameters or data.

Figure 7-24 on page 7-19 shows how to use the $$RCD communications format to request permission to send.

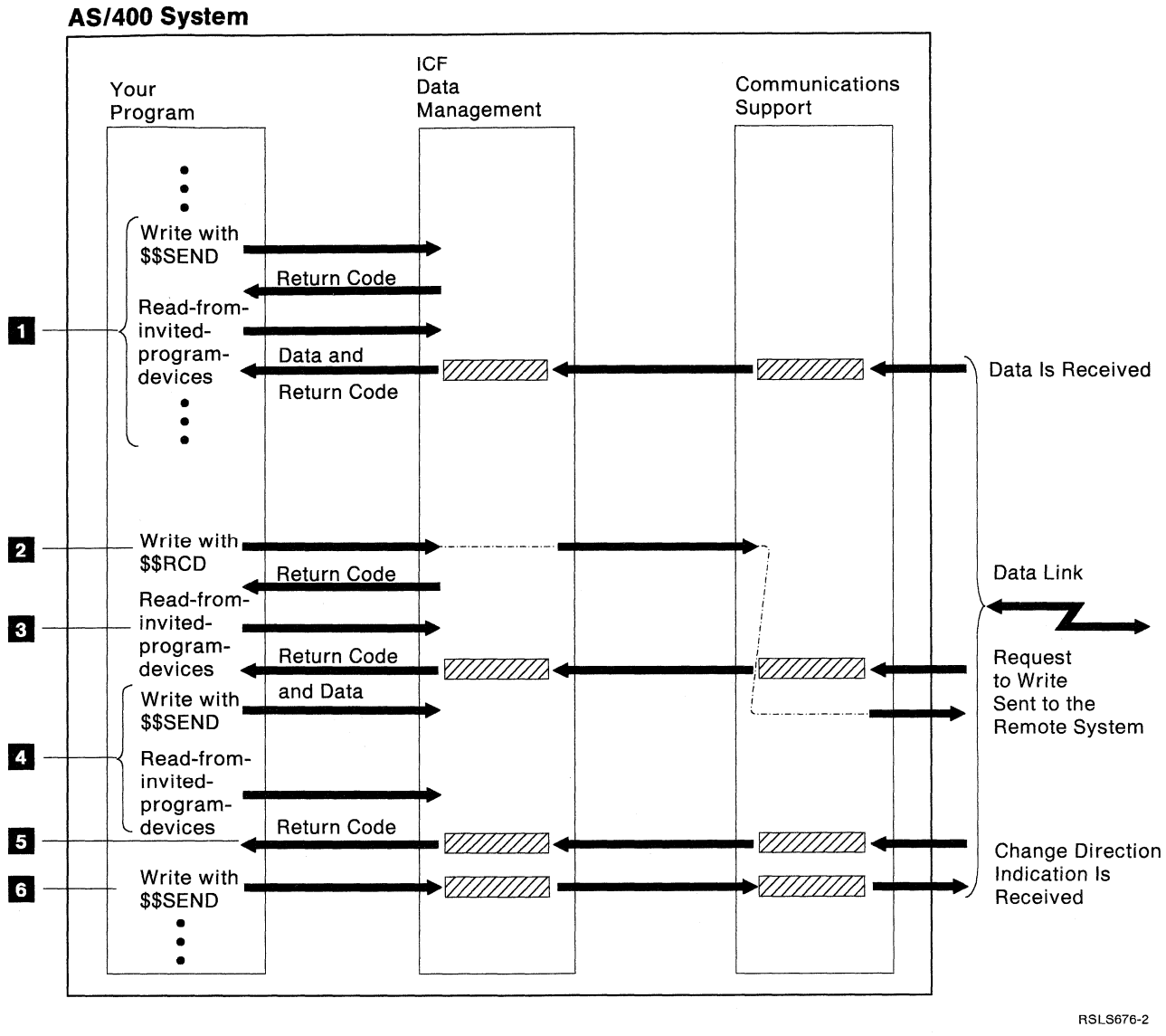


Figure 7-24. Using $$RCD to Request Write

1 Your program is receiving data from the remote system. The first program processes the data received, then receives data again.

2 At some time while data is being received, your program determines that it needs to send a message to the remote system. A write with the $$RCD communications format is used to ask the remote system to stop sending so your program can send the message.

3 After issuing the request-to-write, your program must continue receiving data until it gets a return code indicating that the remote system is ready to receive. To continue receiving, your program issues another read-from-invited-program-devices operation.

4 Another invite function and another read-from-invited-program-devices operation are issued to continue receiving data.

5 When the remote system is ready to receive, it sends one more data record with a change-direction indication. The record says the remote system is now ready to receive data or, as in this example, a message.

6 A write with the $$SEND communications format is used to send the message to the remote system and ask the remote system to continue sending data.

No output fields are associated with the $$RCD communications format.

When your program receives a request-to-write indication from the remote system, a code is set in the input/output (I/O) feedback communications-dependent section. Refer to Table C-5 on page C-5 for more information about where this field is in the I/O feedback area.

The code indicates the following conditions:

0 Continue sending as normal.

1 A request-to-write has been received.

For more specific information on what this code means for the communications type you are using, refer to the appropriate communications programmer's guide.

Figure 7-25 is an example of a C/400 write statement to request the remote system to stop sending data.

```
FILE *icffptr;                          /* Pointer to the ICF file */

:
icffptr = fopen("ICFFILE","ab+ type=record indicators=y");

:
QXXFORMAT(icffptr, "$$RCD     ");   /* Set request-to-write format */
QXXPGMDEV(icffptr, "CM1       ");   /* Set default device to CM1 */

fwrite(NULL, 0, 0, icffptr);          /* Send the request-to-write */
```

Figure 7-25. Request-to-Write C/400 Write Statement

Figure 7-26 is an example of a COBOL/400 WRITE statement to request the remote system to stop sending data.

```
WRITE TRANSACTION-RECORD,
   FORMAT IS '$$RCD', TERMINAL IS ICF-PGMDEV.
```

Figure 7-26. Request-to-Write COBOL/400 WRITE Statement

Figure 7-27 on page 7-21 is an example of a RPG/400 output specification to request that the remote system stop sending data.

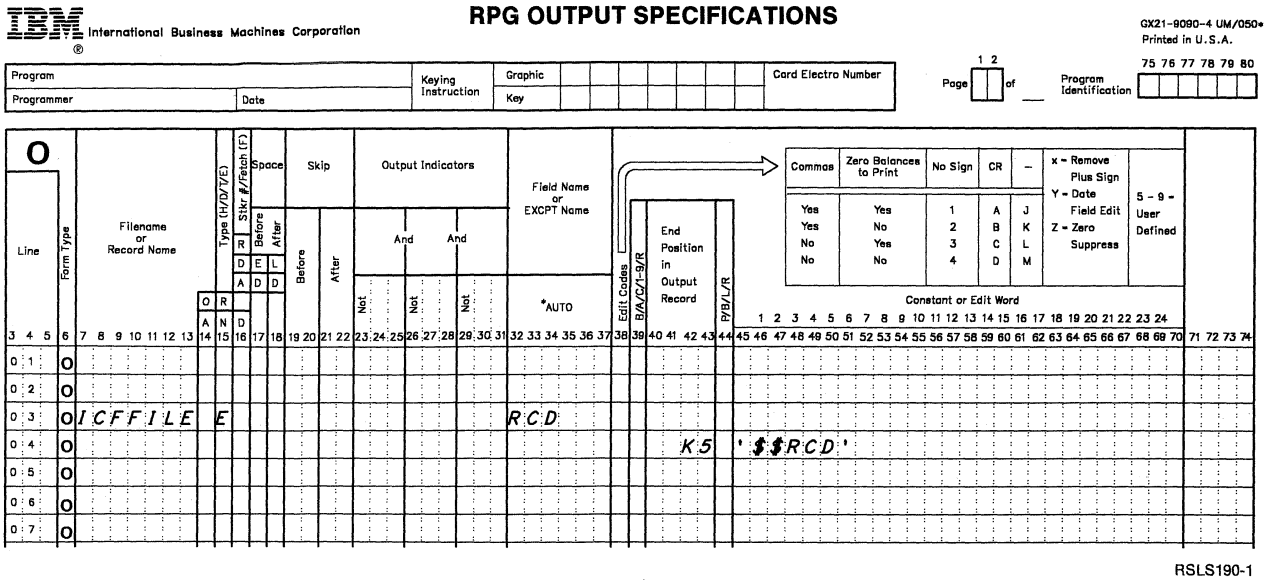


Figure 7-27. Request-to-Write RPG/400 Output Specification

Cancel-Invite

Your program uses the cancel-invite function to cancel any valid invite for which no data has yet been received. Use the $$CNLINV system-supplied format to issue the cancel-invite function.

Figure 7-28 shows how to use the $$CNLINV communications format to issue the cancel-invite function.

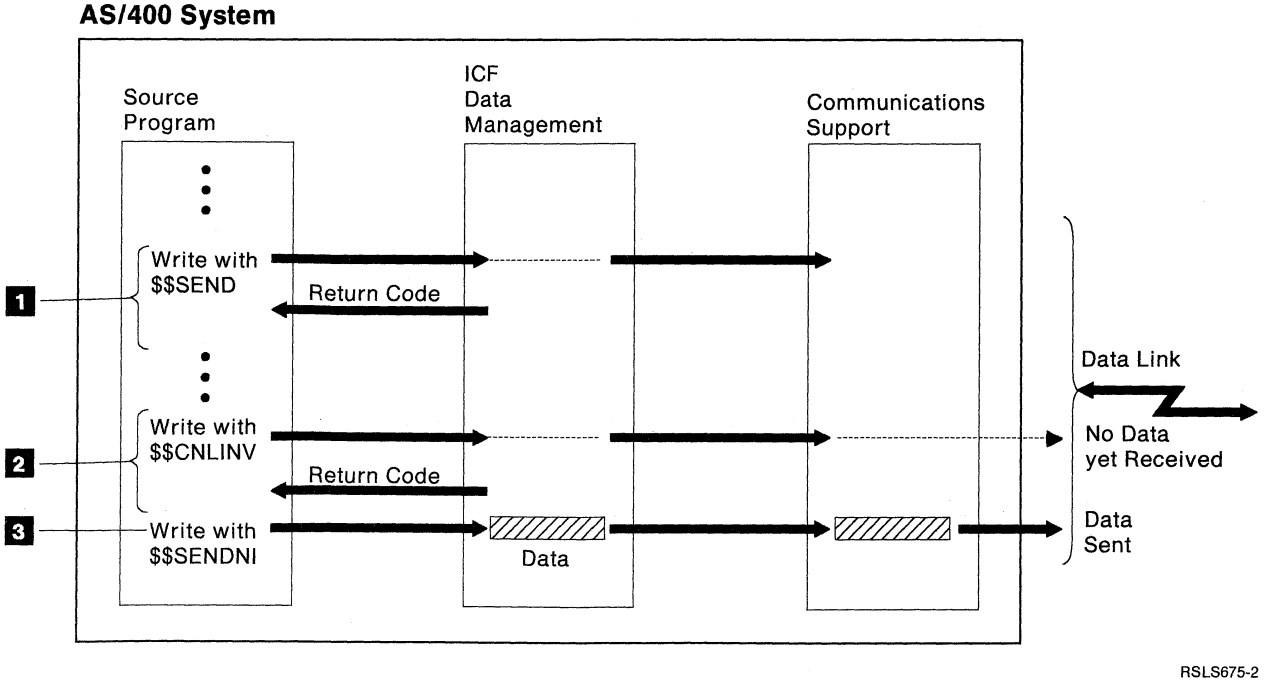


Figure 7-28. Using $$CNLINV to Cancel an Invite

1. Your program issues an invite to receive data from the remote program, then continues processing.
2. Your program can cancel the invite issued previously using the cancel-invite function (issuing a write operation with the $$CNLINV communications format). Your program must check the return code it receives to determine if the invite was cancelled.
3. If a successful return code is received, your program can send data.

No output fields are associated with the $$CNLINV communications format.

Figure 7-29 is an example C/400 write statement that issues a cancel-invite function to a program device that has not received input.

```
FILE *icffptr;                          /* Pointer to the ICF file */

:
icffptr = fopen("ICFFILE","ab+ type=record indicators=y");

:
QXXFORMAT(icffptr, "$$CNLINV  ");   /* Set cancel-invite format */
QXXPGMDEV(icffptr, "CM1       ");   /* Set default device to CM1 */

fwrite(NULL, 0, 0, icffptr);        /* Issue the cancel-invite */
```

Figure 7-29. Cancel-Invite C/400 Write Statement

Figure 7-30 is an example COBOL/400 WRITE statement that issues a cancel-invite function to a program device that has not received input.

```
WRITE TRANSACTION-RECORD
  FORMAT IS '$$CNLINV', TERMINAL IS ICF-PGMDEV.
```

Figure 7-30. Cancel-Invite COBOL/400 WRITE Statement

Figure 7-31 on page 7-23 is an example RPG/400 output specification to issue a cancel-invite to a program device that has not received input.

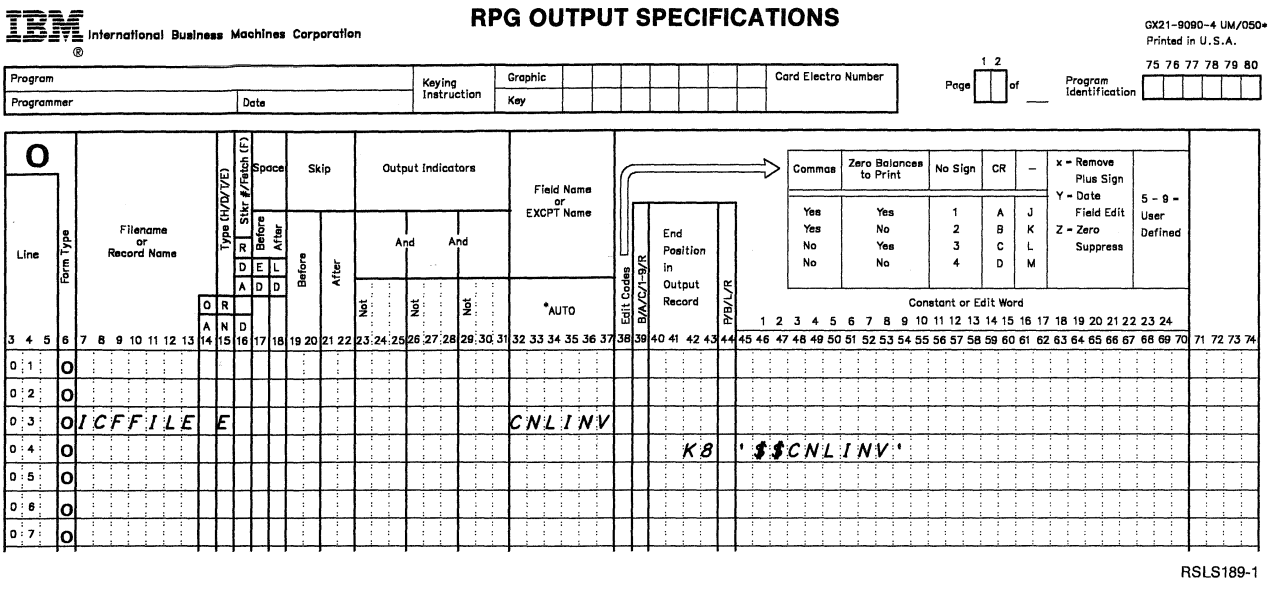


Figure 7-31. Cancel-Invite RPG/400 Output Specifications

Ending a Communications Transaction

A communications transaction can be ended by your program or by the program at the remote system. Your job and the remote system your system is with communicating determine the program that ends the transaction.

Communications with the remote program end when your program ends the transaction. However, the session may still exist if your program started the session. If the session still exists, you can end the session or you may be able to start another program at the remote system.

Detach

Use the detach function to end the transaction. The detach function explicitly informs the remote program that your program is done sending, and ends the transaction.

You can use one of the following system-supplied formats to perform a detach function (in combination with other functions):

- **Evoke with Detach ($$EVOKET)**. Starts the specified program on the remote system and ends the transaction without allowing the target program to communicate in return.
- **Send with Detach ($$SENDET)**. Sends a data record to the remote program and ends the transaction. Note that not all communications types support sending a data record and the detach function on the same operation. Refer to the appropriate communications manual to determine if you can send data with the detach function.

You can use the $$SENDET communications format with an output length of zero to issue a detach function by itself.

Figure 7-32 on page 7-24 shows how to use the $$SENDET communications format to end the transaction.

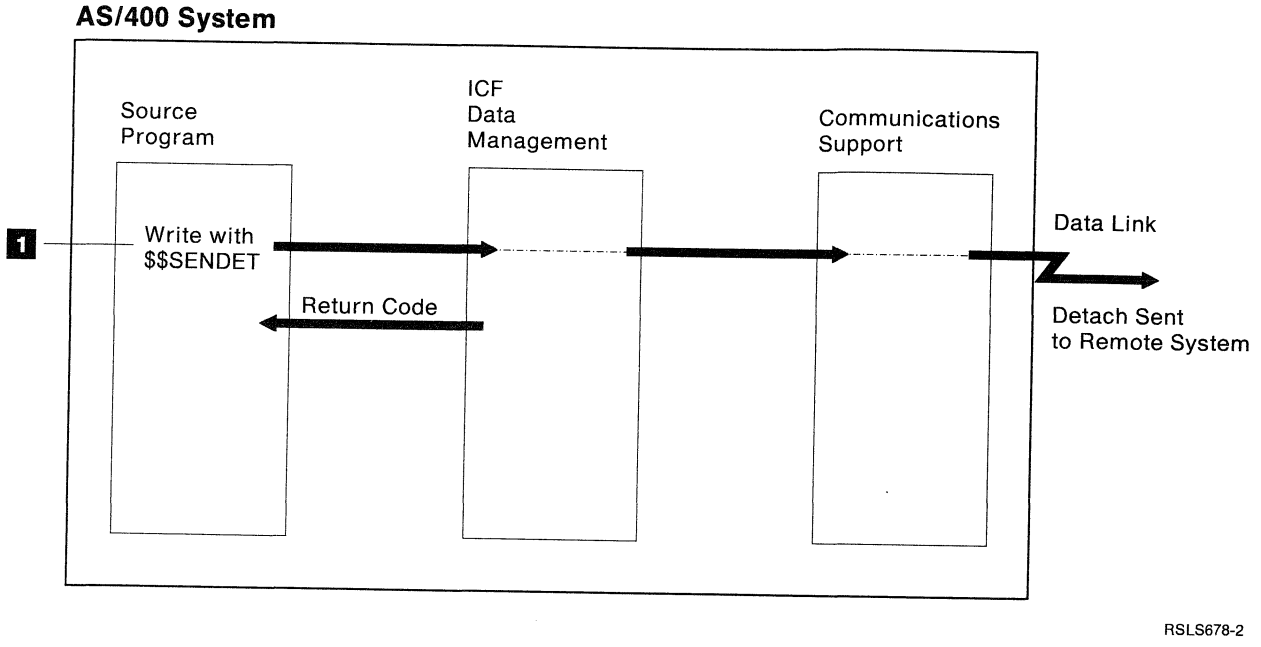


Figure 7-32. Ending the Communications Transaction

1 Your program issues the detach function, by using a write with the $$SENDET system-supplied format, to tell the remote system that your program ended the communications transaction.

Refer to "Starting a Program on the Remote System" on page 7-2 and "Sending Data" on page 7-5 for information about the output format of these functions.

Figure 7-33 is an example of a C/400 write statement to issue a detach.

```
struct {
    char data_length??(4??);
    char data??(80??);
} data_rec;

:
FILE *icffptr;                          /* Pointer to the ICF file */

:
icffptr = fopen("ICFFILE","ab+ type=record indicators=y");

:
QXXFORMAT(icffptr, "$$SENDET  ");   /* Set write-with-detach format */
QXXPGMDEV(icffptr, "CM1       ");   /* Set default device to CM1 */

strncpy(data_rec.data_length, "0080", 4);   /*Set record length*/
fwrite(&data_rec, sizeof(data_rec), 1, icffptr); /* Send detach */
```

Figure 7-33. Detach C/400 Write Statement

Figure 7-34 on page 7-25 is an example of a COBOL/400 WRITE statement to issue a detach.

```
01  DATA_RECORD.
    03  RECORD-LENGTH           PIC 9(4).
    03  THE-RECORD              PIC X(256).
     •
     •
     •
WRITE TRANSACTION-RECORD FROM DATA-RECORD,
   FORMAT IS '$$SENDET', TERMINAL IS ICF-PGMDEV.
```

Figure 7-34. Detach COBOL/400 WRITE Statement

If a detach function is issued by a target program, the session is ended as well as the transaction.

Figure 7-35 is an example of an RPG/400 output specification to issue a detach.

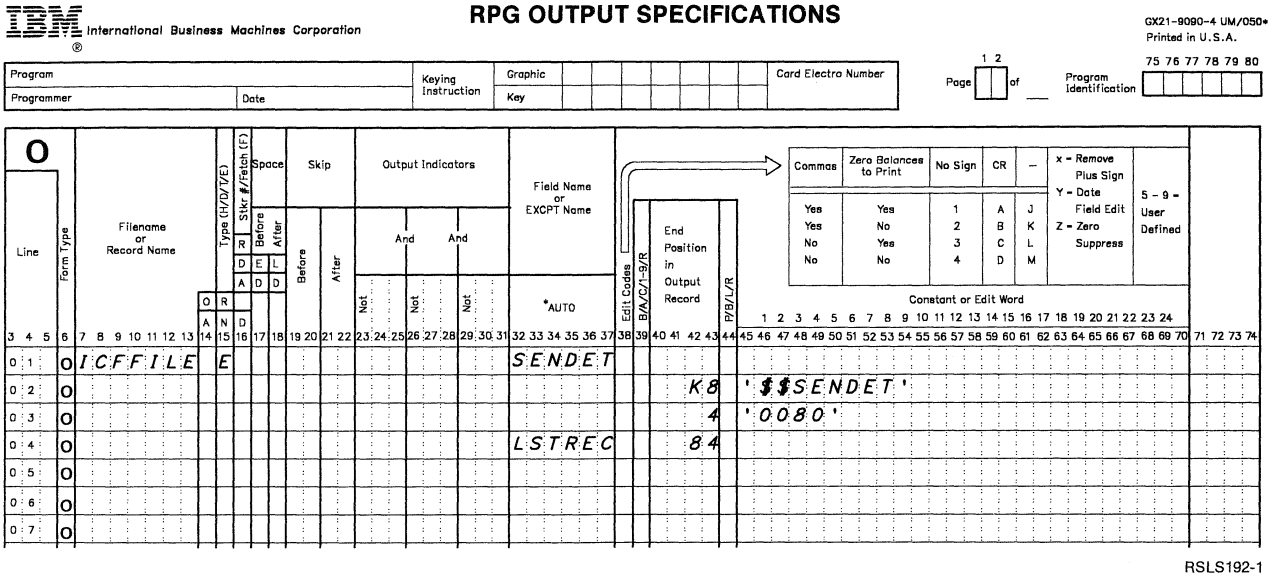


Figure 7-35. Detach RPG/400 Output Specification

Ending the Communications Session

How the communications session is ended depends on whether your program or the remote system started the session.

If your program started the session (source program), your program must end the session using either the release operation or the end-of-session function. You should primarily use the release operation. Use the end-of-session function only when you want to force the session to end.

The release operation ends the session only if all processing is complete. The end-of-session function *always* ends the session. The only possible return codes from end-of-session are 0000 or 830B (program device not acquired).

End of Session

Use the $$EOS system-supplied format to issue the end-of-session function. No data can be sent with the end-of-session function. Figure 7-36 shows how you can end the session by using the release operation and the end-of-session function.

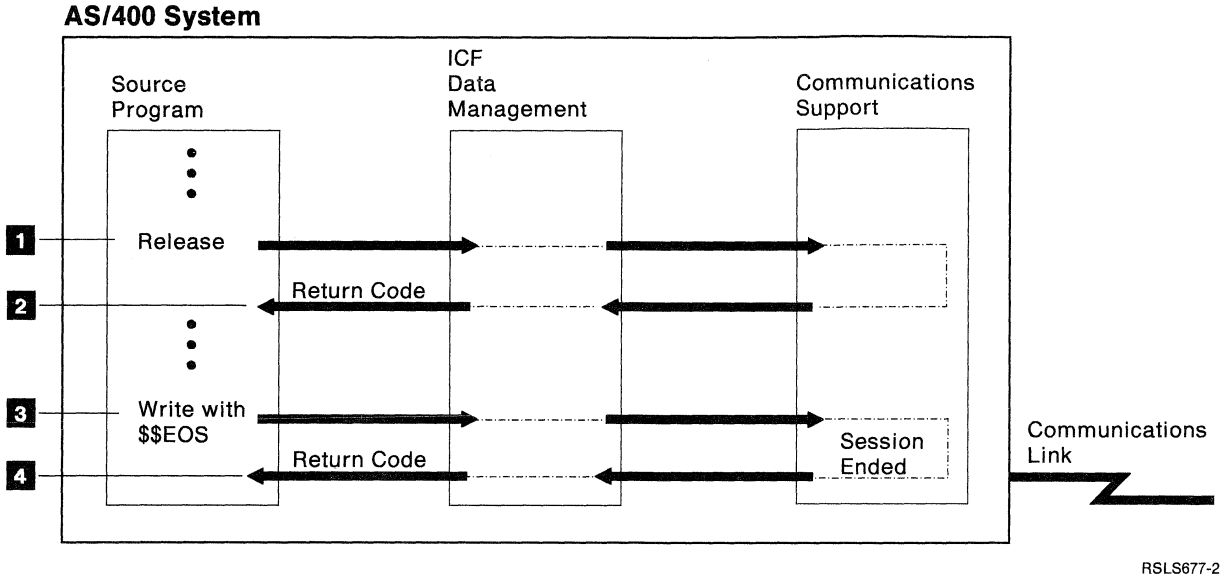


Figure 7-36. Using the Release Operation and End-of-Session Function

1. Your program uses the release operation to end the current communications session.
2. The return code tells your program whether the session was ended or an error occurred while trying to end the session. If, for example, all transactions have not ended when the release operation is issued, an error occurs and the session is not ended.
3. If an error occurs and normal recovery is not possible, your program can use the end-of-session function to end the session.
4. The end-of-session function always ends the session.

If the remote system started the session (the local program is a target program), the target program must issue an end-of-session function or go to end-of-job to end the session.

No output fields are associated with the $$EOS communications format.

Figure 7-37 on page 7-27 is an example of a C/400 write statement specifying the $$EOS format.

```
FILE *icffptr;                          /* Pointer to the ICF file */

:
icffptr = fopen("ICFFILE","ab+ type=record indicators=y");

:
QXXFORMAT(icffptr, "$$EOS     ");   /* Set end-of-session format */
QXXPGMDEV(icffptr, "CM1       ");   /* Set default device to CM1 */

fwrite(NULL, 0, 0, icffptr);          /* Send an end-of-session */
```

Figure 7-37. End-of-Session C/400 Write Statement

Figure 7-38 is an example of a COBOL/400 WRITE statement specifying the $$EOS format.

```
WRITE TRANSACTION-RECORD,
   FORMAT IS '$$EOS', TERMINAL IS ICF-PGMDEV.
```

Figure 7-38. End-of-Session COBOL/400 WRITE Statement

Figure 7-39 is an example of an RPG/400 output specification to issue the end-of-session function using the $$EOS format.

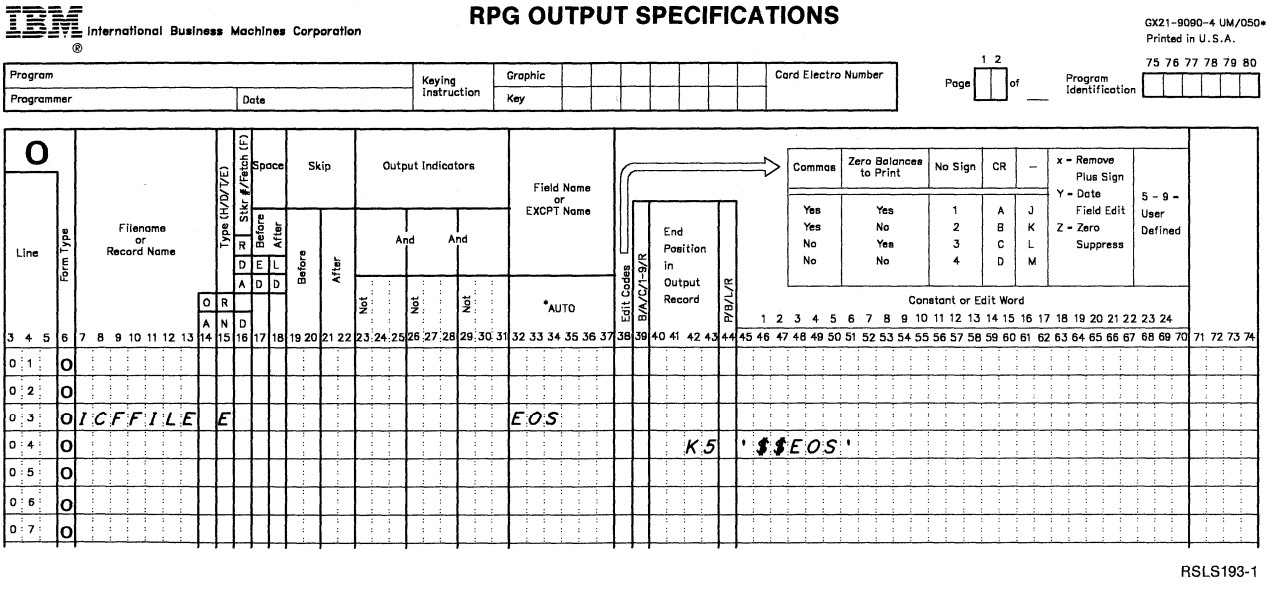


Figure 7-39. End-of-Session RPG/400 Output Specification

System-Supplied Format Support

Table 7-4 shows which system-supplied formats are supported by each communications type.

Table 7-4. System-Supplied Format Support

System-Supplied Formats	APPC	SNUF	BSCEL	Asyn-chronous	Intra-system	Finance	Retail
$$CANL		X			X	X[1]	X
$$CANLNI		X			X	X[1]	X
$$CNLINV		X	X	X	X	X	X
$$EOS	X	X	X	X	X	X	X
$$EVOK	X	X	X	X	X		X
$$EVOKET	X	X	X	X	X		X
$$EVOKNI	X	X	X	X	X		X
$$FAIL	X	X	X	X	X	X	
$$NRSP		X			X	X	X
$$NRSPNI		X			X	X	X
$$RCD	X	X	X		X		
$$SEND	X	X	X	X	X	X	X
$$SENDE		X	X		X	X	X
$$SENDET	X	X	X		X		X
$$SENDFM		X			X	X[2]	X
$$SENDNF		X		X	X	X[2]	X
$$SENDNI	X	X	X	X	X	X	X
$$TIMER	X	X	X	X	X	X	X

1 These keywords are not valid for the 3694 controller. Refer to the *Finance Communications Programmer's Guide* for more details.

2 These keywords are valid for the 3694 controller only. Refer to the *Finance Communications Programmer's Guide* for more details.

Mapping System-Supplied Formats to DDS Keywords

Table 7-5 maps system-supplied formats to DDS keywords.

Table 7-5. Mapping of System-Supplied Formats to DDS Keywords

System-Supplied Formats	DDS Keywords
$$CANL	CANCEL with INVITE
$$CANLNI	CANCEL
$$CNLINV	CNLINVITE
$$EOS	EOS
$$EVOK	EVOKE, SECURITY, and INVITE
$$EVOKET	EVOKE, SECURITY, and DETACH
$$EVOKNI	EVOKE and SECURITY
$$FAIL	FAIL
$$NRSP	NEGRSP with INVITE
$$NRSPNI	NEGRSP
$$RCD	RQSWRT with INVITE
$$SEND	INVITE
$$SENDE	ENDGRP
$$SENDET	DETACH
$$SENDFM	FMH with INVITE
$$SENDNF	FMH
$$SENDNI	No DDS keywords
$$TIMER	TIMER

Chapter 8. Programming Considerations

This chapter presents general communications programming considerations related to the ICF file. Programming considerations specific to a communications type are not discussed. Refer to the appropriate communications programming manual for more information on the programming considerations for a particular communications type.

Return Codes

Return codes are used by the communications application program to determine the program state. Program states are receive, send, or exception. The program checks the return codes and completes the action required by the contents of the return codes.

The meaning of the major ICF return codes and some examples of minor return codes follow. These definitions help you determine the return codes you need to check in your program. For a complete list of return codes, see Appendix B.

Major Codes

All major return codes that represent normal conditions have values 00xx, 02xx, and 03xx. Major return codes that represent input/output (I/O) exceptions have values of 04xx and 34xx. Major return codes that represent *error* conditions have values 08xx, 11xx, and 80xx through 83xx. This division lets you quickly compare codes and determine the type of action required.

The main groups of major return codes are:

- Operation was successfully completed (00xx, 02xx).
- Successful operation. No data was received, but some control information may have been received (03xx).
- Output exception occurred (04xx).
- Miscellaneous program errors occurred (08xx and 11xx).
- Input exception occurred (34xx).
- Nonrecoverable error occurred. The session has been ended and the underlying communications support may no longer be active (80xx).
- Nonrecoverable session error occurred. Session has been ended but the underlying communications support is still active (81xx).
- Open or acquire operation failed. Session was not started but recovery might be possible (82xx).
- Session error occurred. Recovery might be possible (83xx).

Minor Codes

The minor part of a return code identifies the specific condition within the general condition that is identified by the major part of the code. Some examples of the minor codes are:

- A detach was received (xx08), a detach was received with a system message (xx18), or a detach was received with a confirm (xx1C).
- An invalid evoke function was issued (xx29).

General Considerations

You must understand the following ICF environment characteristics before you write programs for ICF communications:

- Your program should check the major/minor return code after every operation to determine the success or failure of the operation.
- Information in the open and I/O feedback areas can be useful to your program. Refer to Chapter 5 for information about these areas and Appendix C for a summary of the fields available in these areas.
- A target program can communicate with the source program by acquiring the program device whose remote location name is *REQUESTER.
- If a target program never acquires a program device for the requesting program device, a diagnostic message is written to the job log when the target job completes. The message indicates that the program ended with an active connection to the session. The message is normal if there is no need for the target program to communicate with the source program. If the target program must communicate with the source program, this message indicates a possible logic problem in your program.
- For most communications types, the first I/O operation following the acquire by the source program must include an evoke function. This operation starts the target program on the remote system with which the source program is to communicate. You can start the target program with program initialization parameters specified as part of the evoke function.
- When a single program is written to function as either a source or a target program, the program may need to determine its role (source or target). A suggested method of making this analysis for a program is to define a CALL parameter that contains one value when the program is started by the remote system and another value when the program is started by the local user or program.
- A target program cannot start error recovery. If a permanent session error occurs, the target program should finish any needed processing and end the program. The source program is responsible for reestablishing the session and transaction. See Appendix B for more information about communications error recovery.

Open or Acquire Considerations

The following open or acquire considerations apply when you write programs for ICF:

- If an ICF file open is a subsequent open of a shared file, the program is attached to the already open file. The state and attributes of the file do not change.

Refer to the *Data Management Guide* for more information on shared file processing.

- Your program can establish more than one session. These sessions can be to the same remote system (if the remote system supports multiple sessions) or to different remote systems. The program device names are used to distinguish between sessions within your program.
- A program device can either be acquired automatically when the ICF file is opened (ACQPGMDEV = program device) or explicitly with the acquire operation. Only one program device can be acquired as part of the open operation. If a program is using multiple sessions, all of the program devices, except the first one, must be explicitly acquired.
- If a target program acquires a program device other than a program device associated with a remote location with the name of *REQUESTER, a new session is allocated and a logical link to the source program is not established. No error is indicated because a target program on one session can also be a source program on another session.

Output Considerations

The following output considerations apply when you write programs for ICF:

- Your program should check the major/minor return code after every output operation to determine if the remote system wants to send data. You can also use a field in the I/O feedback to determine this information. See Appendix C for a summary chart of the I/O feedback area.
- When the program is communicating with multiple sessions, the appropriate program device name must be specified on the write statement in the control area (depending on the language used) before issuing the output operation. Refer to examples in Chapter 9 through Chapter 11.
- The output length of the operation is determined by the specified record format. If you use user-defined formats, the record format length, as determined by the record definition in data description specifications (DDS), is the output length. You can vary this output length at run time by using the VARLEN DDS keyword. If you use system-supplied formats and data is allowed as part of the function, the output length is specified as part of the record format.
- If an output operation is issued with a zero record length, ICF assumes that the needed functions are only the functions indicated by the operation specified, and the data is not placed in the output buffer. For example, if a zero-length write operation is issued with a user-defined format with the INVITE keyword in effect, the program device is invited, but no data is sent to the remote system.
- Multiple output operations can be done with a single write operation. For example, if a write operation is issued with a user-defined format with the FMH and INVITE keywords in effect, or with the system-supplied $$SENDFM format, data is sent to the remote system with FMH information and the program device is invited. Refer to Chapter 6 to determine the processing sequence of the DDS keywords.
- If your program issues an output operation with an output length greater than the length supported for the session, the operation completes with a 831F return code. The maximum output length supported is determined by the communications type you are using, the record length specified in configuration, or the record length specified on the Add Intersystem Communications Function

Device Entry (ADDICFDEVE) or Override Intersystem Communications Function Device Entry (OVRICFDEVE) command.

- If your program issues an output operation while a session is in receive state, the operation may complete with an output exception (a major/minor return code of 0412). If the operation completes with an output exception, the data is not sent to the remote system. Your program must issue an input operation to receive the data or system message that is pending.
- If your program issues an output operation to an invited program device and the communications type you are using supports the cancel-invite function, ICF tries to cancel the invite. If the invite cannot be canceled, the output operation completes with a 0412 return code.
- When your program is reading data from a local source, such as a database file, it may not determine until the next read that it has just read the last record in the file. If this is the case and your program uses a specific indication, such as the end-of-group or detach function, to inform the remote system when the end-of-file indication is reached, then you must ensure that this end-of-file indication is sent with a zero-length record format. If not, then any data in the output buffer from a previous operation is sent to the remote system as user data (for example, the last record may be sent twice).

Input Considerations

You should consider the following when writing programs for ICF communications:

- Your program must examine the major/minor return code to determine if the input operation completed with data. A major code of 00 indicates data reception. A major code of 02 indicates that data was received, but the job is being canceled. A major code of 03 indicates that no data was received. Refer to Appendix B for a complete summary of major/minor return codes.
- When the program is communicating with multiple sessions, the appropriate program device name must be specified on the read statement in the control area (depending on the language used) before issuing the input operation. Refer to examples in Chapter 9 through Chapter 11.
- The input length of the operation is determined by the specified record format. If you use user-defined formats, the record format length, as determined by the record definition in DDS, is the input length. If you use the system-supplied QICDMF file, the input length is always 4096.
- When the actual length of the data received is less than the input length, the system pads the remainder of the record with blanks (except for basic conversations in APPC). A field in the I/O feedback area indicates the actual length of the data received from the remote system. Refer to Appendix C for a summary chart of the I/O feedback area.
- When your program issues an input operation that completes with a return code of 0000 or 0300:
  - The operation is successful.
  - Your program controls the session and can send data.

  When the operation completes with a return code of 0001 or 0301:
  - The operation is successful.
  - The remote program controls the session and your program should continue issuing input operations.

- Response indicators (RCVTRNRND, RCVDETACH, RCVCONFIRM, RCVNEGRSP, RCVCANCEL, RCVFMH, RCVFAIL, and RCVENDGRP) can be received either with data or without data (indicators only). Your program must examine the major/minor return code to determine if the record contains data.
- For most communications types, if an input operation is issued before an evoke is sent or after a detach function is sent or received, the operation completes with a 8327 return code.
- When processing input data, consideration should be given to detecting invalid data within a numeric field. The program can detect invalid data by verifying numeric field contents through program logic, or by permitting the system to detect a decimal-data error when the field is used in an arithmetic operation within the program.

  The format selection processing used (as determined by the FMTSLT parameter on the ADDICFDEVE or OVRICFDEVE command) determines how the record format name is selected to process incoming data.

  If the FMTSLT(*RECID) option is used, ICF returns the record format based on the data received from the remote system. Therefore, the proper record (the one that matches the data received) will be selected.

  If the layout of the record format is described within the program, the program logic determines the placement and types of fields within the record. The program is responsible for processing the data according to a specific record type.

Release, End-of-Session, and Close Considerations

You should consider the following when writing programs for ICF communications.

Release Considerations

The following are release operation considerations for ICF programs:

- If a target program issues a release operation, the logical connection between the current program and the communications session is ended. The communications session remains intact, and can, by using an acquire operation, be used again in the same job. The session state will be the same as it was when it was released.
- If a source program issues a release operation, the state of the operation is verified, and the release request can be rejected. If the program device is invited, the release operation is rejected with a 832C return code.

  If the release operation is successful, the communications session ends.

End-of-Session Considerations

The following are end-of-session function considerations for ICF programs:

- An end-of-session function is always successful if a session exists. The system ends the communications session regardless of the state of the session.
- If a target program issues an end-of-session function, the session is ended and cannot be reestablished with an acquire operation.
- A session remains allocated to the target program until an end-of-session function is performed or until the job ends.

- An end-of-session function issued to an unacquired device results in an 830B return code.

Close Considerations

The following are close operation considerations for ICF programs:

- If a close operation is issued to a shared file, the program issuing the close cannot do I/O operations to the file, but other programs that have the file open can still use the file. Refer to the *Data Management Guide* for more information on shared file processing.
- When a close operation is issued to an ICF file, all sessions associated with a source program for the specified file are ended regardless of the state of the session. Depending on the communications type, any buffered data may or may not be sent to the remote system.
- If a session is associated with a requesting program device, the logical connection between the current program and the communications session is ended. The communications session remains intact, and can, by using an acquire operation, be used again in the same job. The session remains intact. The state of the session remains the same when the program device is acquired again.
- Any active transactions associated with the file may be abnormally ended.

Remote Program Start Considerations

Program start requests that are received from a remote system result in an attempt to start the job. The program specified by the program start request runs within a routing step of the job. All jobs on the AS/400 system operate in an environment called a subsystem.

In order to receive program start requests from a remote system and to start a job to run the program specified on the program start request, a subsystem must be defined on the AS/400 system. You can define a new subsystem, or modify an IBM-supplied subsystem such as QBASE or QCMN, to receive and process program start requests.

Refer to the *Work Management Guide* for more information about QBASE and QCMN.

Defining the Environment

Subsystem descriptions are created using the Create Subsystem Description (CRTSBSD) command. The CRTSBSD command is described in the *Work Management Guide*.

The AS/400 system considers a communications device to be another source of work for a subsystem. Therefore, you must define a communications entry in the subsystem description to identify the communications devices and remote locations for which work (program start requests) can be received by the subsystem. Default communications entries are shipped with the system. However, you can change these entries with the following commands:

- Add Communications Entry (ADDCMNE) command
- Remove Communications Entry (RMVCMNE) command
- Change Communications Entry (CHGCMNE) command

You can specify a default user on the communications entry for a subsystem description on the DFTUSR parameter on the ADDCMNE or CHGCMNE command.

Refer to the *Communications User's Guide* for more information on the use of these commands.

Use the Add Routing Entry (ADDRTGE) command to define the routing information for remotely started jobs. You can specify that the program identified in the program start request be used. Or, you can define routing entries that select the program based on the device description name, mode name, or user profile name. Refer to the *Communications User's Guide* and the *Work Management Guide* for more information on the use of this command.

Handling Program Start Requests

When a subsystem receives a program start request from a remote system, it locates a user profile for the job based on one of the following:

- The password or user ID from the program start request (if one was specified)
- The user ID from the DFTUSR parameter of the ADDCMNE command

If the user ID and password are not passed as part of the program start request (for example, *NONE specified on the SECURITY keyword), the system tries to find a communications entry in the subsystem handling the program start request that allows a default user. If the entry is not found, the program start request is rejected with a security violation error.

The user profile contains the authorization to the objects and functions the job can reference.

The subsystem also locates a job description for the job. The job description defines job attributes, such as the job output queue and the first library list made. You can specify a job description on the ADDCMNE command or in the user profile.

The program name and the library name are passed as part of the program start request. This information is used along with the subsystem description routing entries to start a routing step and to select the program that starts running on the routing step.

The subsystem searches its routing table to determine the name of the program to be run in the job's routing step. In an interactive or batch environment, the routing data normally selects the program QSYS/QCMD, which processes control language (CL) commands. However, QCMD does not process data received with program start requests, and should not be selected as a communications target program. For remotely started jobs, the program to be run is commonly specified on the program start request from the remote system.

The subsystem also uses the routing entry to select a class for the job. The job class defines operation attributes, such as operation priority and time slice.

The system sends message CPF1269 to the QSYSOPR message queue if it is unable to start the requested program. The information in the message can help determine and correct the cause of the problem when working with the remote system programmer. Refer to Appendix B for additional information on the message generated for failed program start requests.

When the communications target job is started and the target program begins running, the target job can access any parameters specified on the evoke request as if they were parameters passed on a Call a Program (CALL) command. The target job communicates with the source program by opening an ICF file, acquiring the requesting program device, and issuing I/O requests to read and write data.

If program initialization parameters are passed on an evoke function or program start request, the following points should be noted:

- If multiple program initialization parameters are passed, the system uses commas to separate these parameters. Therefore, do not include commas in your program initialization parameter data.
- If you specify program initialization parameters with the evoke function, each parameter that is sent should be equal in length to the corresponding parameter specified in the target program. If it is longer than the parameter length in the target program, truncation occurs. If it is shorter than the parameter length in the target program, results may occur that cannot be predicted.

The target job runs as a normal batch job, and is subject to all job control commands, such as Display Job (DSPJOB), Work Job (WRKJOB), and End Job (ENDJOB). The job can be transferred using the Transfer Job (TFRJOB) command, but not the Transfer Batch Job (TFRBCHJOB) command.

Once the transaction ends (when a detach is successfully sent or received), the target program can no longer communicate with the source program. For most communications types, if the program is started with an evoke-with-detach function and you do an acquire operation to the requesting program device, the acquire fails with a 82A9 return code.

Figure 8-1 on page 8-9 shows a sample ICF communications environment.

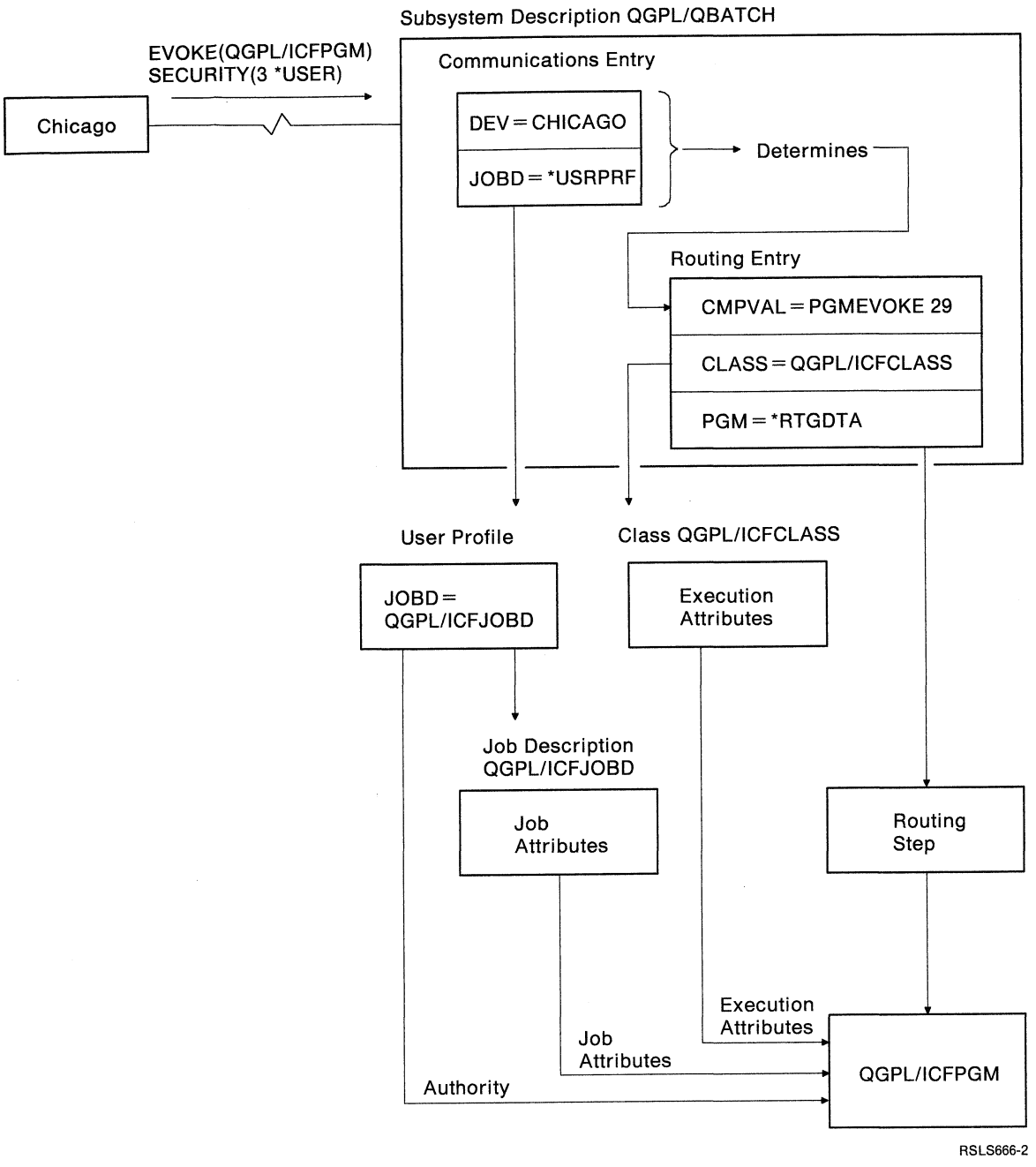


Figure 8-1. Sample ICF Communications Environment

Refer to the *Work Management Guide* for more information on job structures and job processing.

Prestarting Jobs for Program Start Requests

To minimize the time required to carry out a program start request, you can use the prestart job entry to start a job on the AS/400 system *before* the remote program sends a program start request. Because a job is already started and running before the program start request is received, improved program response time results.

You can use prestart jobs for all communications types that support program start request processing: APPC, asynchronous, BSCEL, intrasystem, finance, retail, and SNUF communications.

To use prestart jobs, you must define a prestart job entry and a communications entry in the subsystem description. Each prestart job entry contains a program name, library name, user profile, and other attributes, which the subsystem uses to create and manage a pool of prestart jobs.

Both the communications and prestart job entries must be specified in the same subsystem. If a program start request is received on a subsystem that does not have a prestart job entry with the matching program name, the usual processing required to start a communications batch job occurs when the program start request is received. The subsystem attempts to allocate the communications devices that are identified by the communications entries. When a program start request is received, it is sent to the subsystem that has the required communications device allocated. For more information about adding, changing, and removing communications entries in subsystem descriptions, refer to the *Communications User's Guide.*

When a subsystem is initially started, prestart jobs are started based on the information contained in the prestart job entries. Each prestart job entry identifies the program that is to be started, the number of jobs that need to be started with the program, and the user profile under which the jobs will run when it is *not* servicing a program start request. However, when a program start request is received, the subsystem determines if the program name sent on the program start request matches the program name on one of the prestart job entries. If so, the subsystem ensures that the program start request user ID and password are valid and that the user is authorized to use the device and the library and program. The program start request is then attached to a prestart job. The prestart job runs under the user profile specified on the program start request while it is servicing that request.

Commands

Prestart jobs can start at the same time the subsystem is started, or you can use the Start Prestart Jobs (STRPJ) command to start jobs for a prestart job entry in an active subsystem.

The Change Prestart Job (CHGPJ) command allows you to change some of the attributes for a prestart job based on either the attributes of the job description for a prestart job entry or the attributes of the job description defined in the user profile of a program start request.

The End Prestart Job (ENDPJ) command allows you to end all jobs for a prestart job entry in an active subsystem.

The following commands should be used when working with prestart job entries in the subsystem description:

- The Add Prestart Job Entry (ADDPJE) command adds a prestart job entry to the specified subsystem description.
- The Change Prestart Job Entry (CHGPJE) command changes a prestart job entry in the specified subsystem description.
- The Remove Prestart Job Entry (RMVPJE) command removes a prestart job entry from the specified subsystem description.
- The Display Active Prestart Jobs (DSPACTPJ) command displays the run time statistics and performance information for prestart jobs associated with a prestart job entry in an active subsystem.

For more information about the parameters and attributes associated with these commands, refer to the *CL Reference*.

Application Considerations

Certain programming changes need to be made to the prestart job program to allow it to be started by and communicate with the remote program. The following points must be taken into account when writing prestart job applications:

- A prestart job program should do as much work as possible, for example, allocating objects and opening database files, before attempting to acquire a requesting program device. Once a prestart job is started, this initial processing is done before the acquire of a requesting program device. The acquire operation causes the job to wait until a program start request is received. When a program start request is received, the program then continues with the acquire operation.
- When a prestart job program is done servicing a program start request, it must do an end-of-session function followed by the acquire of a requesting program device. This is the only way the prestart job makes itself available for the next program start request. If a release operation is performed instead of an end-of-session function, the acquire of the requesting program device does not cause the program to wait, and the program device continues to run on the current session.
- Because a job is already started and running before a program start request with program initialization parameters is received, the subsystem stores these parameters in the program initialization parameter data area for the prestart job to which the program start request attaches. After the acquire of the requesting program device, you must use the Retrieve Data Area (RTVDTAARA) command (specifying *PDA as the data area) to retrieve the program initialization parameters (if any) passed on the program start request.

  Note: A maximum length of 2000 bytes is allowed for program initialization parameters passed on a program start request for a prestart job.
- Only resources that are used specifically for a transaction should be deallocated. Any resource that is commonly used for most transactions performed by the prestart job program should remain allocated while the job is waiting for the next request to be received.
- When a program start request attaches to a prestart job, none of the attributes associated with the user profile on the program start request are used. To change the attributes for a job to those of the job description on the user profile specified on the program start request, use the CHGPJ command.

See Figure 8-2 on page 8-13 for a sample prestart job program. For information about sharing database files in the same job and across jobs, see the *Database Guide*.

Security Considerations for Prestart Jobs

When a prestart job is initially started, authority checking for a prestart job entry user profile is performed on every object that is needed to run the job. When a program start request attaches to a prestart job, however, it runs under the user profile specified on the program start request. Before a program start request is allowed to attach to a prestart job, only the program start request user ID and password and its authority to the communications device, library, and program are checked. To avoid cases where the program start request user profile is not authorized to objects to which the prestart job entry user profile is authorized, you should ensure that the user profile specified on the program start request is authorized to at least as many objects as that on the prestart job entry.

To accomplish this, you can do one of the following:

- Create your prestart job program when you are running under the prestart job entry user profile, and specify the value *OWNER for the user profile on the CRTXXXPGM command for your prestart jobs program, where XXX is the program language you are using (C/400, COBOL/400, or RPG/400).
- Explicitly check for object authorization (using the Check Object (CHKOBJ) command) before you change or access any objects.

Files and objects to which a prestart job entry user profile is not authorized should be deallocated before you end your transaction.

Prestart Job Program

Figure 8-2 on page 8-13 is a COBOL/400 prestart jobs program that can be used to handle program start requests. The initial processing, for example, opening the ICF file and printer file, is done first before the acquire of a requesting program device. The acquire operation then causes the prestart job program to wait for a program start request. After the acquire of the requesting program device, a CL program is called to retrieve program initialization parameters (using the RTVDTAARA command) and to change some of the job attributes based on the program start request user profile (using the CHGPJ command). After the main body of the program is done processing, the communications session is ended, and the program loops back to the acquire operation and waits for the next program start request to be received.

```
5728CB1 R01 M02 881028              IBM AS/400  COBOL/400          QGPL/PJPGM            03/01/89 12:11:29       PAGE    1
PROGRAM NAME . . . . . . . . . . . . : PJPGM IN QGPL
SOURCE FILE  . . . . . . . . . . . . : SFX IN QNETUSER     MEMBER - PJPGM     03/01/89 12:02:39
COMPILER OPTION  . . . . . . . . . . : *NONE
CODE GENERATION OPTION . . . . . . . : *NONE
CODE GENERATION SEVERITY LEVEL . . . : 29
PRINT FILE . . . . . . . . . . . . . : QSYSPRT IN *LIBL
FIPS FLAGGING OPTION . . . . . . . . : *NOFIPS *NOSEG *NODEB *NOOBSOLETE
TEXT NOT AVAILABLE FOR MESSAGE LBX6039 FILE QLBLMSG.
FLAGGING LEVEL . . . . . . . . . . . : 0
REPLACE EXISTING PROGRAM . . . . . . : *YES
USER PROFILE . . . . . . . . . . . . : *OWNER
AUTHORITY  . . . . . . . . . . . . . : *CHANGE
TEXT . . . . . . . . . . . . . . . . : SAMPLE  PROGRAM
COMPILER . . . . . . . . . . . . . . : IBM AS/400  COBOL/400
                                                 ENDJOBNOLIST
    1  000100 IDENTIFICATION DIVISION.                                                    01/26/89
    2  000200 PROGRAM-ID.               PJPGM.                                            03/01/89
       000300*************************************************************************   01/26/89
       000400*  THIS IS A PRESTART JOB TARGET PROGRAM                              *      03/01/89
       000500*  THIS PROGRAM WILL LOOP FOREVER UNTIL IS IT IS ENDED.               *      03/01/89
       000600*************************************************************************   01/26/89
    3  000700 ENVIRONMENT DIVISION.                                                       01/26/89
    4  000800 CONFIGURATION SECTION.                                                      01/26/89
    5  000900 SOURCE-COMPUTER.          IBM-AS400.                                        01/26/89
    6  001000 OBJECT-COMPUTER.          IBM-AS400.                                        01/26/89
    7  001100 SPECIAL-NAMES.            I-O-FEEDBACK IS IO-FEEDBACK                       01/26/89
    8  001200                           OPEN-FEEDBACK IS OPEN-FBA.                        01/26/89
    9  001300 INPUT-OUTPUT SECTION.                                                       01/26/89
   10  001400 FILE-CONTROL.                                                              01/26/89
       001500*************************************************************************   01/26/89
       001700*                F I L E   S P E C I F I C A T I O N S              *      01/26/89
       001900*    ICFFILE :  ICF FILE                                           *      01/26/89
       002100*    QSYSPRT:   PRINTER FILE                                       *      03/01/89
       002300*************************************************************************   01/26/89
   11  002400     SELECT ICFFILE    ASSIGN TO WORKSTATION-PJDDS-SI                       03/01/89
   12  002500         ORGANIZATION IS TRANSACTION                                       01/26/89
   13  002600         FILE STATUS IS STATUS-IND MAJ-MIN.                                03/01/89
   14  002700     SELECT QPRINT ASSIGN TO PRINTER-QSYSPRT.                              01/26/89
   15  002800 DATA DIVISION.                                                             01/26/89
   16  002900 FILE SECTION.                                                              01/26/89
   17  003000 FD  ICFFILE                                                                01/26/89
   18  003100     LABEL RECORDS ARE STANDARD.                                            01/26/89
   19  003200 01  ICFREC.                                                               01/26/89
   20  003300     COPY DDS-ALL-FORMATS-I-O OF PJDDS.                                     03/01/89
   21 +000001       05  PJDDS-RECORD PIC X(4101).                      <-ALL-FMTS
      +000002*    I-O FORMAT:INPFMT     FROM FILE PJDDS      OF LIBRARY QGPL          <-ALL-FMTS
      +000003*                                                                         <-ALL-FMTS
   22 +000004       05  INPFMT        REDEFINES PJDDS-RECORD.                        <-ALL-FMTS
   23 +000005           06 INPDATA           PIC X(4096).                            <-ALL-FMTS
      +000006*    I-O FORMAT:DETACH     FROM FILE PJDDS      OF LIBRARY QGPL          <-ALL-FMTS
      +000007*                                                                         <-ALL-FMTS
      +000008*      05  DETACH        REDEFINES PJDDS-RECORD.                        <-ALL-FMTS
      +000009*    I-O FORMAT:EOS        FROM FILE PJDDS      OF LIBRARY QGPL          <-ALL-FMTS
      +000010*                                                                         <-ALL-FMTS
      +000011*      05  EOS           REDEFINES PJDDS-RECORD.                        <-ALL-FMTS
      +000012*    I-O FORMAT:INVITE     FROM FILE PJDDS      OF LIBRARY QGPL          <-ALL-FMTS
      +000013*                                                                         <-ALL-FMTS
      +000014*      05  INVITE        REDEFINES PJDDS-RECORD.                        <-ALL-FMTS
      +000015*    I-O FORMAT:FAIL       FROM FILE PJDDS      OF LIBRARY QGPL          <-ALL-FMTS
      +000016*                                                                         <-ALL-FMTS
      +000017*      05  FAIL          REDEFINES PJDDS-RECORD.                        <-ALL-FMTS
      +000018*  INPUT FORMAT:IOFMT      FROM FILE PJDDS      OF LIBRARY QGPL          <-ALL-FMTS
      +000019*                                                                         <-ALL-FMTS
   24 +000020       05  IOFMT-I       REDEFINES PJDDS-RECORD.                        <-ALL-FMTS
   25 +000021           06 IODATA           PIC X(4096).                            <-ALL-FMTS
```

Figure 8-2 (Part 1 of 3). COBOL/400 Coding for a Prestart Job Program

```
    +000022* OUTPUT FORMAT:IOFMT      FROM FILE PJDDS      OF LIBRARY QGPL            <-ALL-FMTS
                                                ENDJOBNOLIST
    +000023*                                                                        <-ALL-FMTS
 26 +000024       05  IOFMT-O       REDEFINES PJDDS-RECORD.                       <-ALL-FMTS
 27 +000025           06 IODATA            PIC X(4096).                           <-ALL-FMTS
 28 +000026           06 OUTLEN            PIC S9(5).                             <-ALL-FMTS
 29  003400 FD  QPRINT                                                                    01/26/89
 30  003500     LABEL RECORDS ARE OMITTED.                                                01/26/89
 31  003600 01  PRINTREC.                                                                 01/26/89
 32  003700     05  PRTNOTE                      PIC X(132).                             03/01/89
 33  003800 WORKING-STORAGE SECTION.                                                      01/26/89
 34  003900 77  STATUS-IND                       PIC X(2).                               03/01/89
 35  004000 01  MAJ-MIN.                                                                 01/26/89
 36  004100     05  MAJ                          PIC X(2).                               01/26/89
 37  004200     05  MIN                          PIC X(2).                               01/26/89
 38  004300 01  ICF-FEEDBACK.                                                            01/26/89
 39  004400     05  FILLER                       PIC X(149).                             01/26/89
 40  004500     05  ACTUAL-LEN                   PIC 9(5) COMP-4.                        01/26/89
     004600*                                                                             01/26/89
 41  004700 PROCEDURE DIVISION.                                                           01/26/89
     004800                                                                               02/24/89
     004900*                                                                             01/26/89
     005000 START-PROGRAM-PARAGRAPH.                                                      01/26/89
     005100*************************************************************************     03/01/89
     005200* OPEN ICF FILE AND PRINTER FILE                                       *      03/01/89
     005300*************************************************************************     03/01/89
 42  005400     OPEN OUTPUT QPRINT.                                                      01/26/89
 43  005500     OPEN I-O   ICFFILE.                                                      01/26/89
     005600 MAIN-LOOP.                                                                    01/26/89
 44  005700     MOVE "WAITING FOR TRANSACTION" TO PRTNOTE.                               01/26/89
 45  005800     WRITE PRINTREC.                                                          01/26/89
     005900*************************************************************************     03/01/89
     006000* ACQUIRING THE REQUESTER PROGRAM DEVICE CAUSES THIS PRESTART      *          03/01/89
     006100* JOB PROGRAM TO WAIT FOR A PROGRAM START REQUEST.                 *          03/01/89
     006200*************************************************************************     03/01/89
 46  006300     ACQUIRE "REQDEVICE " FOR ICFFILE.                                        03/01/89
     006400*************************************************************************     03/01/89
     006500* IF THIS PRESTART JOB IS BEING ENDED WITH THE CONTROLLED OPTION,  *          03/01/89
     006600* PERFORM END-OF-JOB PROCESSING AND EXIT.                          *          03/01/89
     006700*************************************************************************     03/01/89
 47  006800     IF MAJ-MIN = "8209" GO TO END-JOB.                                       01/30/89
 49  006900     MOVE "TRANSACTION ATTACHED" TO PRTNOTE.                                  01/26/89
 50  007000     WRITE PRINTREC.                                                          01/26/89
     007100*************************************************************************     03/01/89
     007200* CALL A CL PROGRAM TO:                                            *          03/01/89
     007200*   - RETRIEVE PIP DATA FROM THE *PDA DATA AREA (RTVDTAARA).       *          03/01/89
     007300*   - CHANGE TO USE SOME JOB ATTRIBUTES FROM THE PROGRAM START     *          03/01/89
     007300*     REQUEST USER PROFILE (CHGPJ).                                *          03/01/89
     007400*************************************************************************     03/01/89
 51  007500     CALL "CLPGM".                                                            03/01/89
     007600*************************************************************************     03/01/89
     007700* MAIN BODY OF THE PROGRAM, THAT DOES THE COMMUNICATIONS           *          03/01/89
     007800* I/O WITH THE SOURCE PROGRAM, SHOULD BE PLACED HERE.              *          03/01/89
     007900*************************************************************************     03/01/89
     008000*************************************************************************     03/01/89
     008100* WHEN THE MAIN BODY OF THE PROGRAM IS DONE PROCESSING, END        *          03/01/89
     008200* THE COMMUNICATIONS SESSION AND LOOP BACK TO WAIT FOR THE NEXT    *          03/01/89
     008300* PROGRAM START REQUEST TO COME IN.                                *          03/01/89
     008400*************************************************************************     03/01/89
 52  008500     WRITE ICFREC FORMAT IS "EOS".                                            01/26/89
                                          ENDJOBNOLIST
```

Figure 8-2 (Part 2 of 3). COBOL/400 Coding for a Prestart Job Program

```
    53  008600     GO TO MAIN-LOOP.                                          01/26/89
        008700 END-JOB.                                                      01/30/89
    54  008800     MOVE "PRESTART JOB BEING ENDED CONTROLLED" TO PRTNOTE.    01/30/89
    55  008900     WRITE PRINTREC.                                           01/30/89
    56  009000     CLOSE  ICFFILE.                                           01/26/89
    57  009100     CLOSE  QPRINT.                                            01/26/89
    58  009200     STOP RUN.                                                 01/30/89
        009300 MAIN-EXIT.                                                    01/26/89
        009400     EXIT.                                                     01/26/89
                          * * * * *  E N D  O F  S O U R C E  * * * * *
                                                ENDJOBNOLIST
  STMT
*   20  MSGID: LBL0600  SEVERITY: 10  SEQNBR:  003300
        MESSAGE . . . . :   NO INPUT FIELDS FOUND FOR FORMAT DETACH.
*   20  MSGID: LBL0600  SEVERITY: 10  SEQNBR:  003300
        MESSAGE . . . . :   NO INPUT FIELDS FOUND FOR FORMAT EOS.
*   20  MSGID: LBL0600  SEVERITY: 10  SEQNBR:  003300
        MESSAGE . . . . :   NO INPUT FIELDS FOUND FOR FORMAT INVITE.
*   20  MSGID: LBL0600  SEVERITY: 10  SEQNBR:  003300
        MESSAGE . . . . :   NO INPUT FIELDS FOUND FOR FORMAT FAIL.
                                        MESSAGE SUMMARY
  TOTAL    INFO(0-4)    WARNING(5-19)    ERROR(20-29)    SEVERE(30-39)    TERMINAL(40-99)
     4          0             4               0               0                0
                  * * * * *  E N D  O F  C O B O L  M E S S A G E S  * * * * *
 94 SOURCE RECORDS READ
 26 COPY RECORDS READ
 1 COPY MEMBERS PROCESSED
 0 SEQUENCE ERRORS
 10 WAS THE HIGHEST SEVERITY MESSAGE ISSUED
  LBL0901 00  PROGRAM PJPGM CREATED IN LIBRARY QGPL.
                       * * * * *  E N D  O F  C O M P I L A T I O N  * * * * *
                                                ENDJOBNOLIST
```

Figure 8-2 (Part 3 of 3). COBOL/400 Coding for a Prestart Job Program

System Considerations

The specification of the PURGE parameter on the Create Class (CRTCLS) command for the routing entry used for communications can affect communications performance. The PURGE parameter controls the way the job's operating resources are used when the job enters a wait state. If PURGE(*YES) is specified, the job's operating resources are exchanged to auxiliary storage when the job enters a wait state. If PURGE(*NO) is specified, the job's operating resources are not exchanged to auxiliary storage when the job enters a wait state.

For communications, PURGE(*YES) is normally used for interactive applications because the application is normally waiting for a response from a work station operator. The wait delay time can be long enough for another job to do useful processing with the resources. PURGE(*NO) is normally used for batch applications when the AS/400 system is sending data to or receiving data from another computer and the wait delay will be nominal.

For a complete description of these parameters or the CRTCLS command, see the *Work Management Guide*.

Security Considerations

The security provided on the AS/400 system is used to control who can use a communications device description and its associated commands.

When a program issues an evoke to a remote AS/400 system, you must ensure that proper security information is passed on the evoke request. See Chapter 6 and Chapter 7 for a discussion on how to specify the security information on the evoke.

Although the security officer or service user has all object authority, explicit authorization to a target communications device description must be made. For example, if the security officer or service user has not been explicitly granted authority to the target device description, the program start request is rejected by the target system. This aspect of system security is consistent with the work station security implementation for the security officer and service user profiles. Anyone with object management authority for a device description can grant authority for the device description to the security officer or service user. If the security officer or service user creates a device description, the security officer or service user (like anyone else who creates device descriptions) is explicitly authorized to the device description. When a communications device description is authorized to all users (*ALL), the security officer and service user are not included. This allows the security officer or service user to specify the device description from which the security officer or service functions can be performed.

Refer to the appropriate communications programming manual for more information about security considerations.

File Considerations

You should consider the following when deciding whether to use the system-supplied QICDMF file in your program:

- System-supplied formats can be used either with the system-supplied QICDMF file or with an ICF file made using DDS processing keywords and the CRTICFF command.
- User-defined formats cannot be added to the QICDMF file. Therefore, if your program uses the QICDMF file, it cannot use DDS processing keywords or externally described data.
- If your program uses an ICF file created using DDS and the CRTICFF command, your program can use both the record formats defined as part of your ICF file and system-supplied formats. The high-level language you are using may have some restrictions on mixing system-supplied formats and user-defined formats.

File Redirection

You can override ICF files by using the override with file commands.

When you change the file used in a program without changing the file type, the new file being used in the program is processed in the same manner as the original file. The format levels in the file must agree with the compiled program if level checking is done. If level checking is not done, the format of the changed-to file must be compatible with the compiled program or the results cannot be predicted.

If you change from one file type to another, the file-dependent characteristics of the file are ignored and certain defaults are used.

For a complete description of file redirection and the defaults used, see the *Data Management Guide*.

Additional Considerations

When using a particular communications type, you must be familiar with the requirements and restrictions unique to that communications type. For specific details, refer to the appropriate programming manual for the communications type you are using.

Chapter 9. Communications Applications with C/400

Previous chapters in this manual describe the functions provided by ICF. This chapter introduces you to the C/400 interface for ICF and provides program examples.

One program example is presented in this chapter, and both the source and target programs are provided. The example is a multiple-session inquiry application using one display file and four ICF sessions, and is written using user-defined formats (data description specifications or DDS).

Not all programming considerations or techniques are illustrated in the examples in this chapter. Review these examples and the examples provided in the appropriate programming manual before beginning application design and coding.

Note: The examples in this chapter were written to the APPC communications type. Minor changes might be required if another communications type is used.

Introduction to the C/400 Interface

Before you write a C/400 communications application, you must understand the high-level language interface provided by C/400.

The operations you use in the communications portion of your program are similar to work station operations. In the noncommunications portion of your program, you can use all noncommunications operations you normally use to process data that is sent or received between your program and the remote program.

Table 9-1 briefly introduces the C/400 statements you use in the communications portion of your program.

Note: C/400 statements are case sensitive.

Table 9-1. C/400 Statements

ICF Operation	C/400 Statement	Function
Open	fopen	Opens the ICF file
Acquire	QXXACQUIRE	Establishes a session
Get-Attributes	QXXDEVAT	Gets the attributes of a session
Read	fread	Receives data from a specific program device
Read-from-invited-program-devices	QXXREADINVDEV, followed by an fread	Receives data from any invited program device[1]
Write	fwrite	Performs many of the ICF communications functions within a session
Release	QXXRELEASE	Releases the session
Close	fclose	Closes the ICF file

[1] The read-from-invited-program-devices operation could complete without data if the timer interval established with either the timer function or WAITRCD ends, or your job is ended (controlled).

Refer to the *C/400 User's Guide* for details on the syntax and function of each operation.

Multiple-Session Inquiry

This example illustrates an interactive inquiry application that communicates with multiple ICF sessions. A source AS/400 system program accepts inquiries from a display device and sends a request to one of four AS/400 systems. The source program communicates with the display device through a display file, and with the four remote systems through a single ICF file.

The purpose of this example is to show multiple sessions from a single ICF file. The source program communicates with four sessions. From the viewpoint of each of the four target programs, there is only one session (with the requesting program device). Therefore, the target programs do not require any unique logic to support the multiple-session source.

Both the source program and the target program are described. The same target program is evoked in each of the four separate remote systems. Therefore, only one target program is shown in the programming example.

Transaction Flow of the Multiple-Session Inquiry

The program shown in Figure 9-1 is started from a display station, and both the display and the ICF file are opened. CIWS00 is the *REQUESTER device, acquired when the display file opens. CIWS00 is acquired because DEV(*REQUESTER) was specified when the display file was created. Since the ICF file was created with ACQPGMDEV(*NONE), no ICF program devices are acquired during open processing.

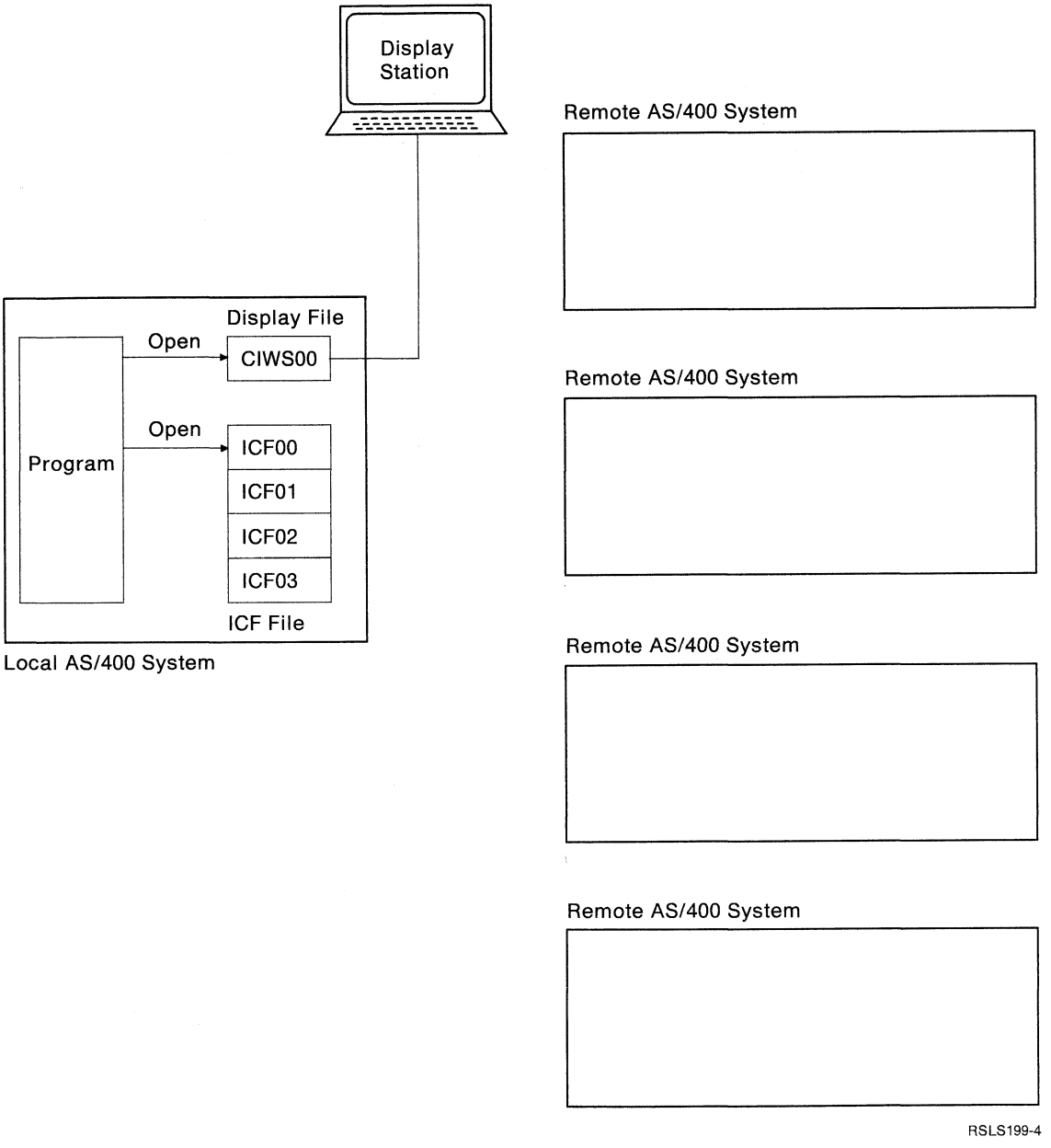


Figure 9-1. Program Starts at Display Station

All other program devices are explicitly acquired by the program, as shown in Figure 9-2.

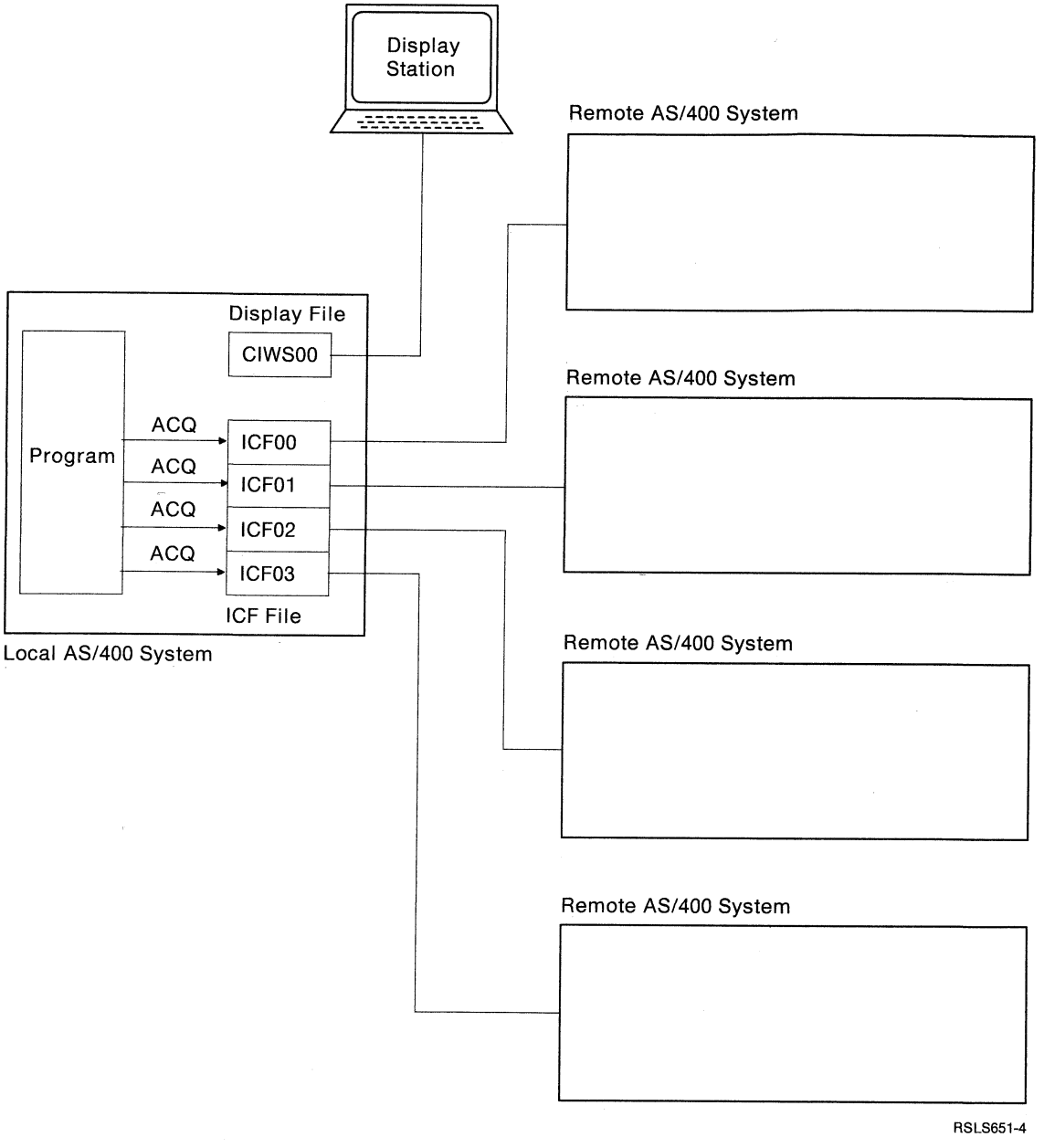


Figure 9-2. Program Devices Explicitly Acquired

All target programs are started with the evoke, as shown in Figure 9-3.

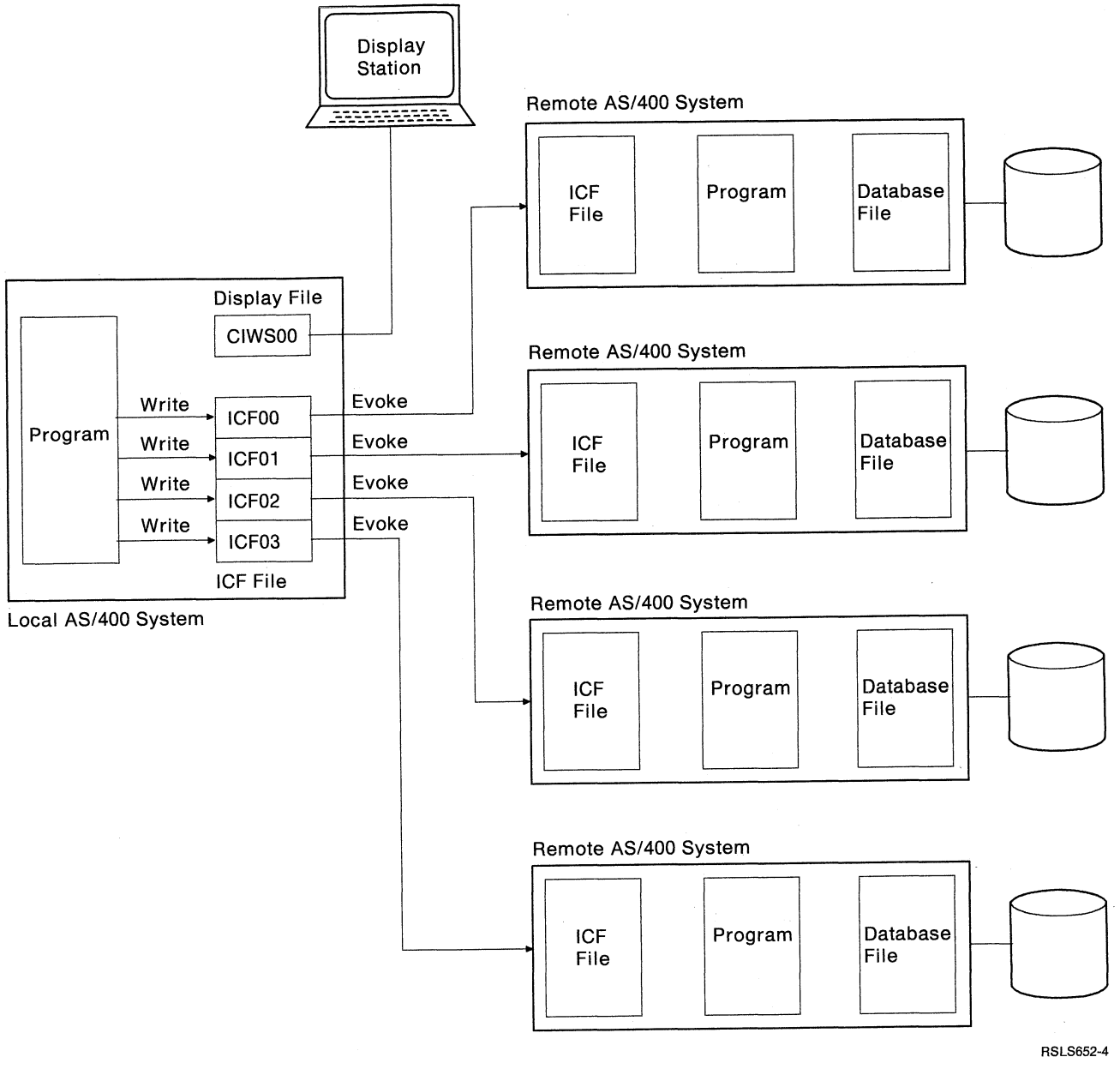


Figure 9-3. Evoke Starts Target Programs

The source program uses a specific program device name. Each target program uses an ICF file with a program device name that is associated with the requesting program device. The target program's only session is the one used to communicate with the source program. The ICF file on the remote system must be opened by the C/400 language support using the open operation, and the requesting program device is acquired when the file is opened using the acquire operation.

The main menu is written to the display station on the local system, and the program waits for a request from the display station, as shown in Figure 9-4.

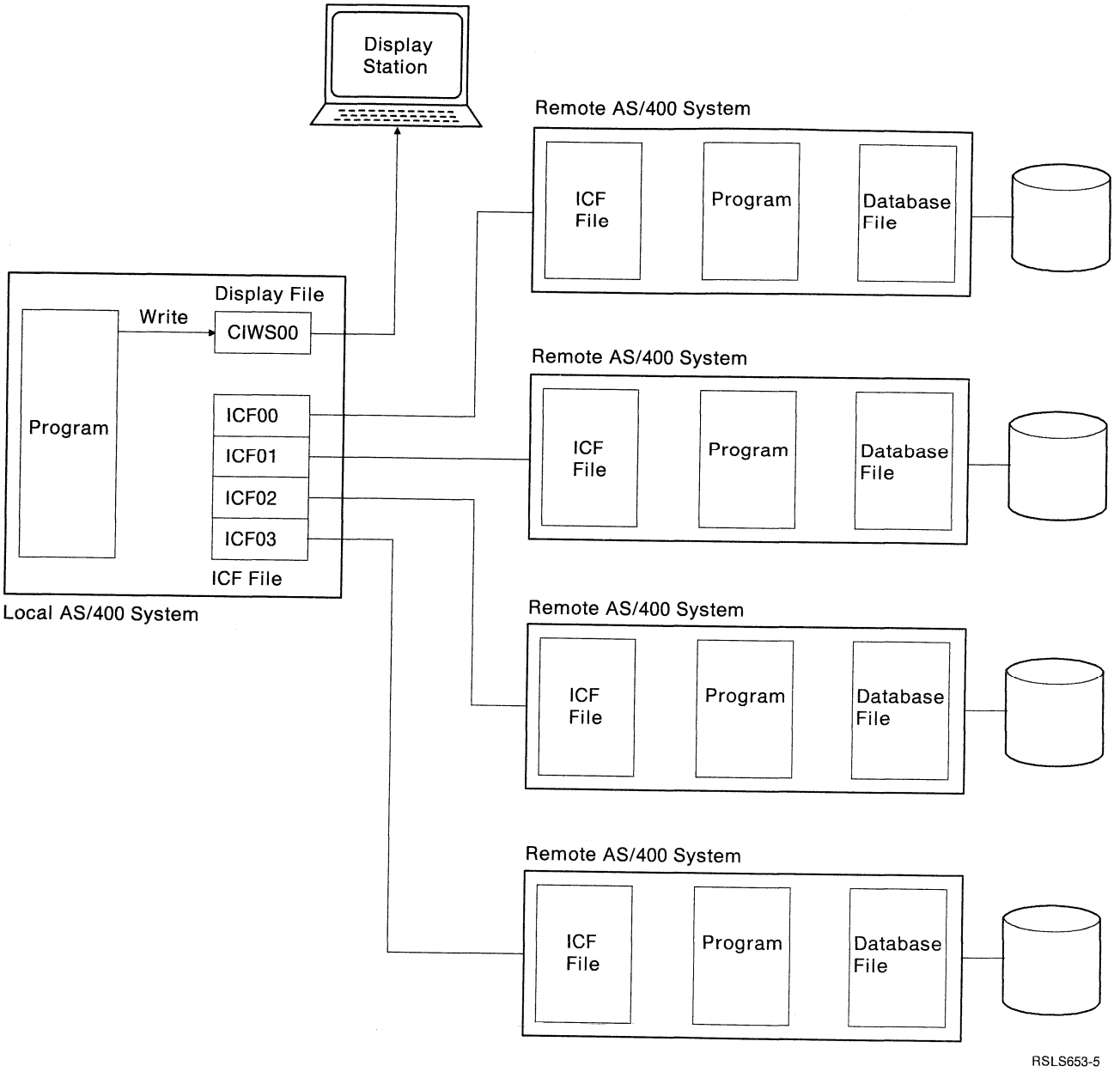


Figure 9-4. Main Menu Written to Display Station

The source program sends an inquiry request to one of the remote systems based on the request made from the display station, as shown in Figure 9-5.

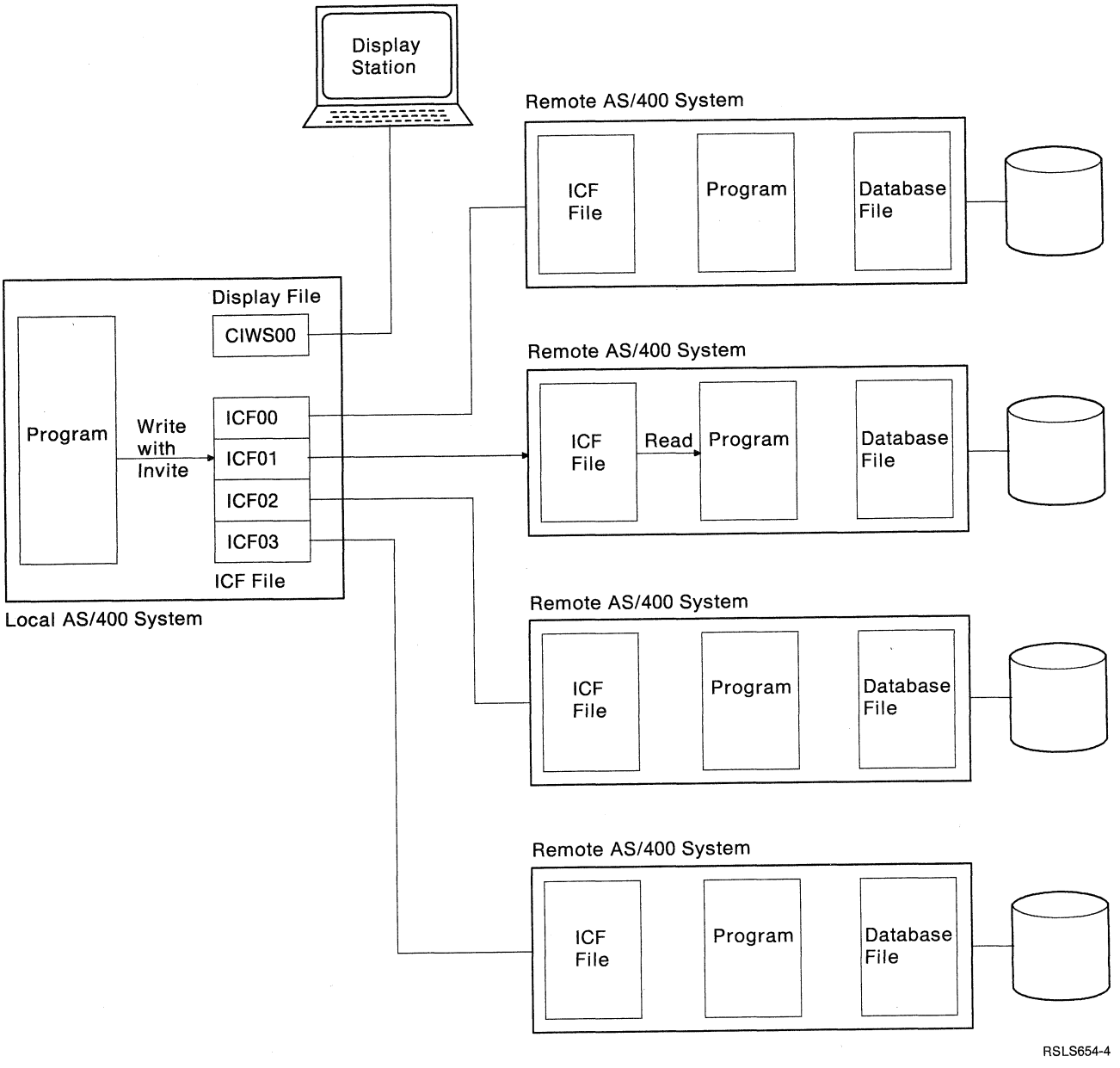


Figure 9-5. Program Sends Inquiry Request to Remote System

The target program responds to the inquiry by sending a reply, as shown in Figure 9-6.

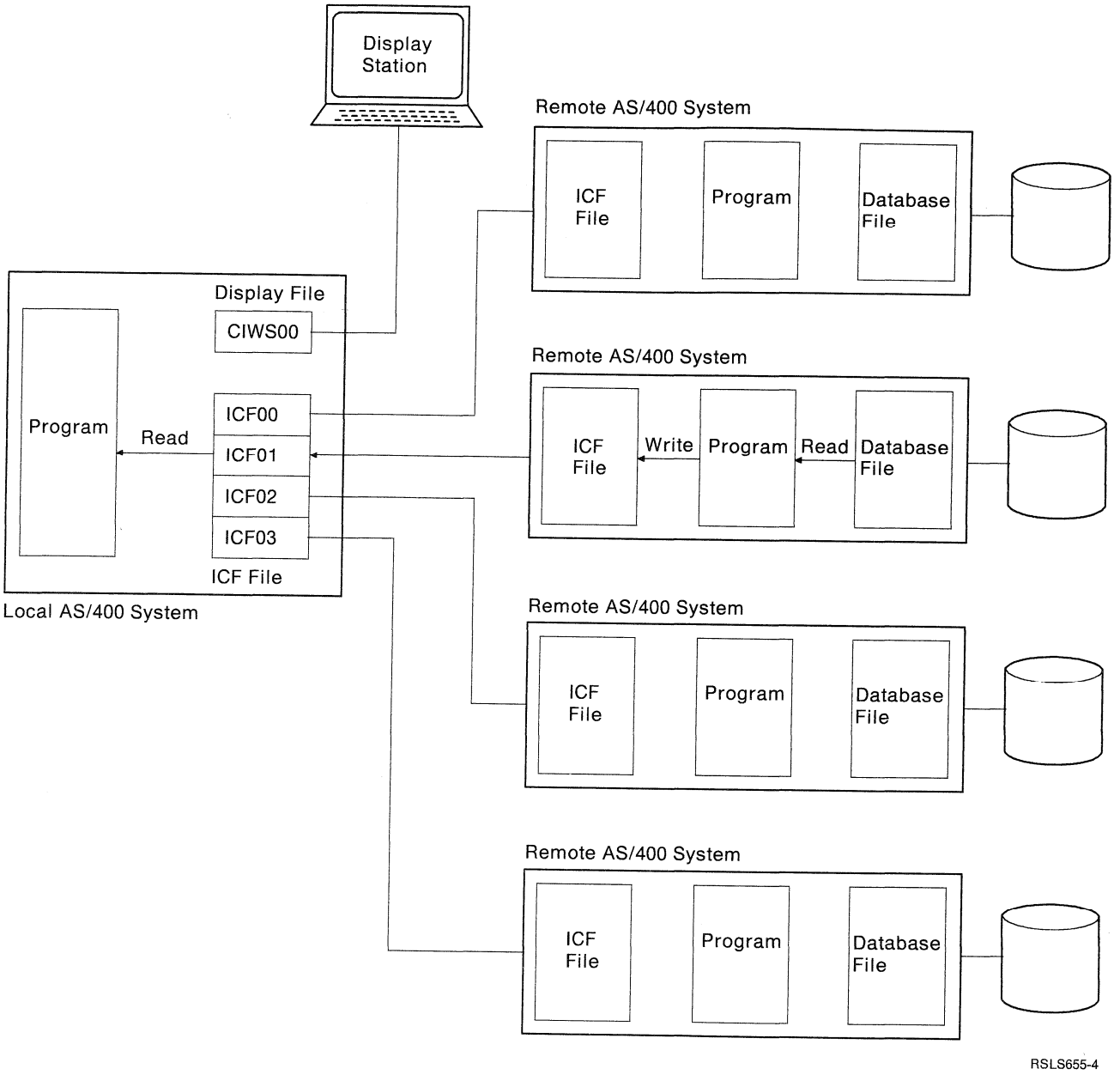


Figure 9-6. Target Program Sends a Reply

The program sends a detach request and ends the session when function key 1 is pressed (while the main inquiry menu is present), as shown in Figure 9-7.

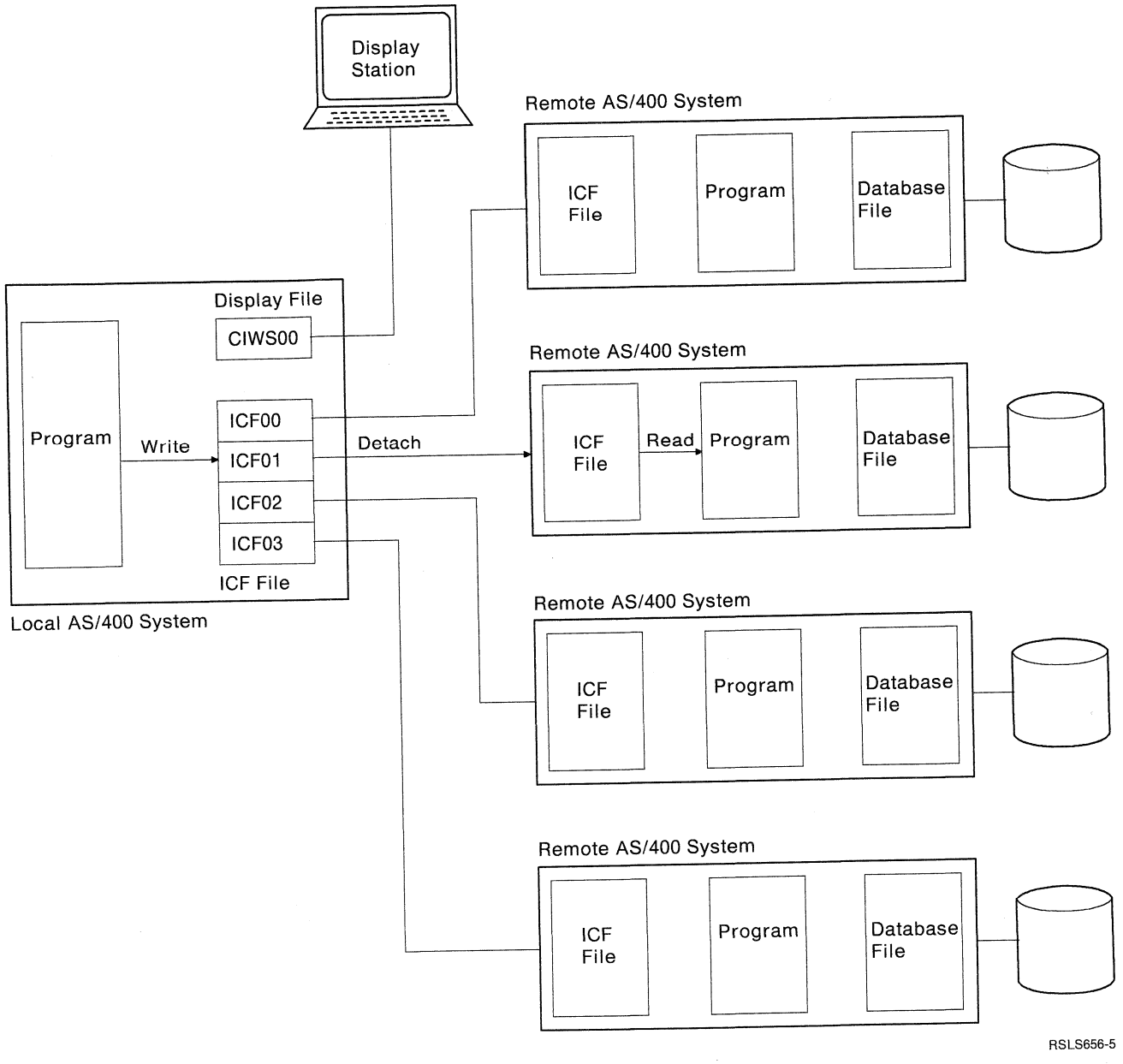


Figure 9-7. Program Ends the Session

Source Program Multiple-Session Inquiry

The following describes a C/400 source program multiple-session inquiry.

Program Files: The C/400 multiple session source program uses the following files:

CMNFIL — An ICF file used to send records to and receive records from the target program.

DSPFIL — A display file used to enter requests that are to be sent to the target program.

QSYSPRT — A printer file used to print error messages resulting from communications errors.

DDS Source: The DDS for the ICF file (CMNFIL) is illustrated in Figure 9-8 on page 9-11.

```
A*****************************************************************
A*                                                               *
A*                         ICF FILE                              *
A*          USED IN SOURCE MULTIPLE SESSION PROGRAM              *
A*                                                               *
A*****************************************************************
A                                      INDARA
A           R ITMRSP
A                                      RECID(1 'I')
A             RECITM         1
A             ITEMNO         6  0
A             DESC          30
A             QTYLST         7  0
A             QTYOH          7  0
A             QTYOO          7  0
A             QTYBO          7  0
A             UNITQ          2
A             PR01           7  2
A             PR05           7  0
A             UFRT           5  2
A             SLSTM          9  2
A             SLSTY         11  2
A             CSTTM          9  2
A             CSTTY         11  2
A             PRO            5  2
A             LOS            9  2
A             FILL1         56
A           R DTLRSP
A                                      RECID(1 'C')
A             RECCUS         1
A             CUSTNO         6  0
A             DNAME         30
A             DLSTOR         6  0
A             DSLSTM         9  0
A             DSPM01         9  0
A             DSPM02         9  0
A             DSPM03         9  0
A             DSTTYD        11  0
A             IDEPT          3  0
A             FILL2         57
A           R DETACH
A                                      DETACH
A           R EOS
A                                      EOS
A           R EVKREQ
A                                      EVOKE(CICFLIB/CTGTDMULCL)
A                                      SECURITY(2 *USER 3 *USER)
A           R ITMREQ
A                                      INVITE
A             ITEMNO         6  0
A           R DTLREQ
A                                      INVITE
A             CUSTNO         6  0
```

Figure 9-8. DDS for Source Program Multiple-Session Inquiry Using CMNFIL

The DDS for the display file (DSPFIL) is illustrated in Figure 9-9.

```
A*****************************************************************
A*                                                               *
A*                        DISPLAY FILE                           *
A*           USED IN SOURCE MULTIPLE SESSION PROGRAM             *
A*                                                               *
A*****************************************************************
A*  BEGINNING MENU
A*********************
A                                      INDARA
A                                      DSPSIZ(*DS3)
A                                      CF01(99) CF02(98) CF03(97)
A          R CIMENU                    TEXT('MENU FOR INQUIRY')
A                                   1 34'INQUIRY MENU'
A                                    3  1'Select one of the following:'
A                                    4  3'1. Item inquiry'
A                                    5  3'2. Customer inquiry'
A                                   11  1'Option:'
A            OPTION         1N  I 11  9VALUES('1' '2')
A                                   19  5DFT('CMD KEY 1 - END ')
A          R DTLMNU                      TEXT('CUSTOMER INQUIRY SCREEN 1')
A                                    2  2DFT('ENTER CUSTOMER')
A            CUSTNO         6N 0I  2 20
A                                   19  5DFT('CMD KEY 1 - END ')
A                                   19 23DFT(' 2 - MAIN MENU ')
A*
A****************************
A*  CUSTOMER INQUIRY SCREEN
A****************************
A          R DTLSCR                      TEXT('CUSTOMER INQUIRY SCR. #2')
A                                    1  3DFT('CUST  DPT LAST ORD   & THIS   +
A                                         $MTH1     &MTH2     $MTH3     THIS+
A                                         YTD NAME')
A            CUSTN          6N     2  2
A            DEPT           3N 0   2  9
A            DLSTR          6N 0   2 13
A            DSLSM          9N 0   2 22
A            DSPM1          9N 0   2 32
A            DSPM2          9N 0   2 42
A            DSPM3          9N 0   2 52
A            DSTYD         11N 0   2 62
A            CNAME         30      2 74
A                                  19  5DFT('CMD KEY 1 - END ')
A                                  19 23DFT(' 2 - MAIN MENU ')
A*
A************************
A*  ITEM INQUIRY SCREEN
A************************
A          R ITMMNU                      TEXT('ITEM INQUIRY SCREEN ONE')
A                                    2  2DFT('ENTER ITEM NUMBER')
A            ITEMNO         6N 0I  2 20
A                                  19  5DFT('CMD KEY 1 - END ')
A                                  19 23DFT(' 2 - MAIN MENU ')
A************************
A*  ITEM DISPLAY
A************************
A          R ITMSC2                      TEXT('ITEM INQUIRY SCREEN TWO') OVE+
A                                         RLAY
A                                    4  2DFT('DESC-')
A            DSC           30      4  8
A                                    5  2DFT('QUANTITY AVAILABLE')
A            QAVAIL         7N 0   5 25
A                                    6 11DFT('ON HAND')
A            QTYH           7N 0   6 25
A                                    7 11DFT('ON ORDER')
A            QTYO           7N 0   7 25
A                                    8 11DFT('BACK ORDER')
A            QTYB           7N 0   8 25
A                                    9  2DFT('UNIT OF MEASURE')
```

Figure 9-9 (Part 1 of 2). DDS for Source Program Multiple-Session Inquiry Using DSPFIL

```
A            UNT            2      9 30
A                                  10  2DFT('PRICE PER UNIT')
A            PR1            7Y 2  10 24EDTCDE(3)
A                                  11  8DFT('QUANTITY')
A            PR5            7Y 0  11 25EDTCDE(3)
A                                  12  8DFT('FREIGHT')
A            UFR            5Y 2  12 26EDTCDE(3)
A                                  13 32DFT('MORE... ')
A                                  19  5DFT('CMD KEY 1 - END ')
A                                  19 23DFT(' 2 - MAIN MENU ')
A                                  19 40DFT(' 3 - ITEM MENU ')
A*****************************
A*  ITEM ADDITIONAL DISPLAY
A*****************************
A          R ITMSC3                     TEXT('ITEM INQUIRY SCREEN 3  ') OVE+
A                                       RLAY
A                                   5  2DFT('SALES MONTH')
A            SLSM           9Y 2    5 16EDTCDE(1)
A                                   6  8DFT('Y-T-D')
A            SLSY          11Y 2    6 14EDTCDE(1)
A                                   7  2DFT('COSTS MONTH')
A            CSTM           9Y 2    7 16EDTCDE(1)
A                                   8  8DFT('Y-T-D')
A            CSTY          11Y 2    8 14EDTCDE(1)
A                                   9  2DFT('PROFIT PCT')
A            PROFIT         5Y 2    9 22EDTCDE(1)
A                                  10  2DFT('LOST SALES')
A            LOSTS         11Y 2   10 14EDTCDE(1)
A                                  19  5DFT('CMD KEY 1 - END ')
A                                  19 23DFT(' 2 - MAIN MENU ')
A************************
A*   TIMOUT SCREEN.
A************************
A          R TIMOUT                     TEXT('TIME OUT SCREEN')         OVE+
A                                       RLAY
A                                  20  2DFT('REMOTE SYSTEM TIMED OUT. ENTER+
A                                        1 TO TRY AGAIN OR 2 TO END.')
A            TIMRSP          1   I 20 61
```

Figure 9-9 (Part 2 of 2). DDS for Source Program Multiple-Session Inquiry Using DSPFIL

ICF File Creation and Program Device Entry Definition: The command needed to create the ICF file is:

```
CRTICFF  FILE(CICFLIB/CMNFIL)  SRCFILE(CICFLIB/QICFPUB)  SRCMBR(CMNFIL)
   ACQPGMDEV(*NONE)  MAXPGMDEV(4)  WAITRCD(30)
   TEXT("SOURCE ICF FILE FOR MULTIPLE SESSION PROGRAM")
```

The commands needed to define the four program device entries are:

```
OVRICFDEVE PGMDEV(ICF00) RMTLOCNAME(CHICAGO) FMTSLT(*RECID)

OVRICFDEVE PGMDEV(ICF01) RMTLOCNAME(NEWYORK) FMTSLT(*RECID)

OVRICFDEVE PGMDEV(ICF02) RMTLOCNAME(DETROIT) FMTSLT(*RECID)

OVRICFDEVE PGMDEV(ICF03) RMTLOCNAME(MADISON) FMTSLT(*RECID)
```

Program Explanation: The following explains the structure of the program example illustrated in Figure 9-10 on page 9-17. The ICF file used is defined by the user, and uses externally described data formats (DDS). The reference numbers in the explanation below correspond to the numbers in the following program example.

In the following example, the ICF file used is externally described, although the structure must be defined in the program manually by the programmer. The C/400 compiler does not automatically map the file structures into the program.

1 This section defines the ICF file (CMNFIL) structure used in the program.

CMNFIL is the ICF file used to send records to and receive records from each of the four target programs. CMNFIL is implemented with the file-level keyword, INDARA, indicating a separate indicator area is used.

2 This section defines the display file (DSPFIL) structure used in the program.

DSPFIL is the display file used to receive user's requests and to report the information received based on the request. DSPFIL is implemented with the file-level keyword, INDARA, indicating a separate indicator area is used.

3 This structure is used to print the major or minor return code to the print file QSYSPRT. Output to QSYSPRT is by record, so the filler field is used to blank out characters that may have been set by a previous write to the print file. See section **26**.

4 The variables to be used globally by the program are defined here. The common and display/ICF feedback area pointers, and the file pointers are defined.

5 The routines, except for the main routine, are prototyped so the compiler knows the type of value returned and the type of parameters passed, if any.

6 The ICF and display files are opened for record input/output, and the printer file is opened for output. The ICF and display files are opened with the separate indicator area option specified.

7 The separate indicator area is defined and initialized. The variable `dsp_indic` is of type `_SYSindara`, which is a character array of size 99.

8 The four program devices used by the program are explicitly acquired.

The device for the work station is implicitly acquired when the DSPFIL file is opened.

9 The program devices are specified before calling the evoke function four times to start transactions with the target program.

10 The main menu is displayed on the screen, and the user is asked to enter a 1 to process a request for item information, or a 2 to process a request for customer information.

If CMD 1 is pressed, a detach is sent to the remote program, the sessions are released, the files are closed, and the program ends. If CMD 1 is pressed, indicator 99 is set, which is position 98 in the separate indicator area array. The numbers are offset by one since arrays begin at subscript 0 (`dsp_indic??(0??)` is where indicator 1 would be set). The separate indicator array must be initialized to character zeros before each input operation where indicators are checked.

11 This function evokes the target program. If an error occurs, the program is ended.

When the program start request is received at the remote system, CICFLIB is searched for CTGTDMULCL and that program then starts. CTGTDMULCL is a CL program that contains the following statements:

```
ADDLIBLE CICFLIB
CALL CICFLIB/CTGTDMUL
```

12 This procedure displays the item number inquiry screen, and reads input from the user. If CMD 1 is pressed, the end-program flag is set, and the program ends upon return to the main section of the program (`main`). If CMD 2 is pressed, control goes back to the main menu and a new main menu is displayed.

If a valid number was entered, then a function to send the item number to the remote program is called, and if no errors were encountered, a read to the ICF file is issued to get the item information.

13 This function selects the proper program device to use to send the item number to the target program. ICF01 is used for numbers less than 400000, ICF02 for numbers greater than 400000 and less than 700000, and ICF03 for larger numbers. The number is copied from a display file structure to an ICF file structure, and then it is sent with an invite. The value returned from checking the return code (0 for 0000 return code, 1 otherwise) is returned to the caller.

14 This function gets called to receive item information from the target program, after the user has used a valid item number and that number has been sent to the target program.

The read is a read-from-invited-devices operation, since the item number was previously sent with an invite and a QXXREADINVDEV instruction is performed before the read.

A check is made to see if the remote system has timed out. (The wait time was specified on the CRTICFF command). A 0310 return code means a time-out has occurred after a read-from-invited-devices operation was issued by the source program. If a time-out did occur, a message is written to the screen asking the user to try again (enter 1) or to end the program (enter 2). This function gets called again if the user enters 1, but if 2 is entered, control returns to **10** and the end-program flag is set.

If no data was received, a 03XX return code is received, and the request is then sent again.

If the data returns in the wrong format, an error indication is returned to **10**.

The target program may not have found the record corresponding to the item number sent, in which case 000000 would be returned from the target as the item number. If 000000 was received, then control goes to **12**.

The record received from the remote system is copied into the record used to print the information on the screen, and then the information is written to the screen using format ITMSC2.

If CMD 1 is pressed from this screen, the end-program flag is set causing the program to end on return to the main section of the program (`main`). If CMD 2 is pressed, the main menu is written to the screen. If CMD 3 is pressed, the item inquiry menu is written to the screen. By pressing Enter, the profit and loss figures are calculated and written to the work station.

15 This procedure converts some of the values from character strings to integers, so that profit and loss figures may be calculated. The character strings returned by the remote system do not contain the end-of-string character '\0' necessary for the atoi routine to work (atoi is a system supplied routine to convert character strings to integer), and so '\0' is inserted before conversion.

The integers must then be converted back to character strings to conform with the character data types in 2. Note that the strings are converted to integers instead of float, since there is only an implied decimal point in the character strings (the decimal points are "inserted" when they are displayed using the display file DSPFIL).

16 This procedure converts integers to character strings. The integers are converted a digit at a time into characters.

17 This procedure processes customer information requests, and is much like 12. The customer inquiry screen is displayed, indicators are checked, and functions are called to handle the sending and receiving of data.

18 This function sends the customer number to the remote system with an invite. Program device ICF00 is used for processing customer information.

19 This function receives information about a customer from the remote system and is structured like 14. The major difference between the routine in section 19 and the one in 14 is that a check for a 000000 customer number received from the remote system is made before any other checks if the record format is valid. If the target program cannot find the record corresponding to the item or customer number, the record ID field is set to 'I' since the record ID does not exist.

20 This function calls a procedure (section 27) to access the display/ICF feedback area. It then checks for a 00 major return code, which indicates that the previous I/O operation was successful.

21 This function calls a procedure (section 27) to access the display/ICF feedback area. It then checks for a 03 major return code, which indicates that data was not received.

22 This function calls a procedure (section 27) to access the display/ICF feedback area. It then checks for a 0310 return code, which indicates that the timer interval has ended.

23 This procedure issues a detach function to each of the program devices that were acquired, indicating to the remote systems that this program is about to end.

24 This procedure releases the sessions that were acquired, and then calls another procedure to close the files.

25 This procedure closes the ICF, display, and printer files. If an error occurs while attempting to close a file, another close operation is issued.

26 This procedure retrieves the return code to be printed from the display/ICF feedback area, and then prints the appropriate message to the print file.

27 This procedure updates the common I/O feedback area and the display/ICF feedback area. The pointers must be reset after each I/O operation, after which information is to be retrieved from the feedback areas. Therefore, this procedure is called before any checking of return codes is done.

```
5728CX1 R00 M02 880101 IBM C/400          CICFLIB/QICFPUB/CSRCDMUL          06/15/89 10:41:42          Page   1
Command options:
  Program name . . . . . .:  CICFLIB/
  Source file name . . . . .:  CICFLIB/
  Compiler options . . . . .:  *NOAGGR  *NODEBUG   *NOEXPMAC    *GEN     *NOALWB IND  *NOLIS T     *NOPP ONLY   *NOS
                            :  *NOSHOWINC  *SOURCE   *NOXREF
  Generation options . . . .:  *NOLIST  *NOXREF   *GEN    *NOATR     *NODUMP      *NOSUB STR   *SUBS
                            :  *NOALWBND
  User profile . . . . . .:  *USER
  Language level . . . . .:  *EXTENDED
  Source margins . . . . .:  1 - 32767
  Sequence columns. . . . . :  none
  Authority . . . . . :  *CHANGE
  Text description . . . . .:  Source C program for APPC
  Printer file . . . . . .:  *LIBL/
  Lines per page . . . . .:  66
Actual Program Source:
  Member . . . . . .:  CSRCDMUL
  File . . . . . .:  QICFPUB
  Library . . . . . :  CICFLIB
  Last change . . . . . :  89/06/15  09:50:2
  Description . . . . . :  Source C program for APPC
5728CX1 R00 M02 880101 IBM C/400          CICFLIB/QICFPUB/CSRCDMUL          06/15/89 10:41:42          Page   2
LINE  STMT                              * * * * *  S O U R C E * * * * *                                SEQNO INCLUDE
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9----+----0
    1       |/*--------------------------------------------------------------------------*/                |     1
    2       |/*  This program assigns four sessions as follows:                          */                |     2
    3       |/*    'ICF00' to inquire about a customer account before an order is      */                |     3
    4       |/*            processed.                                                  */                |     4
    5       |/*    'ICF01' to inquire about the inventory status of an item being      */                |     5
    6       |/*            ordered (item 000001 thru 399999).                          */                |     6
    7       |/*    'ICF02' to inquire about the inventory status of an item being      */                |     7
    8       |/*            ordered (item 400000 thru 699999).                          */                |     8
    9       |/*    'ICF03' to inquire about the inventory status of an item being      */                |     9
   10       |/*            ordered (item 700000 thru 999999).                          */                |    10
   11       |/*  A display device is used to enter the request (using a customer       */                |    11
   12       |/*  and an item menu) that is sent to the remote system.                  */                |    12
   13       |/*--------------------------------------------------------------------------*/                |    13
   14       |                                                                                            |    14
   15       |#define ON 1                                                                               |    15
   16       |#define OFF 0                                                                              |    16
   17       |#define ION '1'                          /* Indicator is set on */                         |    17
   18       |#define IOFF '0'                                                                           |    18
   19       |#define ERROR 1                          /* Error occurred */                              |    19
   20       |#define NOERROR 0                                                                          |    20
   21       |#define NORM_END 1                       /* Print normal end message */                    |    21
   22       |#define RCD_ERR 2                        /* Print wrong record error msg */                |    22
   23       |#define ERR_END 3                        /* Print generic error message */                 |    23
   24       |#include <stdio.h>                       /* Standard I/O header */                         |    24
   25       |#include <stdlib.h>                      /* General utilities */                           |    25
   26       |#include <stddef.h>                      /* Standard definitions */                        |    26
   27       |#include <string.h>                      /* String handling utilities */                   |    27
   28       |#include <xxfdbk.h>                      /* Feedback area structures */                    |    28
   29       |#include <xxasio.h>                      /* Indicator area structure */                    |    29
   30       | 1                                                                                         |    30
   31       |/*--------------------------------------------------------------------------*/                |    31
   32       |/* Define the ICF file's structure.                                        */                |    32
   33       |/*--------------------------------------------------------------------------*/                |    33
   34       |                                                                                            |    34
   35       |struct {                                                                                    |    35
   36       |    char recitm;                                                                            |    36
   37       |    char itemno??(6??);                                                                     |    37
   38       |    char desc??(30??);                                                                      |    38
   39       |    char qtylst??(7??);                                                                     |    39
```

Figure 9-10 (Part 1 of 13). Source Program Example—CSRCDMUL (User-Defined Formats)

```
   40       |    char qtyoh??(7??);                                                                  |     40
   41       |    char qtyoo??(7??);                                                                  |     41
   42       |    char qtybo??(7??);                                                                  |     42
   43       |    char unitq??(2??);                                                                  |     43
   44       |    char pr01??(7??);                                                                   |     44
   45       |    char pr05??(7??);                                                                   |     45
   46       |    char ufrt??(5??);                                                                   |     46
   47       |    char slstm??(9??);                                                                  |     47
   48       |    char slsty??(11??);                                                                 |     48
   49       |    char csttm??(9??);                                                                  |     49
   50       |    char cstty??(11??);                                                                 |     50
   51       |    char pro??(5??);                                                                    |     51
   52       |    char los??(9??);                                                                    |     52
   53       |    char fill1??(56??);                                                                 |     53
   54       |} itmrsp_icf_i;                                                                         |     54
   55       |                                                                                        |     55
   56       |struct {                                                                                |     56
   57       |    char reccus;                                                                        |     57
   58       |    char custno??(6??);                                                                 |     58
   59       |    char dname??(30??);                                                                 |     59
   60       |    char dlstor??(6??);                                                                 |     60
   61       |    char dslstm??(9??);                                                                 |     61
 5728CX1 R00 M02 880101 IBM C/400          CICFLIB/QICFPUB/CSRCDMUL              06/15/89 10:41:42          Page   3
LINE  STMT                              * * * * *   S O U R C E * * * * *                          SEQNO INCLUDE
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9----+----0
   62       |    char dspm01??(9??);                                                                 |     62
   63       |    char dspm02??(9??);                                                                 |     63
   64       |    char dspm03??(9??);                                                                 |     64
   65       |    char dsttyd??(11??);                                                                |     65
   66       |    char idept??(3??);                                                                  |     66
   67       |    char fill2??(57??);                                                                 |     67
   68       |} dtlrsp_icf_i;                                                                         |     68
   69       |                                                                                        |     69
   70       |struct {                                                                                |     70
   71       |    char pgmid??(10??);                                                                 |     71
   72       |    char lib??(10??);                                                                   |     72
   73       |} evkreq_icf_o;                                                                         |     73
   74       |                                                                                        |     74
   75       |struct {                                                                                |     75
   76       |    char itemno??(6??);                                                                 |     76
   77       |} itmreq_icf_o;                                                                         |     77
   78       |                                                                                        |     78
   79       |struct {                                                                                |     79
   80       |    char custno??(6??);                                                                 |     80
   81       |} dtlreq_icf_o;                                                                         |     81
   82       | 2                                                                                      |     82
   83       |/*--------------------------------------------------------------------*/               |     83
   84       |/* Define the display file's structure.                              */               |     84
   85       |/*--------------------------------------------------------------------*/               |     85
   86       |                                                                                        |     86
   87       |struct {                                                                                |     87
   88       |    char option;                                                                        |     88
   89       |} cimenu_dsp_i;                                                                         |     89
   90       |                                                                                        |     90
   91       |struct {                                                                                |     91
   92       |    char custno??(6??);                                                                 |     92
   93       |} dtlmnu_dsp_i;                                                                         |     93
   94       |                                                                                        |     94
   95       |struct {                                                                                |     95
   96       |    char custn??(6??);                                                                  |     96
   97       |    char dept??(3??);                                                                   |     97
   98       |    char dlstr??(6??);                                                                  |     98
   99       |    char dslsm??(9??);                                                                  |     99
  100       |    char dspm1??(9??);                                                                  |    100
  101       |    char dspm2??(9??);                                                                  |    101
  102       |    char dspm3??(9??);                                                                  |    102
  103       |    char dstyd??(11??);                                                                 |    103
  104       |    char cname??(30??);                                                                 |    104
```

Figure 9-10 (Part 2 of 13). Source Program Example—CSRCDMUL (User-Defined Formats)

```
  105       |} dtlscr_dsp_o;                                                                                    |  105
  106       |                                                                                                   |  106
  107       |struct {                                                                                           |  107
  108       |    char itemno??(6??);                                                                            |  108
  109       |} itmmnu_dsp_i;                                                                                    |  109
  110       |                                                                                                   |  110
  111       |struct {                                                                                           |  111
  112       |    char dsc??(30??);                                                                              |  112
  113       |    char qavail??(7??);                                                                            |  113
  114       |    char qtyh??(7??);                                                                              |  114
  115       |    char qtyo??(7??);                                                                              |  115
  116       |    char qtyb??(7??);                                                                              |  116
  117       |    char unt??(2??);                                                                               |  117
  118       |    char pr1??(7??);                                                                               |  118
  119       |    char pr5??(7??);                                                                               |  119
  120       |    char ufr??(5??);                                                                               |  120
  121       |} itmsc2_dsp_o;                                                                                    |  121
  122       |                                                                                                   |  122
 5728CX1 R00 M02 880101 IBM C/400          CICFLIB/QICFPUB/CSRCDMUL                  06/15/89 10:41:42          Page   4
LINE  STMT                              * * * * *   S O U R C E * * * * *                                  SEQNO INCLUDE
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9----+----0
  123       |struct {                                                                                           |  123
  124       |    char slsm??(9??);                                                                              |  124
  125       |    char slsy??(11??);                                                                             |  125
  126       |    char cstm??(9??);                                                                              |  126
  127       |    char csty??(11??);                                                                             |  127
  128       |    char profit??(5??);                                                                            |  128
  129       |    char losts??(11??);                                                                            |  129
  130       |} itmsc3_dsp_o;                                                                                    |  130
  131       |                                                                                                   |  131
  132       |struct {                                                                                           |  132
  133       |    char timrsp;                                                                                   |  133
  134       |} timout_dsp_i_o;                                                                                  |  134
  135       | 3                                                                                                 |  135
  136       |/*-----------------------------------------------------------------------*/                         |  136
  137       |/* Define structure used to write to the print file.                     */                         |  137
  138       |/*-----------------------------------------------------------------------*/                         |  138
  139       |                                                                                                   |  139
  140       |struct {                                                                                           |  140
  141       |    char major??(2??);                                                                             |  141
  142       |    char minor??(2??);                                                                             |  142
  143       |    char filler??(32??);                  /* Used for padding with blanks */                       |  143
  144       |} print_rec;                                                                                       |  144
  145       | 4                                                                                                 |  145
  146       |/*-----------------------------------------------------------------------*/                         |  146
  147       |/* Declare global variables.                                             */                         |  147
  148       |/*-----------------------------------------------------------------------*/                         |  148
  149       |                                                                                                   |  149
  150       |FILE  *icffptr;                           /* Ptr to ICF file */                                    |  150
  151       |FILE  *dspfptr;                           /* Ptr to display file */                                |  151
  152       |FILE  *prtfptr;                           /* Ptr to print file */                                  |  152
  153       |XXIOFB_T  *comm_fdbk;                     /* Ptr to common I/O feedback */                         |  153
  154       |XXIOFB_DSP_ICF_T  *dsp_icf_fdbk;          /* Ptr to dsp/ICF I/O feedback */                        |  154
  155       | 5                                                                                                 |  155
  156       |int evoke_target(void);                                                                            |  156
  157       |void process_item_req(int *, _SYSindara);                                                          |  157
  158       |int send_item_req(void);                                                                           |  158
  159       |int rec_item_info(int *, _SYSindara);                                                              |  159
  160       |void calc_profit_loss(void);                                                                       |  160
  161       |void convert_to_char(int, char *, int);                                                            |  161
  162       |void process_cust_req(int *, _SYSindara);                                                          |  162
  163       |int send_cust_req(void);                                                                           |  163
  164       |int rec_cust_info(int *, _SYSindara);                                                              |  164
  165       |int pos_ret_code(void);                                                                            |  165
```

Figure 9-10 (Part 3 of 13). Source Program Example—CSRCDMUL (User-Defined Formats)

```
  166       |int pos_ret_code(void);                                                                    |    166
  167       |int check_no_data(void);                                                                   |    167
  168       |int check_timeout(void);                                                                   |    168
  169       |void detach(void);                                                                         |    169
  170       |void end_job(void);                                                                        |    170
  171       |void close_files(void);                                                                    |    171
  172       |void print_msg(int);                                                                       |    172
  173       |void get_access_to_fb(void);                                                               |    173
  174       |                                                                                           |    174
  175       |main()                                                                                     |    175
  176       |{                                                                                          |    176
  177       |    int  end_pgm_flag = OFF;              /* Signals program end request */                |    177
  178       |    _SYSindara dsp_indic;                 /* Display separate indic area */                |    178
  179       | 6                                                                                         |    179
  180       |    /*-------------------------------------------------------*/                           |    180
  181       |    /* Open ICF, display, and printer files.  If an error occurs,*/                       |    181
  182       |    /* the program will end.                                 */                          |    182
  183       |    /*-------------------------------------------------------*/                           |    183
 5728CX1 R00 M02 880101 IBM C/400            CICFLIB/QICFPUB/CSRCDMUL                 06/15/89 10:41:42          Page   5
LINE  STMT                              * * * * *   S O U R C E * * * * *                          SEQNO INCLUDE
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9----+----0
  184       |                                                                                           |    184
  185     1 |    if ((icffptr = fopen("CICFLIB/CMNFIL", "ab+ type=record indicators=y"))                |    185
  186       |                                                                  == NULL)                 |    186
  187     2 |         exit(ERROR);                                                                      |    187
  188     3 |    if ((dspfptr = fopen("CICFLIB/DSPFIL", "ab+ type=record indicators=y"))                |    188
  189       |                                                                  == NULL) {               |    189
  190     4 |         fclose(icffptr);                                                                  |    190
  191     5 |         exit(ERROR);                                                                      |    191
  192       |    }                                                                                      |    192
  193     6 |    if ((prtfptr = fopen("CICFLIB/QSYSPRT", "wb type=record")) == NULL) {                  |    193
  194     7 |         fclose(icffptr);                                                                  |    194
  195     8 |         fclose(dspfptr);                                                                  |    195
  196     9 |         exit(ERROR);                                                                      |    196
  197       |    }                                                                                      |    197
  198       | 7                                                                                         |    198
  199       |    /*-------------------------------------------------------*/                           |    199
  200       |    /* Set up separate indicator area for the display file.  */                           |    200
  201       |    /*-------------------------------------------------------*/                           |    201
  202       |                                                                                           |    202
  203       |     memset(dsp_indic, IOFF, 99);                                                          |    203
  204    11 |     QXXSINDARA(dspfptr, dsp_indic);                                                       |    204
  205       | 8                                                                                         |    205
  206       |    /*-------------------------------------------------------*/                           |    206
  207       |    /* Explicitly acquire four sessions. If an error occurs on   */                       |    207
  208       |    /* any of the acquire operations then the ICF and display    */                       |    208
  209       |    /* files will be closed, an error message will be printed,   */                       |    209
  210       |    /* and the program will end.                                 */                       |    210
  211       |    /*-------------------------------------------------------*/                           |    211
  212       |                                                                                           |    212
  213    12 |    QXXACQUIRE(icffptr, "ICF00     ");                                                     |    213
  214    13 |    if (pos_ret_code() == ERROR) {                                                         |    214
  215    14 |        close_files();                                                                     |    215
  216    15 |        exit(ERROR);                                                                       |    216
  217       |    }                                                                                      |    217
  218    16 |    QXXACQUIRE(icffptr, "ICF01     ");                                                     |    218
  219    17 |    if (pos_ret_code() == ERROR) {                                                         |    219
  220    18 |        close_files();                                                                     |    220
  221    19 |        exit(ERROR);                                                                       |    221
  222       |    }                                                                                      |    222
  223    20 |    QXXACQUIRE(icffptr, "ICF02     ");                                                     |    223
  224    21 |    if (pos_ret_code() == ERROR) {                                                         |    224
  225    22 |        close_files();                                                                     |    225
  226    23 |        exit(ERROR);                                                                       |    226
  227       |    }                                                                                      |    227
  228    24 |    QXXACQUIRE(icffptr, "ICF03     ");                                                     |    228
  229    25 |    if (pos_ret_code() == ERROR) {                                                         |    229
  230    26 |        close_files();                                                                     |    230
```

Figure 9-10 (Part 4 of 13). Source Program Example—CSRCDMUL (User-Defined Formats)

```
  231    27 |        exit(ERROR);                                                                                    |  231
  232       |    }                                                                                                   |  232
  233       | 9                                                                                                      |  233
  234       |    /*-------------------------------------------------------*/                                        |  234
  235       |    /* Evoke the target four times. If an error occurs on any of */                                     |  235
  236       |    /* the evoke operations then detach, release, and close      */                                     |  236
  237       |    /* operations will be issued, an error message printed, and  */                                     |  237
  238       |    /* the program will end.                                     */                                     |  238
  239       |    /*-------------------------------------------------------*/                                        |  239
  240       |                                                                                                        |  240
  241    28 |    QXXPGMDEV(icffptr, "ICF00     ");                                                                   |  241
  242    29 |    if (evoke_target() == ERROR)                                                                        |  242
  243    30 |        exit(ERROR);                                                                                    |  243
  244    31 |    QXXPGMDEV(icffptr, "ICF01     ");                                                                   |  244
 5728CX1 R00 M02 880101 IBM C/400            CICFLIB/QICFPUB/CSRCDMUL                    06/15/89 10:41:42          Page   6
LINE  STMT                              * * * * *   S O U R C E * * * * *                                  SEQNO INCLUDE
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9----+----0
  245    32 |    if (evoke_target() == ERROR)                                                                        |  245
  246    33 |        exit(ERROR);                                                                                    |  246
  247    34 |    QXXPGMDEV(icffptr, "ICF02     ");                                                                   |  247
  248    35 |    if (evoke_target() == ERROR)                                                                        |  248
  249    36 |        exit(ERROR);                                                                                    |  249
  250    37 |    QXXPGMDEV(icffptr, "ICF03     ");                                                                   |  250
  251    38 |    if (evoke_target() == ERROR)                                                                        |  251
  252    39 |        exit(ERROR);                                                                                    |  252
  253       | 10                                                                                                     |  253
  254       |    /*-------------------------------------------------------*/                                        |  254
  255       |    /* Put out the main menu to the display, and depending on    */                                     |  255
  256       |    /* the input, either request item information or customer    */                                     |  256
  257       |    /* information from remote system.  If CMD 1 (indicator 99)  */                                     |  257
  258       |    /* is pressed on any screen, the program ends. If the user   */                                     |  258
  259       |    /* picks option 1, an item inquiry is processed, and if the  */                                     |  259
  260       |    /* user picks option 2 a customer inquiry is processed.  If  */                                     |  260
  261       |    /* CMD 1 wasn't pressed, nor option 1 or 2, then the input   */                                     |  261
  262       |    /* is invalid.                                               */                                     |  262
  263       |    /*-------------------------------------------------------*/                                        |  263
  264       |                                                                                                        |  264
  265    40 |    while (end_pgm_flag == OFF) {                                                                       |  265
  266    41 |        QXXFORMAT(dspfptr, "CIMENU    ");                                                               |  266
  267    42 |        fwrite(NULL, 0, 0, dspfptr);                                                                   |  267
  268       |        memset(dsp_indic, IOFF, 99);                                                                    |  268
  269    44 |        fread(&cimenu_dsp_i, sizeof(cimenu_dsp_i), 1, dspfptr);                                         |  269
  270    45 |        if (dsp_indic??(98??) == IOFF) {                                                                |  270
  271    46 |            if (cimenu_dsp_i.option == '1')                                                             |  271
  272    47 |                process_item_req(&end_pgm_flag, dsp_indic);                                             |  272
  273       |            else                                                                                        |  273
  274    48 |                if (cimenu_dsp_i.option == '2')                                                         |  274
  275    49 |                    process_cust_req(&end_pgm_flag, dsp_indic);                                         |  275
  276       |        }                                                                                               |  276
  277       |        else {                                                                                          |  277
  278    50 |            print_msg(NORM_END);                                                                        |  278
  279    51 |            end_pgm_flag = ON;                                                                          |  279
  280       |        }                                                                                               |  280
  281       |    }                                                                                                   |  281
  282    52 |    detach();                                                                                           |  282
  283    53 |    end_job();                                                                                          |  283
  284       |}                                                                                                       |  284
  285       |                                                                                                        |  285
  286       |                                                                                                        |  286
  287       |/*-------------------------------------------------------------------------*/                          |  287
  288       |/*                        Evoke the Target Program                         */                          |  288
  289       |/* Evoke the target program. If an error occurs then a detach will be   */                          |  289
  290       |/* sent, a release will be issued, and the ICF, display, and printer    */                          |  290
  291       |/* files will be closed.                                                */                          |  291
  292       |/*-------------------------------------------------------------------------*/                          |  292
  293       | 11                                                                                                     |  293
  294       |evoke_target()                                                                                          |  294
  295       |{                                                                                                       |  295
```

Figure 9-10 (Part 5 of 13). Source Program Example—CSRCDMUL (User-Defined Formats)

```
  296     1 |    QXXFORMAT(icffptr, "EVKREQ    ");                                                                  |    296
  297     2 |    fwrite(NULL, 0, 0, icffptr);                                                                        |    297
  298     3 |    if (pos_ret_code() == ERROR) {                                                                      |    298
  299     4 |        print_msg(ERR_END);                                                                             |    299
  300     5 |        detach();                                                                                       |    300
  301     6 |        end_job();                                                                                      |    301
  302     7 |        return(ERROR);                                                                                  |    302
  303       |    }                                                                                                   |    303
  304     8 |    return(NOERROR);                                                                                    |    304
  305       |}                                                                                                       |    305
 5728CX1 R00 M02 880101 IBM C/400            CICFLIB/QICFPUB/CSRCDMUL                  06/15/89 10:41:42              Page   7
LINE  STMT                              * * * * *   S O U R C E * * * * *                                    SEQNO INCLUDE
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9----+----0
  306       |                                                                                                      |    306
  307       |                                                                                                      |    307
  308       |/*--------------------------------------------------------------------------*/                        |    308
  309       |/*                      Process Item Request Screen                         */                        |    309
  310       |/* This routine puts out the item request screen and reads the input    */                           |    310
  311       |/* from the user.  CMD 1 (end job) and CMD 2 (go to main menu)          */                           |    311
  312       |/* are checked.  Routines to send a valid item number and to receive    */                           |    312
  313       |/* item information are called.                                         */                           |    313
  314       |/*--------------------------------------------------------------------------*/                        |    314
  315       | 12                                                                                                   |    315
  316       |void process_item_req(int *end_pgm_flag, _SYSindara dsp_indic)                                        |    316
  317       |{                                                                                                     |    317
  318     1 |    QXXFORMAT(dspfptr, "ITMMNU    ");                                                                 |    318
  319     2 |    fwrite(NULL, 0, 0, dspfptr);                                                                      |    319
  320       |    memset(dsp_indic, IOFF, 99);                                                                      |    320
  321     4 |    fread(&itmmnu_dsp_i, sizeof(itmmnu_dsp_i), 1, dspfptr);                                           |    321
  322     5 |    if (dsp_indic??(98??) == ION) {          /* CMD 1, indic 99 */                                    |    322
  323     6 |        print_msg(NORM_END);                                                                          |    323
  324     7 |        *end_pgm_flag = ON;                                                                           |    324
  325       |    }                                                                                                 |    325
  326       |    else                                                                                              |    326
  327     8 |        if (dsp_indic??(97??) == IOFF)       /* CMD 2, indic 98 */                                    |    327
  328     9 |            if (send_item_req() == NOERROR) {                                                         |    328
  329    10 |                if (rec_item_info(end_pgm_flag, dsp_indic) == ERROR) {                                |    329
  330    11 |                    print_msg(ERR_END);                                                               |    330
  331    12 |                    *end_pgm_flag = ON;                                                               |    331
  332       |                }                                                                                     |    332
  333       |            }                                                                                         |    333
  334       |            else {                                                                                    |    334
  335    13 |                print_msg(ERR_END);                                                                   |    335
  336    14 |                *end_pgm_flag = ON;                                                                   |    336
  337       |            }                                                                                         |    337
  338       |}                                                                                                     |    338
  339       |                                                                                                      |    339
  340       |                                                                                                      |    340
  341       |/*--------------------------------------------------------------------------*/                        |    341
  342       |/*                         Send Item Request                                */                        |    342
  343       |/* This routine sends the item number entered to the appropriate target */                           |    343
  344       |/* program based on the range of the number.                            */                           |    344
  345       |/*--------------------------------------------------------------------------*/                        |    345
  346       | 13                                                                                                   |    346
  347       |send_item_req()                                                                                       |    347
  348       |{                                                                                                     |    348
  349     1 |    QXXFORMAT(icffptr, "ITMREQ    ");                                                                 |    349
  350     2 |    if (strncmp(itmmnu_dsp_i.itemno, "399999", 6) != 1)                                               |    350
  351     3 |        QXXPGMDEV(icffptr, "ICF01     ");                                                             |    351
  352       |    else                                                                                              |    352
  353     4 |        if (strncmp(itmmnu_dsp_i.itemno, "699999", 6) != 1)                                           |    353
  354     5 |            QXXPGMDEV(icffptr, "ICF02     ");                                                        |    354
  355       |        else                                                                                          |    355
  356     6 |            QXXPGMDEV(icffptr, "ICF03     ");                                                        |    356
  357       |    strncpy(itmreq_icf_o.itemno, itmmnu_dsp_i.itemno, 6);                                             |    357
  358     8 |    fwrite(&itmreq_icf_o, sizeof(itmreq_icf_o), 1, icffptr);                                          |    358
  359     9 |    return(pos_ret_code());                                                                           |    359
  360       |}                                                                                                     |    360
```

Figure 9-10 (Part 6 of 13). Source Program Example—CSRCDMUL (User-Defined Formats)

```
  361       |                                                                                                                  |   361
  362       |                                                                                                                  |   362
  363       |/*------------------------------------------------------------------------*/                                      |   363
  364       |/*                     Receive Item Information                           */                                      |   364
  365       |/* The item information is read from the remote system.  A check is     */                                      |   365
  366       |/* made for three conditions following the read:  1) The remote system  */                                      |   366
 5728CX1 R00 M02 880101 IBM C/400            CICFLIB/QICFPUB/CSRCDMUL                    06/15/89 10:41:42              Page   8
LINE  STMT                              * * * * *   S O U R C E * * * * *                                              SEQNO INCLUDE
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9----+----0
  367       |/* timed out, 2) no data was received, and 3) data was returned in an   */                                      |   367
  368       |/* unexpected format.                                                    */                                      |   368
  369       |/* If the remote system times out (maj/min 0310), a message is written  */                                      |   369
  370       |/* to the screen asking to try again (enter 1) or to end the program    */                                      |   370
  371       |/* (enter 2).                                                            */                                      |   371
  372       |/* If no data is received (major 03) the request is sent again to the   */                                      |   372
  373       |/* remote system.                                                        */                                      |   373
  374       |/* If the record returns with the wrong record format, the program will */                                      |   374
  375       |/* end on error.                                                         */                                      |   375
  376       |/* If CMD 1 (in99) is pressed, the end program flag is set causing the  */                                      |   376
  377       |/* program to end on return to main.  If CMD 2 (in98) is pressed,       */                                      |   377
  378       |/* the main menu is written to the screen.  If CMD 3 (in97) is pressed, */                                      |   378
  379       |/* the item inquiry menu is written to the screen.                       */                                      |   379
  380       |/* If the remote program didn't find the item, and the item number      */                                      |   380
  381       |/* "000000" was returned, then the item request screen is displayed.    */                                      |   381
  382       |/*------------------------------------------------------------------------*/                                      |   382
  383       | 14                                                                                                             |   383
  384       |rec_item_info(int *end_pgm_flag, _SYSindara dsp_indic)                                                          |   384
  385       |{                                                                                                               |   385
  386     1 |    QXXREADINVDEV(icffptr);                                                                                     |   386
  387     2 |    fread(&itmrsp_icf_i, sizeof(itmrsp_icf_i), 1, icffptr);                                                     |   387
  388     3 |    if (check_timeout() == 0) {                                                                                 |   388
  389     4 |        QXXFORMAT(dspfptr, "TIMOUT    ");                                                                       |   389
  390     5 |        fwrite(NULL, 0, 0, dspfptr);                                                                            |   390
  391     6 |        fread(&timout_dsp_i_o, sizeof(timout_dsp_i_o), 1, dspfptr);                                             |   391
  392     7 |        if (timout_dsp_i_o.timrsp == '1')                                                                       |   392
  393     8 |            return(rec_item_info(end_pgm_flag, dsp_indic));                                                     |   393
  394       |        else                                                                                                    |   394
  395     9 |            if (timout_dsp_i_o.timrsp == '2') {                                                                 |   395
  396    10 |                print_msg(NORM_END);                                                                            |   396
  397    11 |                *end_pgm_flag = ON;                                                                             |   397
  398    12 |                return(NOERROR);                                                                                |   398
  399       |            }                                                                                                   |   399
  400       |    }                                                                                                           |   400
  401       |    else                                                                                                        |   401
  402    13 |        if (check_no_data() == 0) {         /* No data received */                                              |   402
  403    14 |            QXXFORMAT(icffptr, "ITMREQ    ");                                                                   |   403
  404    15 |            fwrite(&itmreq_icf_o, sizeof(itmreq_icf_o), 1, icffptr);                                            |   404
  405    16 |            return(pos_ret_code());                                                                             |   405
  406       |        }                                                                                                       |   406
  407       |        else {                               /* Check for valid record fmt */                                   |   407
  408    17 |            comm_fdbk = QXXIOFBK(icffptr);                                                                      |   408
  409    18 |            if (strncmp(comm_fdbk->rec_format, "ITMRSP    ", 10) != 0) {                                        |   409
  410    19 |                print_msg(RCD_ERR);                                                                             |   410
  411    20 |                return(ERROR);                                                                                  |   411
  412       |            }                                                                                                   |   412
  413       |        }                                                                                                       |   413
  414    21 |    if (strncmp(itmrsp_icf_i.itemno, "000000", 6) != 1)  /* Item not found */                                   |   414
  415    22 |        process_item_req(end_pgm_flag, dsp_indic);                                                              |   415
  416       |    else {                                                                                                      |   416
  417       |        strncpy(itmsc2_dsp_o.dsc, itmrsp_icf_i.desc, 30);                                                       |   417
  418       |        strncpy(itmsc2_dsp_o.qavail, itmrsp_icf_i.qtylst, 7);                                                   |   418
  419       |        strncpy(itmsc2_dsp_o.qtyo, itmrsp_icf_i.qtyoo, 7);                                                      |   419
  420       |        strncpy(itmsc2_dsp_o.qtyh, itmrsp_icf_i.qtyoh, 7);                                                      |   420
  421       |        strncpy(itmsc2_dsp_o.qtyb, itmrsp_icf_i.qtybo, 7);                                                      |   421
  422       |        strncpy(itmsc2_dsp_o.unt, itmrsp_icf_i.unitq, 2);                                                       |   422
  423       |        strncpy(itmsc2_dsp_o.pr1, itmrsp_icf_i.pr01, 7);                                                        |   423
  424       |        strncpy(itmsc2_dsp_o.pr5, itmrsp_icf_i.pr05, 7);                                                        |   424
  425       |        strncpy(itmsc2_dsp_o.ufr, itmrsp_icf_i.ufrt, 5);                                                        |   425
```

Figure 9-10 (Part 7 of 13). Source Program Example—CSRCDMUL (User-Defined Formats)

```
  426    32 |        QXXFORMAT(dspfptr, "ITMSC2    ");   /* Display item screen 2 */                                  |    426
  427    33 |        fwrite(&itmsc2_dsp_o, sizeof(itmsc2_dsp_o), 1, dspfptr);                                           |    427
 5728CX1 R00 M02 880101 IBM C/400            CICFLIB/QICFPUB/CSRCDMUL                  06/15/89 10:41:42              Page   9
LINE  STMT                          * * * * *   S O U R C E * * * * *                                          SEQNO INCLUDE
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9----+----0
  428       |        memset(dsp_indic, IOFF, 99);                                                                    |    428
  429    35 |        fread(NULL, 0, 0, dspfptr);                                                                     |    429
  430    36 |        if (dsp_indic??(98??) == ION) {     /* CMD 1, indic 99 */                                       |    430
  431    37 |            print_msg(NORM_END);                                                                        |    431
  432    38 |            *end_pgm_flag = ON;                                                                         |    432
  433       |        }                                                                                               |    433
  434       |        else                                                                                            |    434
  435    39 |            if (dsp_indic??(96??) == ION)   /* CMD 3, indic 97 */                                       |    435
  436    40 |                process_item_req(end_pgm_flag, dsp_indic);                                              |    436
  437       |            else                                                                                        |    437
  438    41 |                if (dsp_indic??(97??) == IOFF) {                                                        |    438
  439    42 |                    calc_profit_loss();                                                                 |    439
  440    43 |                    QXXFORMAT(dspfptr, "ITMSC3    ");  /* Display screen 3 */                       |    440
  441    44 |                    fwrite(&itmsc3_dsp_o, sizeof(itmsc3_dsp_o), 1, dspfptr);                            |    441
  442       |                    memset(dsp_indic, IOFF, 99);                                                        |    442
  443    46 |                    fread(NULL, 0, 0, dspfptr);                                                         |    443
  444       |                }                                                                                   |    444
  445       |    }                                                                                                   |    445
  446    47 |    return(NOERROR);                                                                                    |    446
  447       |}                                                                                                       |    447
  448       |                                                                                                        |    448
  449       |                                                                                                        |    449
  450       |/*--------------------------------------------------------------------*/                                |    450
  451       |/*                    Calculate Profit/Loss                           */                                |    451
  452       |/* This routine calculates profit and loss figures and displays them  */                                |    452
  453       |/* on screen two of the item.  Some fields are converted to integer for */                              |    453
  454       |/* calculation and then back to character strings to be displayed.    */                               |    454
  455       |/* The end-of-string character, '\0', must be concatenated to the end  */                              |    455
  456       |/* of the strings to be converted to integer for atoi to work properly. */                             |    456
  457       |/*--------------------------------------------------------------------*/                                |    457
  458       | 15                                                                                                     |    458
  459       |void calc_profit_loss()                                                                                 |    459
  460       |{                                                                                                       |    460
  461       |    int slstm_i, losts_i, profit_temp;                                                                  |    461
  462       |    char slstm_c??(10??), csttm_c??(10??), qtylst_c??(8??);                                             |    462
  463       |    char pr01_c??(8??), losts_char??(11??), profit_temp_char??(5??);                                    |    463
  464       |                                                                                                        |    464
  465       |    strncpy(slstm_c, itmrsp_icf_i.slstm, 9);                                                            |    465
  466     2 |    slstm_c??(9??) = '\0';                                                                              |    466
  467       |    strncpy(csttm_c, itmrsp_icf_i.csttm, 9);                                                            |    467
  468     4 |    csttm_c??(9??) = '\0';                                                                              |    468
  469       |    strncpy(qtylst_c, itmrsp_icf_i.slstm, 7);                                                           |    469
  470     6 |    qtylst_c??(7??) = '\0';                                                                             |    470
  471       |    strncpy(pr01_c, itmrsp_icf_i.slstm, 7);                                                             |    471
  472     8 |    pr01_c??(7??) = '\0';                                                                               |    472
  473     9 |    slstm_i = atoi(slstm_c);                /* Convert string to integer */                             |    473
  474    10 |    profit_temp = slstm_i + atoi(itmrsp_icf_i.csttm);                                                   |    474
  475    11 |    profit_temp *= 100;                                                                                 |    475
  476    12 |    if (slstm_i > 0)                                                                                    |    476
  477    13 |        profit_temp /= slstm_i;                                                                         |    477
  478    14 |    losts_i = atoi(qtylst_c) * atoi(pr01_c);                                                            |    478
  479       |    strncpy(itmsc3_dsp_o.slsm, itmrsp_icf_i.slstm, 9);                                                  |    479
  480       |    strncpy(itmsc3_dsp_o.slsy, itmrsp_icf_i.slsty, 11);                                                 |    480
  481       |    strncpy(itmsc3_dsp_o.cstm, itmrsp_icf_i.csttm, 9);                                                  |    481
  482       |    strncpy(itmsc3_dsp_o.csty, itmrsp_icf_i.cstty, 11);                                                 |    482
  483    19 |    convert_to_char(losts_i, losts_char, 11);   /* Convert integer to string */                         |    483
  484       |    strncpy(itmsc3_dsp_o.losts, losts_char, 11);                                                        |    484
  485    21 |    convert_to_char(profit_temp, profit_temp_char, 5);                                                  |    485
  486       |    strncpy(itmsc3_dsp_o.profit, profit_temp_char, 5);                                                  |    486
  487       |}                                                                                                       |    487
  488       |                                                                                                        |    488
```

Figure 9-10 (Part 8 of 13). Source Program Example—CSRCDMUL (User-Defined Formats)

```
 5728CX1 R00 M02 880101 IBM C/400          CICFLIB/QICFPUB/CSRCDMUL          06/15/89 10:41:42          Page   10
LINE  STMT                              * * * * *  S O U R C E * * * * *                              SEQNO INCLUDE
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9----+----0
  489       |                                                                                          |   489
  490       |/*--------------------------------------------------------------------*/                  |   490
  491       |/*               Convert Integer to Character String                  */                  |   491
  492       |/* This routine converts the integer passed in to a character string a  */                  |   492
  493       |/* character at a time.                                                 */                  |   493
  494       |/*--------------------------------------------------------------------*/                  |   494
  495       | 16                                                                                       |   495
  496       |void convert_to_char(int num, char s??(??), int len)                                      |   496
  497       |{                                                                                         |   497
  498       |    int i, j, sign;                                                                       |   498
  499       |    char c;                                                                               |   499
  500       |                                                                                          |   500
  501     1 |    if ((sign = num) < 0)                  /* Check for negative number */                |   501
  502     2 |        num = -num;                                                                       |   502
  503     3 |    i = 0;                                                                                |   503
  504     4 |    do {                                                                                  |   504
  505     5 |        s??(i++??) = num % 10 + '0';       /* Get remainder of division */                |   505
  506     6 |    } while ((num /= 10) > 0);             /* Process one digit at a time */              |   506
  507     7 |    for (j = len - i; j > 0; j--, i++)     /* Pad spaces in the string array */           |   507
  508     8 |        s??(i??) = ' ';                                                                   |   508
  509     9 |    if (sign < 0)                          /* Insert '-' if negative number */            |   509
  510    10 |        s??(len-1??) = '-';                                                               |   510
  511    11 |    for (i = 0, j = len - 1 ; i < j; i++, j--) {  /* Reverse string in place */           |   511
  512    12 |        c = s??(i??);                                                                     |   512
  513    13 |        s??(i??) = s??(j??);                                                              |   513
  514    14 |        s??(j??) = c;                                                                     |   514
  515       |    }                                                                                     |   515
  516       |}                                                                                         |   516
  517       |                                                                                          |   517
  518       |                                                                                          |   518
  519       |/*--------------------------------------------------------------------*/                  |   519
  520       |/*                    Process Customer Request                        */                  |   520
  521       |/* This routine puts out the customer request screen and reads the     */                  |   521
  522       |/* input from the user.  CMD 1 (end job) and CMD 2 (go to main menu)    */                  |   522
  523       |/* are checked.  Routines to send the customer number and to receive    */                  |   523
  524       |/* customer information are called.                                     */                  |   524
  525       |/*--------------------------------------------------------------------*/                  |   525
  526       | 17                                                                                       |   526
  527       |void process_cust_req(int *end_pgm_flag, _SYSindara dsp_indic)                            |   527
  528       |{                                                                                         |   528
  529     1 |    QXXFORMAT(dspfptr, "DTLMNU    ");                                                     |   529
  530     2 |    fwrite(NULL, 0, 0, dspfptr);                                                          |   530
  531       |    memset(dsp_indic, IOFF, 99);                                                          |   531
  532     4 |    fread(&dtlmnu_dsp_i, sizeof(dtlmnu_dsp_i), 1, dspfptr);                               |   532
  533     5 |    if (dsp_indic??(98??) == ION) {        /* CMD 1, indic 99 */                          |   533
  534     6 |        print_msg(NORM_END);                                                              |   534
  535     7 |        *end_pgm_flag = ON;                                                               |   535
  536       |    }                                                                                     |   536
  537       |    else                                                                                  |   537
  538     8 |        if (dsp_indic??(97??) == IOFF)     /* CMD 2, indic 98 */                          |   538
  539     9 |            if (send_cust_req() == NOERROR) {                                            |   539
  540    10 |                if (rec_cust_info(end_pgm_flag, dsp_indic) == ERROR) {                    |   540
  541    11 |                    print_msg(ERR_END);                                                   |   541
  542    12 |                    *end_pgm_flag = ON;                                                   |   542
  543       |                }                                                                         |   543
  544       |            }                                                                             |   544
  545       |            else {                                                                        |   545
  546    13 |                print_msg(ERR_END);                                                       |   546
  547    14 |                *end_pgm_flag = ON;                                                       |   547
  548       |            }                                                                             |   548
  549       |}                                                                                         |   549
 5728CX1 R00 M02 880101 IBM C/400          CICFLIB/QICFPUB/CSRCDMUL          06/15/89 10:41:42          Page   11
LINE  STMT                              * * * * *  S O U R C E * * * * *                              SEQNO INCLUDE
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9----+----0
  550       |                                                                                          |   550
```

Figure 9-10 (Part 9 of 13). Source Program Example—CSRCDMUL (User-Defined Formats)

```
   551       |                                                                                                    |   551
   552       |/*--------------------------------------------------------------------*/                           |   552
   553       |/*                        Send Customer Request                       */                           |   553
   554       |/* This routine sends the customer number entered by the user to the    */                           |   554
   555       |/* remote program.  The number has already been read in.               */                           |   555
   556       |/*--------------------------------------------------------------------*/                           |   556
   557       | 18                                                                                                 |   557
   558       |send_cust_req()                                                                                     |   558
   559       |{                                                                                                   |   559
   560       |        strncpy(dtlreq_icf_o.custno, dtlmnu_dsp_i.custno, 6);                                       |   560
   561     2 |        QXXFORMAT(icffptr, "DTLREQ    ");                                                           |   561
   562     3 |        QXXPGMDEV(icffptr, "ICF00     ");                                                          |   562
   563     4 |        fwrite(&dtlreq_icf_o, sizeof(dtlreq_icf_o), 1, icffptr);                                    |   563
   564     5 |        return(pos_ret_code());                                                                     |   564
   565       |}                                                                                                   |   565
   566       |                                                                                                    |   566
   567       |                                                                                                    |   567
   568       |/*--------------------------------------------------------------------*/                           |   568
   569       |/*                     Receive Customer Information                   */                           |   569
   570       |/* This routine attempts to receive customer information from the      */                           |   570
   571       |/* target program.  A check is made for the following three conditions  */                           |   571
   572       |/* following the read operation: 1) The remote system timed out, 2) no  */                           |   572
   573       |/* data was received, and 3) data returned in an unexpected format.    */                           |   573
   574       |/* If the remote system times out (maj/min 0310), a message is written  */                           |   574
   575       |/* to the screen asking user to try again (enter 1) or to end the      */                           |   575
   576       |/* program (enter 2).                                                  */                           |   576
   577       |/* If no data is received (major 03) the request is sent again to the   */                           |   577
   578       |/* remote system.                                                      */                           |   578
   579       |/* If the record returns with the wrong record format, the program will */                           |   579
   580       |/* end on error.                                                       */                           |   580
   581       |/* If the remote program didn't find the customer, and the item number  */                           |   581
   582       |/* "000000" was returned, the main menu is displayed.                  */                           |   582
   583       |/*--------------------------------------------------------------------*/                           |   583
   584       | 19                                                                                                 |   584
   585       |rec_cust_info(int *end_pgm_flag, _SYSindara dsp_indic)                                              |   585
   586       |{                                                                                                   |   586
   587     1 |    QXXREADINVDEV(icffptr);                                                                         |   587
   588     2 |    fread(&dtlrsp_icf_i, sizeof(dtlrsp_icf_i), 1, icffptr);                                         |   588
   589     3 |    if (check_timeout() == 0) {                                                                     |   589
   590     4 |        QXXFORMAT(dspfptr, "TIMOUT    ");                                                          |   590
   591     5 |        fwrite(NULL, 0, 0, dspfptr);                                                                |   591
   592     6 |        fread(&timout_dsp_i_o, sizeof(timout_dsp_i_o), 1, dspfptr);                                 |   592
   593     7 |        if (timout_dsp_i_o.timrsp == '1')                                                           |   593
   594     8 |            return(rec_cust_info(end_pgm_flag, dsp_indic));                                         |   594
   595       |        else                                                                                        |   595
   596     9 |            if (timout_dsp_i_o.timrsp == '2') {                                                     |   596
   597    10 |                print_msg(NORM_END);                                                                |   597
   598    11 |                *end_pgm_flag = ON;                                                                 |   598
   599    12 |                return(NOERROR);                                                                    |   599
   600       |            }                                                                                       |   600
   601       |    }                                                                                               |   601
   602       |    else                                                                                            |   602
   603    13 |        if (check_no_data() == 0) {          /* No data received */                                 |   603
   604    14 |            QXXFORMAT(icffptr, "DTLREQ    ");                                                       |   604
   605    15 |            fwrite(&dtlreq_icf_o, sizeof(dtlreq_icf_o), 1, icffptr);                                |   605
   606    16 |            return(pos_ret_code());                                                                 |   606
   607       |        }                                                                                           |   607
   608       |        else {                                /* Was item found ? */                                |   608
   609    17 |            if (strncmp(dtlrsp_icf_i.custno, "000000", 6) != 1)                                     |   609
   610    18 |                return(NOERROR);                                                                    |   610
 5728CX1 R00 M02 880101 IBM C/400            CICFLIB/QICFPUB/CSRCDMUL              06/15/89 10:41:42          Page   12
LINE  STMT                              * * * * *   S O U R C E * * * * *                                  SEQNO INCLUDE
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9----+----0
   611       |            else {                           /* Check record format */                             |   611
   612    19 |                comm_fdbk = QXXIOFBK(icffptr);                                                      |   612
   613    20 |                if (strncmp(comm_fdbk->rec_format, "DTLRSP    ", 10) != 0) {                        |   613
   614    21 |                    print_msg(RCD_ERR);                                                             |   614
   615    22 |                    return(ERROR);                                                                  |   615
```

Figure 9-10 (Part 10 of 13). Source Program Example—CSRCDMUL (User-Defined Formats)

```
  616       |               }                                                                                    |   616
  617       |           }                                                                                        |   617
  618       |       }                                                                                            |   618
  619       |   strncpy(dtlscr_dsp_o.custn, dtlrsp_icf_i.custno, 6);                                             |   619
  620       |   strncpy(dtlscr_dsp_o.cname, dtlrsp_icf_i.dname, 30);                                             |   620
  621       |   strncpy(dtlscr_dsp_o.dlstr, dtlrsp_icf_i.dlstor, 6);                                             |   621
  622       |   strncpy(dtlscr_dsp_o.dslsm, dtlrsp_icf_i.dslstm, 9);                                             |   622
  623       |   strncpy(dtlscr_dsp_o.dspm1, dtlrsp_icf_i.dspm01, 9);                                             |   623
  624       |   strncpy(dtlscr_dsp_o.dspm2, dtlrsp_icf_i.dspm02, 9);                                             |   624
  625       |   strncpy(dtlscr_dsp_o.dstyd, dtlrsp_icf_i.dsttyd, 11);                                            |   625
  626       |   strncpy(dtlscr_dsp_o.dept, dtlrsp_icf_i.idept, 3);                                               |   626
  627    31 |   QXXFORMAT(dspfptr, "DTLSCR    ");       /* Display customer info */                             |   627
  628    32 |   fwrite(&dtlscr_dsp_o, sizeof(dtlscr_dsp_o), 1, dspfptr);                                        |   628
  629       |   memset(dsp_indic, IOFF, 99);                                                                     |   629
  630    34 |   fread(NULL, 0, 0, dspfptr);                                                                      |   630
  631    35 |   if (dsp_indic??(98??) == ION) {                                                                  |   631
  632    36 |       print_msg(NORM_END);                                                                         |   632
  633    37 |       *end_pgm_flag = ON;                                                                          |   633
  634       |   }                                                                                                |   634
  635    38 |   return(NOERROR);                                                                                 |   635
  636       |}                                                                                                   |   636
  637       |                                                                                                    |   637
  638       |                                                                                                    |   638
  639       |/*-----------------------------------------------------------------------*/                          |   639
  640       |/*                   Check For Successful Operation                    */                          |   640
  641       |/* If the major return code is 00 the last operation attempted was      */                          |   641
  642       |/* successful, otherwise an error occurred.                             */                          |   642
  643       |/*-----------------------------------------------------------------------*/                          |   643
  644       | 20                                                                                                 |   644
  645       |pos_ret_code()                                                                                      |   645
  646       |{                                                                                                   |   646
  647     1 |   get_access_to_fb();                                                                              |   647
  648     2 |   if (strncmp(dsp_icf_fdbk->major_ret_code, "00", 2) == 0)                                         |   648
  649     3 |       return(NOERROR);                                                                             |   649
  650       |   else                                                                                             |   650
  651     4 |       return(ERROR);                                                                               |   651
  652       |}                                                                                                   |   652
  653       |                                                                                                    |   653
  654       |                                                                                                    |   654
  655       |/*-----------------------------------------------------------------------*/                          |   655
  656       |/*                   Check for No Data Received                        */                          |   656
  657       |/* If the major return code is 03 then no data was received on an input */                          |   657
  658       |/* operation.                                                           */                          |   658
  659       |/*-----------------------------------------------------------------------*/                          |   659
  660       | 21                                                                                                 |   660
  661       |check_no_data()                                                                                     |   661
  662       |{                                                                                                   |   662
  663     1 |   get_access_to_fb();                                                                              |   663
  664     2 |   if (strncmp(dsp_icf_fdbk->major_ret_code, "03", 2) == 0)                                         |   664
  665     3 |       return(0);                                                                                   |   665
  666       |   else                                                                                             |   666
  667     4 |       return(1);                                                                                   |   667
  668       |}                                                                                                   |   668
  669       |                                                                                                    |   669
  670       |                                                                                                    |   670
  671       |/*-----------------------------------------------------------------------*/                          |   671
 5728CX1 R00 M02 880101 IBM C/400            CICFLIB/QICFPUB/CSRCDMUL                 06/15/89 10:41:42          Page   13
LINE  STMT                              * * * * *   S O U R C E * * * * *                                SEQNO INCLUDE
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9----+----0
  672       |/*                          Check for Timeout                          */                          |   672
  673       |/* If the major/minor return code is 0310, then a timeout occurred when */                          |   673
  674       |/* reading from an invited program device.  Return 1 if timout occurred */                          |   674
  675       |/* otherwise return 0.                                                  */                          |   675
  676       |/*-----------------------------------------------------------------------*/                          |   676
  677       | 22                                                                                                 |   677
  678       |check_timeout()                                                                                     |   678
  679       |{                                                                                                   |   679
  680     1 |   get_access_to_fb();                                                                              |   680
```

Figure 9-10 (Part 11 of 13). Source Program Example—CSRCDMUL (User-Defined Formats)

```
  681     2 |    if ((strncmp(dsp_icf_fdbk->major_ret_code, "03", 2) == 0) &&                                  |    681
  682       |        (strncmp(dsp_icf_fdbk->minor_ret_code, "10", 2) == 0))                                    |    682
  683     3 |        return(0);                                                                                |    683
  684       |    else                                                                                          |    684
  685     4 |        return(1);                                                                                |    685
  686       |}                                                                                                 |    686
  687       |                                                                                                  |    687
  688       |                                                                                                  |    688
  689       |/*------------------------------------------------------------------------*/                       |    689
  690       |/*                          Issue Detach                                  */                       |    690
  691       |/* A detach is sent to each program device to end the transaction with  */                       |    691
  692       |/* the remote system.                                                    */                       |    692
  693       |/*------------------------------------------------------------------------*/                       |    693
  694       | 23                                                                                               |    694
  695       |void detach()                                                                                     |    695
  696       |{                                                                                                 |    696
  697     1 |    QXXFORMAT(icffptr, "DETACH    ");                                                             |    697
  698     2 |    QXXPGMDEV(icffptr, "ICF00     ");                                                             |    698
  699     3 |    fwrite(NULL, 0, 0, icffptr);                                                                  |    699
  700     4 |    QXXPGMDEV(icffptr, "ICF01     ");                                                             |    700
  701     5 |    fwrite(NULL, 0, 0, icffptr);                                                                  |    701
  702     6 |    QXXPGMDEV(icffptr, "ICF02     ");                                                             |    702
  703     7 |    fwrite(NULL, 0, 0, icffptr);                                                                  |    703
  704     8 |    QXXPGMDEV(icffptr, "ICF03     ");                                                             |    704
  705     9 |    fwrite(NULL, 0, 0, icffptr);                                                                  |    705
  706       |}                                                                                                 |    706
  707       |                                                                                                  |    707
  708       |                                                                                                  |    708
  709       |/*------------------------------------------------------------------------*/                       |    709
  710       |/*                          End the Job                                   */                       |    710
  711       |/* A release is sent to each program device to end the sessions, and    */                       |    711
  712       |/* the ICF, display, and printer files are closed.                       */                       |    712
  713       |/*------------------------------------------------------------------------*/                       |    713
  714       | 24                                                                                               |    714
  715       |void end_job()                                                                                    |    715
  716       |{                                                                                                 |    716
  717     1 |    QXXRELEASE(icffptr, "ICF00     ");                                                            |    717
  718     2 |    QXXRELEASE(icffptr, "ICF01     ");                                                            |    718
  719     3 |    QXXRELEASE(icffptr, "ICF02     ");                                                            |    719
  720     4 |    QXXRELEASE(icffptr, "ICF03     ");                                                            |    720
  721     5 |    close_files();                                                                                |    721
  722       |}                                                                                                 |    722
  723       |                                                                                                  |    723
  724       |                                                                                                  |    724
  725       |/*------------------------------------------------------------------------*/                       |    725
  726       |/*                          Close the Files                               */                       |    726
  727       |/* Close the ICF, display, and print files.  If an error occurs issue   */                       |    727
  728       |/* another close which will be successful.                              */                       |    728
  729       |/*------------------------------------------------------------------------*/                       |    729
  730       | 25                                                                                               |    730
  731       |void close_files()                                                                                |    731
  732       |{                                                                                                 |    732
 5728CX1 R00 M02 880101 IBM C/400            CICFLIB/QICFPUB/CSRCDMUL                  06/15/89 10:41:42          Page   14
LINE  STMT                                  * * * * *   S O U R C E * * * * *                          SEQNO INCLUDE
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9----+----0
  733     1 |    if (fclose(icffptr) == ERROR)                                                                 |    733
  734     2 |        fclose(icffptr);                                                                          |    734
  735     3 |    if (fclose(dspfptr) == ERROR)                                                                 |    735
  736     4 |        fclose(dspfptr);                                                                          |    736
  737     5 |    if (fclose(prtfptr) == ERROR)                                                                 |    737
  738     6 |        fclose(prtfptr);                                                                          |    738
  739       |}                                                                                                 |    739
  740       |                                                                                                  |    740
  741       |                                                                                                  |    741
  742       |/*------------------------------------------------------------------------*/                       |    742
  743       |/*                          Print Message                                 */                       |    743
  744       |/* Write message and return code to print file.                          */                       |    744
```

Figure 9-10 (Part 12 of 13). Source Program Example—CSRCDMUL (User-Defined Formats)

```
745       |/*-------------------------------------------------------------------*/
746       | 26
747       |void print_msg(int mtype)
748       |{
749     1 |    get_access_to_fb();
750       |    strncpy(print_rec.major, dsp_icf_fdbk->major_ret_code, 2);
751       |    strncpy(print_rec.minor, dsp_icf_fdbk->minor_ret_code, 2);
752       |    strncpy(print_rec.filler, "                                ", 32);
753     5 |    fwrite("RETURN CODE:                      ", 36, 1, prtfptr);
754     6 |    fwrite(&print_rec, sizeof(print_rec), 1, prtfptr);
755     7 |    if (mtype == NORM_END)
756     8 |        fwrite("PROGRAM CSRCDMUL COMPLETED NORMALLY ", 36, 1, prtfptr);
757       |    else
758     9 |        if (mtype == RCD_ERR)
759    10 |            fwrite("RECORD FORMAT IS INCORRECT ON READ  ", 36, 1, prtfptr);
760       |        else
761    11 |            fwrite("PROGRAM ENDED DUE TO ERROR IN CMNFIL", 36, 1, prtfptr);
762       |}
763       |
764       |
765       |/*-------------------------------------------------------------------*/
766       |/*                Get Access to DSP/ICF Feedback                     */
767       |/* The feedback areas are updated after each display or ICF file I/O  */
768       |/* operation, and so the pointers must be updated to point to the "new" */
769       |/* feedback areas to get the return code.  The offset to the display/  */
770       |/* ICF feedback area is contained in the common I/O feedback and is    */
771       |/* added to the pointer to the common feedback area to get access to   */
772       |/* display/ICF feedback area.                                          */
773       |/*-------------------------------------------------------------------*/
774       | 27
775       |void get_access_to_fb()
776       |{
777     1 |    comm_fdbk = QXXIOFBK(icffptr);
778     2 |    dsp_icf_fdbk = (XXIOFB_DSP_ICF_T *)((char *)comm_fdbk +
779       |                                        comm_fdbk->file_dep_fb_offset);
780       |}
INCLUDE FILES  ---  FILE NO   NAME          ACTUAL FILE USED
                      1   stdio.h        QCC/H/STDIO
                      2   stddef.h       QCC/H/STDDEF
                      3   errno.h        QCC/H/ERRNO
                      4   signal.h       QCC/H/SIGNAL
                      5   ctype.h        QCC/H/CTYPE
                      6   stdarg.h       QCC/H/STDARG
                      7   errno.h        QCC/H/ERRNO
                      8   stdlib.h       QCC/H/STDLIB
                      9   stddef.h       QCC/H/STDDEF
                     10   string.h       QCC/H/STRING
                     11   xxfdbk.h       QCC/H/XXFDBK
                     12   xxasio.h       QCC/H/XXASIO
 5728CX1 R00 M02 880101 IBM C/400          CICFLIB/QICFPUB/CSRCDMUL          06/15/89 10:41:42          Page   15
                    * * * * *  E N D  O F  S O U R C E  * * * * *
```

Figure 9-10 (Part 13 of 13). Source Program Example—CSRCDMUL (User-Defined Formats)

Target Program Multiple-Session Inquiry

The following describes the C/400 target program for multiple-session inquiry program example.

Program Files: The C/400 multiple-session target program uses the following files:

CFILE — An ICF file used to send records to and receive records from the source program. It is done with the file-level INDARA DDS keyword, indicating a separate indicator area.

PFILE — A database file used to retrieve the record for the item requested from the remote system.

QSYSPRT — A printer file used to print error messages resulting from communications errors.

DDS Source: The DDS for the ICF file (CFILE) is illustrated in Figure 9-11.

```
A*****************************************************************
A*                                                               *
A*                          ICF FILE                             *
A*          USED IN TARGET MULTIPLE SESSION PROGRAM              *
A*                                                               *
A*****************************************************************
A                                      INDARA
A  05                                  RQSWRT
A  10                                  ALWWRT
A                                      INDTXT(10 '10 END TRANS.')
A  15                                  EOS
A  20                                  FAIL
A                                      INDTXT(20 '20 F ABORT ST')
A                                      RCVFAIL(25 'RECEIVED FAIL')
A  30                                  DETACH
A                                      INDTXT(30 '30>DETACH TGT')
A                                      RCVDETACH(44 'RECV DETACH')
A                                      RCVTRNRND(40 'END OF TRN')
A          R SNDPART
A                                      INVITE
A            RECTYP         1
A            ITEMNO         6
A            EDATA        130
A            FILL1         13
A          R RCVPART
A            RECID2         6
```

Figure 9-11. DDS Source for ICF File Used in Target Program Multiple Session Inquiry

The DDS source for the database file (PFILE) is illustrated in Figure 9-12.

```
5714PW1 R01M00  880301                SEU SOURCE LISTING                    10/16/87 07:43:14              PAGE 1
SOURCE FILE . . . . . . . QICFPUB/ICFLIB
MEMBER  . . . . . . . . . PFILE
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
  100      A                                  LIFO                                                        07/02/87
  200      A          R DBREC                                                                           05/06/87
  300      A            RECCUS         1                                                               10/01/87
  400      A            DBSEQ          6                                                               08/18/87
  500      A            DBDATA       130                                                               07/02/87
  600      A            DBFILL        13                                                               10/01/87
  700      A          K DBSEQ                                                                           07/04/87
                              * * * *  E N D  O F  S O U R C E  * * * *
```

Figure 9-12. DDS Source for ICF File Used in Target Program Multiple Session Inquiry

The command needed to create the ICF file is:

```
CRTICFF  FILE(CICFLIB/CFILE)  SRCFILE(CICFLIB/QICFPUB)  SRCMBR(CFILE)
   ACQPGMDEV(*NONE)  TEXT("TARGET ICF FILE FOR MULTIPLE SESSION PROGRAM")
```

The command needed to define the program device entry is:

```
OVRICFDEVE PGMDEV(ICF00) RMTLOCNAME(*REQUESTER)
```

Program Explanation: The following explains the structure of the program example illustrated in Figure 9-13 on page 9-33. The ICF file used is defined by the user, and uses externally described data formats (DDS). The reference numbers in the explanation below correspond to the numbers in the following program example.

In the following example, the ICF file used is externally described, although the structure must be defined in the program manually by the programmer. The C/400 compiler does not automatically map the file structures into the program.

1 This section defines the ICF file (CFILE) structure used in the program.

CFILE is the ICF file used to send records to and receive records from the remote system. CFILE is implemented with the file-level keyword, INDARA, indicating that a separate indicator area is used.

2 This structure is used to print the return code to the print file QSYSPRT. Output to QSYSPRT is by record, so the filler field is used to blank out characters that may have been set by a previous write operation to the print file. See **15** and **16**.

3 The structures used as parameters on the call to the RPG program SEARCH are defined. This is done so that SEARCH can be prototyped (parameter types are specified when prototyping).

4 The variables to be used globally by the program are defined here. The common and display/ICF feedback area pointers, and the file pointers are defined.

5 The routines, except the main routine, are prototyped so the compiler knows the type of value returned and the type of parameters passed, if any.

6 The ICF and printer files are opened for record input/output. The ICF file is opened with the separate indicator area option specified.

7 The separate indicator area is defined and initialized. The variable `icf_indic` is of type `_SYSindara`, which is a character array of size 99.

8 The program device ICF00 used by the program is explicitly acquired.

9 This program continues reading from and writing to the ICF file until either a detach is received from the source program, the source program ends abnormally, or an error occurs in the transaction.

A call to a procedure (section **14**) to close the files is made before the program ends.

10 This function reads data from the ICF file (CFILE) that was sent from the remote system.

A check is made to see if a detach was received (indicator 44 in CFILE) from the remote system, in which case control goes to **9** and the program ends. Note that the subscript to the indicator in the separate indicator area array is offset by one since the first indicator starts at position zero in the array.

If a turnaround indication is received (indicator 40 in CFILE), then the return code is checked for a 3431 (not a serious error) or a 0000. Any other return code received causes an error message to be printed and the program to end on return to 9. If the turnaround is not received, the program ends.

11 The structures passed as parameters on the call to SEARCH are declared using the definitions from section 3.

The number received from the remote system is used by the RPG program SEARCH to find the corresponding item or customer record. SEARCH (see Figure 9-14 on page 9-39) moves the information found in the record to the structure dbrec. If the record is not found, SEARCH puts 000000 in the dbseq field of the structure dbrec, and then sets the field reccus in the structure to 'I' (since the record ID does not exist).

The data is sent to the remote system with an invite, and the return code is checked for a successful operation. If the major return code is not 00, the program ends.

12 This function calls a procedure (section 17) to access the display/ICF feedback area. It then checks for a 3431 return code or a 00 major return code.

13 This function calls a procedure (section 17) to access the display/ICF feedback area. It then checks for a 00 major return code, which indicates that the last operation was successful.

14 This procedure closes the ICF and printer files. If an error should occur while attempting to close a file, another close operation is issued.

15 This procedure retrieves the return code to be printed from the display/ICF feedback area, and prints a normal end message to the print file.

16 This procedure retrieves the return code to be printed from the display/ICF feedback area, and prints an abnormal end message to the print file.

17 This procedure updates the common I/O feedback and the display/ICF feedback areas. The pointers must be reset after each I/O operation where information from the feedback areas is needed, so this procedure is called before checking the return code.

```
5728CX1 R00 M02 880101 IBM C/400          CICFLIB/QICFPUB/CTGTDMUL                06/15/89 11:02:45          Page   1
Command options:
  Program name . . . . . . . . . . . .:  CICFLIB/
  Source file name . . . . . . . . . .:  CICFLIB/
  Compiler options . . . . . . . . . .:  *NOAGGR  *NODEBUG   *NOEXPMAC    *GEN     *NOALWB IND  *NOLIS T     *NOPP ONLY   *NOS
                                      :  *NOSHOWINC  *SOURCE   *NOXREF
  Generation options . . . . . . . . .:  *NOLIST  *NOXREF   *GEN    *NOATR     *NODUMP      *NOSUB STR   *SUBS
                                      :  *NOALWBND
  User profile . . . . . . . . . . . .:  *USER
  Language level . . . . . . . . . . .:  *EXTENDED
  Source margins . . . . . . . . . . .:  1 - 32767
  Sequence columns. . . . . . .       :  none
  Authority . . . . . . . . . . . . . :  *CHANGE
  Text description . . . . . . . . . .:  Target C program for APPC
  Printer file . . . . . . . . . . . .:  *LIBL/
  Lines per page . . . . . . . . . . .:  66
Actual Program Source:
  Member . . . . . . . . . . . . . . .:  CTGTDMUL
  File . . . . . . . . . . . . . . . .:  QICFPUB
  Library . . . . . . . . . . . . . . :  CICFLIB
  Last change . . . . . . . . . . . . :  89/06/15  10:59:5
  Description . . . . . . . . . . . . :  Target C program for APPC
5728CX1 R00 M02 880101 IBM C/400          CICFLIB/QICFPUB/CTGTDMUL                06/15/89 11:02:45          Page   2
LINE  STMT                              * * * * *   S O U R C E * * * * *                                SEQNO INCLUDE
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9----+----0
    1       |#pragma linkage(SEARCH,OS)                  /* Define link to RPG pgm */                      |     1
    2       |                                                                                              |     2
    3       |/*--------------------------------------------------------------------*/                      |     3
    4       |/* This program will handle the request for either a customer number or */                     |     4
    5       |/* an item number.  This is accomplished by making the database file    */                     |     5
    6       |/* structure (key length, key position, record length, record size,     */                     |     6
    7       |/* etc.) the same for both files with only the record contents          */                     |     7
    8       |/* different.  An RPG program is called to search the database file.    */                     |     8
    9       |/* This program ends when a detach request is received from the source  */                     |     9
   10       |/* program.                                                             */                     |    10
   11       |/* Indicators associated with the ICF file are defined and are refer-   */                     |    11
   12       |/* enced for every I/O operation issued.                                */                     |    12
   13       |/*--------------------------------------------------------------------*/                      |    13
   14       |                                                                                              |    14
   15       |#define ION '1'                           /* Indicator set on */                             |    15
   16       |#define IOFF '0'                                                                             |    16
   17       |#define ERROR 1                           /* Error occurred */                               |    17
   18       |#define NOERROR 0                                                                            |    18
   19       |#define TRUE 1                                                                               |    19
   20       |#define FALSE 0                                                                              |    20
   21       |#include <stdio.h>                        /* Standard I/O header */                          |    21
   22       |#include <stdlib.h>                       /* General utilities */                            |    22
   23       |#include <stddef.h>                       /* Standard definitions */                         |    23
   24       |#include <string.h>                       /* String handling utilities */                    |    24
   25       |#include <xxfdbk.h>                       /* Feedback area structures */                     |    25
   26       |#include <xxasio.h>                       /* Indicator area structure */                     |    26
   27       | 1                                                                                           |    27
   28       |/*--------------------------------------------------------------------*/                      |    28
   29       |/* Define the ICF file's structure.                                     */                     |    29
   30       |/*--------------------------------------------------------------------*/                      |    30
   31       |                                                                                              |    31
   32       |struct {                                                                                      |    32
   33       |    char rectyp;                                                                              |    33
   34       |    char itemno??(6??);                                                                       |    34
   35       |    char edata??(130??);                                                                      |    35
   36       |    char fill1??(13??);                                                                       |    36
   37       |} sndpart_icf_o;                                                                              |    37
   38       |                                                                                              |    38
   39       |struct {                                                                                      |    39
   40       |    char recid2??(6??);                                                                       |    40
   41       |} rcvpart_icf_i;                                                                              |    41
   42       | 2                                                                                           |    42
```

Figure 9-13 (Part 1 of 6). Target Program Example—CTGTDMUL

```
   43       |/*--------------------------------------------------------------------*/                    |     43
   44       |/* Define the structure to be used to write to the print file.       */                    |     44
   45       |/*--------------------------------------------------------------------*/                    |     45
   46       |                                                                                           |     46
   47       |struct {                                                                                   |     47
   48       |    char major??(2??);                                                                     |     48
   49       |    char minor??(2??);                                                                     |     49
   50       |    char filler??(29??);                  /* Used for padding with blanks */               |     50
   51       |} print_rec;                                                                               |     51
   52       | 3                                                                                         |     52
   53       |/*--------------------------------------------------------------------*/                    |     53
   54       |/* Define parameters for call to RPG program SEARCH.                 */                    |     54
   55       |/*--------------------------------------------------------------------*/                    |     55
   56       |                                                                                           |     56
   57       |typedef struct {                          /* Declare structure type */                     |     57
   58       |    char reccus;                          /* for prototyping */                            |     58
   59       |    char dbseq??(6??);                                                                     |     59
   60       |    char dbdata??(130??);                                                                  |     60
   61       |    char dbfill??(13??);                                                                   |     61
 5728CX1 R00 M02 880101 IBM C/400          CICFLIB/QICFPUB/CTGTDMUL              06/15/89 11:02:45          Page   3
LINE  STMT                          * * * * *   S O U R C E * * * * *                                 SEQNO INCLUDE
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9----+----0
   62       |} dbrec_dta_i;                                                                             |     62
   63       |                                                                                           |     63
   64       |typedef struct {                          /* Declare structure type */                     |     64
   65       |    char num??(6??);                      /* for prototyping */                            |     65
   66       |} identifier;                                                                              |     66
   67       | 4                                                                                         |     67
   68       |/*--------------------------------------------------------------------*/                    |     68
   69       |/* Declare global variables.                                         */                    |     69
   70       |/*--------------------------------------------------------------------*/                    |     70
   71       |                                                                                           |     71
   72       |XXIOFB_T *comm_fdbk;                      /* Ptr to common I/O feedback */                 |     72
   73       |XXIOFB_DSP_ICF_T *dsp_icf_fdbk;           /* Ptr to dsp/ICF I/O feedback */                |     73
   74       |FILE *icffptr;                            /* Ptr to ICF file */                            |     74
   75       |FILE *prtfptr;                            /* Ptr to print file */                          |     75
   76       | 5                                                                                         |     76
   77       |int read_cfile(_SYSindara);                                                                |     77
   78       |int send_data(void);                                                                       |     78
   79       |int check_ret_code(void);                                                                  |     79
   80       |int pos_ret_code(void);                                                                    |     80
   81       |void close_files(void);                                                                    |     81
   82       |void print_norm_end(void);                                                                 |     82
   83       |void print_error_end(void);                                                                |     83
   84       |void get_access_to_fb(void);                                                               |     84
   85       |extern void SEARCH(identifier *, dbrec_dta_i *);  /* RPG program */                        |     85
   86       |                                                                                           |     86
   87       |main()                                                                                     |     87
   88       |{                                                                                          |     88
   89       |    _SYSindara icf_indic;                 /* ICF separate indic area */                    |     89
   90       |    int error_or_end = FALSE;             /* Set if I/O error occurred */                  |     90
   91       | 6                                                                                         |     91
   92       |    /*-------------------------------------------------------*/                           |     92
   93       |    /* Open the ICF and printer files.  If an error occurs the   */                        |     93
   94       |    /* program ends.                                         */                           |     94
   95       |    /*-------------------------------------------------------*/                           |     95
   96       |                                                                                           |     96
   97     1 |    if ((icffptr = fopen("CICFLIB/CFILE", "ab+ type=record indicators=y"))                 |     97
   98       |                                                                    == NULL)               |     98
   99     2 |        exit(ERROR);                                                                       |     99
  100     3 |    if ((prtfptr = fopen("CICFLIB/QSYSPRT", "wb type=record")) == NULL) {                  |    100
  101     4 |        fclose(icffptr);                                                                   |    101
  102     5 |        exit(ERROR);                                                                       |    102
  103       |    }                                                                                      |    103
  104       | 7                                                                                         |    104
  105       |    /*-------------------------------------------------------*/                           |    105
  106       |    /* Set up separate indicator area for the ICF file.      */                           |    106
```

Figure 9-13 (Part 2 of 6). Target Program Example—CTGTDMUL

```
  107       |    /*-------------------------------------------------------------*/                                  |    107
  108       |                                                                                                       |    108
  109       |    memset(icf_indic, IOFF, 99);                                                                       |    109
  110     7 |    QXXSINDARA(icffptr, icf_indic);                                                                    |    110
  111       | 8                                                                                                     |    111
  112       |    /*-------------------------------------------------------------*/                                  |    112
  113       |    /* Explicitly acquire a session.  If an error occurs the     */                                  |    113
  114       |    /* files will be closed and the program will end.            */                                  |    114
  115       |    /*-------------------------------------------------------------*/                                  |    115
  116       |                                                                                                       |    116
  117     8 |    QXXACQUIRE(icffptr, "ICF00     ");                                                                 |    117
  118     9 |    if (pos_ret_code() == ERROR) {                                                                     |    118
  119    10 |        print_error_end();                                                                             |    119
  120    11 |        close_files();                                                                                 |    120
  121    12 |        exit(ERROR);                                                                                   |    121
  122       |    }                                                                                                  |    122
 5728CX1 R00 M02 880101 IBM C/400            CICFLIB/QICFPUB/CTGTDMUL                  06/15/89 11:02:45          Page   4
LINE  STMT                              * * * * *   S O U R C E * * * * *                                  SEQNO INCLUDE
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9----+----0
  123       | 9                                                                                                     |    123
  124       |    /*-------------------------------------------------------------*/                                  |    124
  125       |    /* Process requests from the source program until a detach   */                                  |    125
  126       |    /* is received or an error occurs.                           */                                  |    126
  127       |    /*-------------------------------------------------------------*/                                  |    127
  128       |                                                                                                       |    128
  129    13 |    QXXPGMDEV(icffptr, "ICF00     ");                                                                  |    129
  130    14 |    while (error_or_end == FALSE) {                                                                    |    130
  131    15 |        if (read_cfile(icf_indic) == NOERROR) {                                                        |    131
  132    16 |            if (send_data() == ERROR)                                                                 |    132
  133    17 |                error_or_end = TRUE;                                                                   |    133
  134       |        }                                                                                              |    134
  135       |        else                                                                                           |    135
  136    18 |            error_or_end = TRUE;                                                                       |    136
  137       |    }                                                                                                  |    137
  138    19 |    close_files();                                                                                     |    138
  139       |}                                                                                                      |    139
  140       |                                                                                                       |    140
  141       |                                                                                                       |    141
  142       |/*---------------------------------------------------------------------*/                              |    142
  143       |/*                        Read ICF File                                */                              |    143
  144       |/* This routine issues a read operation to the program device.  The    */                              |    144
  145       |/* detach indication is checked, and if it was signalled, main is      */                              |    145
  146       |/* notified.  If the receive turnaround indicator isn't set, or an     */                              |    146
  147       |/* unexpected return code is received, main is notified of an error.   */                              |    147
  148       |/*---------------------------------------------------------------------*/                              |    148
  149       | 10                                                                                                    |    149
  150       |read_cfile(_SYSindara icf_indic)                                                                       |    150
  151       |{                                                                                                      |    151
  152     1 |    QXXFORMAT(icffptr, "RCVPART   ");                                                                  |    152
  153       |    memset(icf_indic, IOFF, 99);                                                                       |    153
  154     3 |    fread(&rcvpart_icf_i, sizeof(rcvpart_icf_i), 1, icffptr);                                          |    154
  155     4 |    if (icf_indic??(43??) == ION) {         /* Detach indicator (44) set */                            |    155
  156     5 |        print_norm_end();                                                                              |    156
  157     6 |        return(ERROR);                                                                                 |    157
  158       |    }                                                                                                  |    158
  159       |    else                                                                                               |    159
  160     7 |        if (icf_indic??(39??) == ION)       /* Rec. turnaround (40) set */                             |    160
  161     8 |            if (check_ret_code() == NOERROR)                                                           |    161
  162     9 |                return(NOERROR);                                                                       |    162
  163       |            else {                                                                                     |    163
  164    10 |                print_error_end();                                                                     |    164
  165    11 |                return(ERROR);                                                                         |    165
  166       |            }                                                                                          |    166
  167       |        else {                                                                                         |    167
  168    12 |            print_error_end();                                                                         |    168
  169    13 |            return(ERROR);                                                                             |    169
  170       |        }                                                                                              |    170
  171       |}                                                                                                      |    171
```

Figure 9-13 (Part 3 of 6). Target Program Example—CTGTDMUL

```
  172       |                                                                                                    |   172
  173       |                                                                                                    |   173
  174       |/*--------------------------------------------------------------------------*/                     |   174
  175       |/*                        Write to ICF File                                 */                     |   175
  176       |/* A request from the source program results in reading a single record */                         |   176
  177       |/* containing the requested customer or order number.  The response will*/                         |   177
  178       |/* be returned in a single record containing either the item or customer*/                         |   178
  179       |/* information, depending on the database content.                         */                     |   179
  180       |/* The response is sent to the source program by writing to the program */                         |   180
  181       |/* device file using format sndpart.                                       */                     |   181
  182       |/* An RPG program is called to look for the record in the data file     */                         |   182
  183       |/* using keyed access.  The address of structures must be passed since  */                         |   183
 5728CX1 R00 M02 880101 IBM C/400          CICFLIB/QICFPUB/CTGTDMUL              06/15/89 11:02:45          Page   5
LINE  STMT                              * * * * *   S O U R C E * * * * *                                  SEQNO INCLUDE
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9----+----0
  184       |/* record subfields aren't allowed as parameters, and dbrec is updated. */                        |   184
  185       |/* When the requested customer or item number is not found, "000000" is */                         |   185
  186       |/* propagated to the key field before the response is sent back to the  */                         |   186
  187       |/* source program by the RPG program SEARCH.                           */                         |   187
  188       |/*--------------------------------------------------------------------------*/                     |   188
  189       | 11                                                                                               |   189
  190       |send_data()                                                                                       |   190
  191       |{                                                                                                 |   191
  192       |    dbrec_dta_i dbrec;                                                                            |   192
  193       |    identifier ident;                                                                             |   193
  194       |                                                                                                  |   194
  195       |    strncpy(ident.num, rcvpart_icf_i.recid2, 6);                                                  |   195
  196     2 |    SEARCH(&ident, &dbrec);                                                                       |   196
  197     3 |    if (strncmp(dbrec.dbseq, "000000", 6) == 0)  /* Record not found */                           |   197
  198     4 |        sndpart_icf_o.rectyp = 'I';              /* Set rcd id to default */                      |   198
  199       |    else {                                                                                        |   199
  200     5 |        sndpart_icf_o.rectyp = dbrec.reccus;                                                      |   200
  201       |        strncpy(sndpart_icf_o.edata, dbrec.dbdata, 130);                                          |   201
  202       |    }                                                                                             |   202
  203       |    strncpy(sndpart_icf_o.itemno, dbrec.dbseq, 6);                                                |   203
  204     8 |    QXXFORMAT(icffptr, "SNDPART   ");                                                             |   204
  205     9 |    fwrite(&sndpart_icf_o, sizeof(sndpart_icf_o), 1, icffptr);                                    |   205
  206    10 |    if (pos_ret_code() == NOERROR)                                                                |   206
  207    11 |        return(NOERROR);                                                                          |   207
  208       |    else {                                                                                        |   208
  209    12 |        print_error_end();                                                                        |   209
  210    13 |        return(ERROR);                                                                            |   210
  211       |    }                                                                                             |   211
  212       |}                                                                                                 |   212
  213       |                                                                                                  |   213
  214       |                                                                                                  |   214
  215       |/*--------------------------------------------------------------------------*/                     |   215
  216       |/*                          Check Return Code                               */                     |   216
  217       |/* This routine checks the return code after a receive operation for    */                         |   217
  218       |/* 0000 and 3431.  Anything else is considered an error.                */                         |   218
  219       |/*--------------------------------------------------------------------------*/                     |   219
  220       | 12                                                                                               |   220
  221       |check_ret_code()                                                                                  |   221
  222       |{                                                                                                 |   222
  223     1 |    get_access_to_fb();                                                                           |   223
  224     2 |    if (strncmp(dsp_icf_fdbk->major_ret_code, "34", 2) == 0 &&                                    |   224
  225       |        strncmp(dsp_icf_fdbk->minor_ret_code, "31", 2) == 0)                                      |   225
  226     3 |        return(NOERROR);                                                                          |   226
  227       |    else                                                                                          |   227
  228     4 |        return(pos_ret_code());                                                                   |   228
  229       |}                                                                                                 |   229
  230       |                                                                                                  |   230
  231       |                                                                                                  |   231
  232       |/*--------------------------------------------------------------------------*/                     |   232
  233       |/*                     Check for Successful Operation                     */                     |   233
  234       |/* This routine checks the major return code of 00 to see if the last   */                         |   234
  235       |/* operation was successful.                                            */                         |   235
```

Figure 9-13 (Part 4 of 6). Target Program Example—CTGTDMUL

```
236       |/*--------------------------------------------------------------------*/
237       | 13
238       |pos_ret_code()
239       |{
240     1 |    get_access_to_fb();
241     2 |    if (strncmp(dsp_icf_fdbk->major_ret_code, "00", 2) == 0)
242     3 |        return(NOERROR);
243       |    else
244     4 |        return(ERROR);
 5728CX1 R00 M02 880101 IBM C/400          CICFLIB/QICFPUB/CTGTDMUL                    06/15/89 11:02:45                Page   6
LINE  STMT                              * * * * *   S O U R C E * * * * *                                    SEQNO INCLUDE
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9----+----0
245       |}
246       |
247       |
248       |/*--------------------------------------------------------------------*/
249       |/*                          Close the Files                           */
250       |/* Close the ICF and print files.  If an error occurs, issue another  */
251       |/* close which will be successful.                                    */
252       |/*--------------------------------------------------------------------*/
253       | 14
254       |void close_files()
255       |{
256     1 |    if (fclose(icffptr) == ERROR)
257     2 |        fclose(icffptr);
258     3 |    if (fclose(prtfptr) == ERROR)
259     4 |        fclose(prtfptr);
260       |}
261       |
262       |
263       |/*--------------------------------------------------------------------*/
264       |/*                       Print Normal End Message                     */
265       |/* Write normal end message and return code to print file.            */
266       |/*--------------------------------------------------------------------*/
267       | 15
268       |void print_norm_end()
269       |{
270     1 |    get_access_to_fb();
271       |    strncpy(print_rec.major, dsp_icf_fdbk->major_ret_code, 2);
272       |    strncpy(print_rec.minor, dsp_icf_fdbk->minor_ret_code, 2);
273       |    strncpy(print_rec.filler, "                                    ", 29);
274     5 |    fwrite("RETURN CODE:                    ", 33, 1, prtfptr);
275     6 |    fwrite(&print_rec, sizeof(print_rec), 1, prtfptr);
276     7 |    fwrite("CTGTDMUL HAS COMPLETED NORMALLY  ", 33, 1, prtfptr);
277       |}
278       |
279       |
280       |/*--------------------------------------------------------------------*/
281       |/*                     Print Abnormal End Message                     */
282       |/* Write abnormal end message and return code to print file.          */
283       |/*--------------------------------------------------------------------*/
284       | 16
285       |void print_error_end()
286       |{
287     1 |    get_access_to_fb();
288       |    strncpy(print_rec.major, dsp_icf_fdbk->major_ret_code, 2);
289       |    strncpy(print_rec.minor, dsp_icf_fdbk->minor_ret_code, 2);
290       |    strncpy(print_rec.filler, "                                    ", 29);
291     5 |    fwrite("RETURN CODE:                    ", 33, 1, prtfptr);
292     6 |    fwrite(&print_rec, sizeof(print_rec), 1, prtfptr);
293     7 |    fwrite("CTGTDMUL HAS COMPLETED ABNORMALLY", 33, 1, prtfptr);
294       |}
295       |
296       |
297       |/*--------------------------------------------------------------------*/
298       |/*                     Get Access to DSP/ICF Feedback                 */
299       |/* The feedback areas are updated after each display or ICF file I/O  */
300       |/* operation, and so the pointers must be updated to point to the "new" */
```

Figure 9-13 (Part 5 of 6). Target Program Example—CTGTDMUL

```
  301       |/* feedback areas to get the return code.  The offset to the display/   */                    |    301
  302       |/* ICF feedback area is contained in the common I/O feedback and is     */                    |    302
  303       |/* added to the pointer to the common feedback area to get access to    */                    |    303
  304       |/* display/ICF feedback area.                                           */                    |    304
  305       |/*---------------------------------------------------------------------*/                    |    305
 5728CX1 R00 M02 880101 IBM C/400            CICFLIB/QICFPUB/CTGTDMUL                 06/15/89 11:02:45          Page   7
LINE  STMT                              * * * * *   S O U R C E * * * * *                          SEQNO INCLUDE
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9----+----0
  306       | 17
  307       |void get_access_to_fb()                                                                    |    306
  308       |{                                                                                          |    307
  309     1 |    comm_fdbk = QXXIOFBK(icffptr);                                                         |    308
  310     2 |    dsp_icf_fdbk = (XXIOFB_DSP_ICF_T *)((char *)comm_fdbk +                                |    309
  311       |                                        comm_fdbk->file_dep_fb_offset);                    |    310
  312       |}                                                                                          |    311
                                                                                                        |    312
INCLUDE FILES  ---  FILE NO   NAME           ACTUAL FILE USED
                      1   stdio.h         QCC/H/STDIO
                      2   stddef.h        QCC/H/STDDEF
                      3   errno.h         QCC/H/ERRNO
                      4   signal.h        QCC/H/SIGNAL
                      5   ctype.h         QCC/H/CTYPE
                      6   stdarg.h        QCC/H/STDARG
                      7   errno.h         QCC/H/ERRNO
                      8   stdlib.h        QCC/H/STDLIB
                      9   stddef.h        QCC/H/STDDEF
                     10   string.h        QCC/H/STRING
                     11   xxfdbk.h        QCC/H/XXFDBK
                     12   xxasio.h        QCC/H/XXASIO
                    * * * * *  E N D  O F  S O U R C E  * * * * *
```

Figure 9-13 (Part 6 of 6). Target Program Example—CTGTDMUL

The following RPG/400 program is called by the C/400 target program to search the database file.

```
5728RG1 R02M00  891006          IBM AS/400 RPG/400                CICFLIB/SEARCH      06/15/89  16:57:04      Page      1
Compiler . . . . . :   IBM AS/400 RPG/400
Command Options:
  Program  . . . . . :   CICFLIB/SEARCH
  Source file  . . . . . :   CICFLIB/QICFPUB
  Source member  . . . . . :   *PGM
  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 . . . . . :   SEARCH
  File . . . . . :   QICFPUB
  Library  . . . . . :   CICFLIB
  Last Change  . . . . . :   05/05/89  11:59:53
  Description  . . . . . :   RPG program to search pfile for a record
5728RG1 R02M00  891006          IBM AS/400 RPG/400                CICFLIB/SEARCH      06/15/89  16:57:04      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  H*****************************************************************                 05/05/89
     200  H*                                                               *                 05/05/89
     300  H* THIS PROGRAM SEARCHES A DATABASE FILE FOR A RECORD WITH THE     *                 05/05/89
     400  H* SAME KEY AS THE ONE PASSED.  IF THE THE RECORD IS FOUND, THE    *                 05/05/89
     500  H* FIELDS IN THE DATA STRUCTURE PASSED IN WILL BE FILLED WITH THE  *                 05/05/89
     600  H* THE CUSTOMER OR ITEM INFO, AND IF THE RECORD IS NOT FOUND 000000 *                05/05/89
     700  H* IS PUT IN THE DBSEQ FIELD OF THE STRUCTURE PASSED.             *                 05/05/89
     800  H*                                                               *                 05/05/89
     900  H*****************************************************************                 05/05/89
          H                                                                                                  *****
    1000  FPFILE   IF  E           K        DISK                                            05/05/89
           RECORD FORMAT(S):  LIBRARY ICFLIB FILE PFILE.
                    EXTERNAL FORMAT DBREC RPG NAME DBREC
  A000000   INPUT  FIELDS FOR RECORD DBREC FILE PFILE FORMAT DBREC.
  A000001                                        1   1 RECCUS
  A000002                                        2   7 DBSEQ
  A000003                                        8 137 DBDATA
  A000004                                      138 150 DBFILL
    1100  IPCUST       DS                                                                    03/27/89
    1200  I                                        1   6 IDENT                               05/05/89
    1300  ICINFO       DS                                                                    03/27/89
    1400  I                                        1   1 REC                                 05/05/89
    1500  I                                        2   7 SEQ                                 05/05/89
    1600  I                                        8 137 DATA                                05/05/89
    1700  I                                      138 150 FILL                                05/05/89
    1800  C           *ENTRY    PLIST                                                       03/27/89
    1900  C                     PARM           PCUST                                        03/27/89
    2000  C                     PARM           CINFO                                        03/27/89
    2100  C           IDENT     CHAINPFILE               98     98 IF NOT FD   1            05/05/89
    2200  C                     MOVE RECCUS    REC                                          05/05/89
    2300  C                     MOVE DBSEQ     SEQ                                          05/05/89
    2400  C                     MOVE DBDATA    DATA                                         05/05/89
    2500  C   98                MOVE '000000'  SEQ                                          05/05/89
    2600  C                     RETRN                                                       03/27/89
          * * * * *  E N D   O F   S O U R C E   * * * * *
```

Figure 9-14 (Part 1 of 2). RPG/400 Program Example Called by CTGTDMUL

```
          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                CICFLIB/SEARCH        06/15/89  16:57:04      Page        3
                  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  PFILE
            DBREC
                      DBSEQ                   CHAR    6
 5728RG1 R02M00  891006            IBM AS/400 RPG/400                CICFLIB/SEARCH        06/15/89  16:57:04      Page        4
                        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  PFILE       DISK           1000D    2100
          DBREC                    1000D A000000
 Field References:
        FIELD       ATTR    REFERENCES (M=MODIFIED D=DEFINED)
* 7031  *ENTRY      PLIST      1800D
        CINFO       DS(150)    1300D    2000
        DATA        A(130)     1600D    2400M
        DBDATA      A(130)  A000003D    2400
* 7031  DBFILL      A(13)   A000004D
        DBSEQ       A(6)    A000002D    2300
* 7031  FILL        A(13)      1700D
        IDENT       A(6)       1200D    2100
        PCUST       DS(6)      1100D    1900
        REC         A(1)       1400D    2200M
        RECCUS      A(1)    A000001D    2200
        SEQ         A(6)       1500D    2300M    2500M
        '000000'    LITERAL    2500
 Indicator References:
        INDICATOR  REFERENCES (M=MODIFIED D=DEFINED)
        98           2100M    2500
     * * * * *  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                CICFLIB/SEARCH        06/15/89  16:57:04      Page        5
                      M e s s a g e   S u m m a r y
* QRG7031 Severity:  00   Number:    3
          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                CICFLIB/SEARCH        06/15/89  16:57:04      Page        6
                       F i n a l   S u m m a r y
 Message Count:  (by Severity Number)
           TOTAL    00     10     20     30     40     50
             3       3      0      0      0      0      0
 Program Source Totals:
    Records . . . . . . . . . . :   26
    Specifications  . . . . . . :   17
    Table Records . . . . . . . :   0
    Comments  . . . . . . . . . :   9
 PRM has been called.
 Program SEARCH is placed in library CICFLIB. 00 highest Error-Severity-Code.
          * * * * *  E N D   O F   C O M P I L A T I O N   * * * * *
```

Figure 9-14 (Part 2 of 2). RPG/400 Program Example Called by CTGTDMUL

Chapter 10. Communications Applications with COBOL/400

Previous chapters in this manual describe the functions provided by ICF. This chapter introduces you to the COBOL/400 interfaces for ICF and provides program examples.

Two application examples are presented in this chapter. For each example, both the source and target programs are provided. Each program is written first with user-defined formats (data description specifications, DDS) and then with system-supplied formats.

The first example in this section is a batch data transfer application using a single session. The second example is a multiple-session inquiry application using one display file and four ICF sessions.

Not all programming considerations or techniques are illustrated in each example in this section. Review these examples and the examples provided in the appropriate programming manual before beginning application design and coding.

Note: The examples in this section were written to the APPC communications type. Minor changes might be required if another communications type is used.

Introduction to the COBOL/400 Interface

Before you write a COBOL/400 communications application, you must understand the high-level language interface provided by COBOL/400.

The operations you use in the communications portion of your program are similar to work station operations. ICF files are defined as transaction files in COBOL/400. In the noncommunications portion of your program, you can use all noncommunications operations you normally use to process data that is sent or received between your program and the remote program.

Table 10-1 briefly introduces the COBOL/400 statements you use in the communications portion of your program.

Table 10-1 (Page 1 of 2). COBOL/400 Statements

ICF Operation	COBOL/400 Statement	Function
Open	OPEN	Opens the ICF file
Acquire	ACQUIRE	Establishes a session
Get-Attributes	ACCEPT	Gets the attributes of a session
Read	READ[1]	Receives data from a specific program device
Read-from- invited-program- devices	READ[1]	Receives data from any invited program device[2]
Write	WRITE	Performs many of the ICF communications functions within a session
Release	DROP	Releases the session

Table 10-1 (Page 2 of 2). COBOL/400 Statements

ICF Operation	COBOL/400 Statement	Function
Close	CLOSE	Closes the ICF file

1 A COBOL/400 read operation can be directed either to a specific program device or to all invited program devices. The support provided by the COBOL/400 compiler determines whether to issue an ICF read or read-from-invited-program-devices operation based on the presence of a format name or a terminal name on the read operation. For example, if a READ is sent with a specific format or terminal specified, the read operation is interpreted as an ICF read operation. Refer to the COBOL/400 language manual for more information.

2 The read-from-invited-program-devices operation could complete without data if the timer interval established with either the timer function or WAITRCD expires, or your job is ended (controlled).

Refer to *COBOL/400 User's Guide* for details on the syntax and function of each operation.

COBOL/400 Status Values

You must also understand the relationship between COBOL/400 file status values and ICF major/minor return codes.

Table 10-2 shows the file status values as returned by COBOL/400 after an input/output (I/O) operation for each major and minor return code. Use this list to determine the ICF return code or group of codes that corresponds to the file status value.

Table 10-2. File Status Values for Major and Minor Return Codes

ICF Return Code	COBOL/400 Return Code
00xx	00
02xx	9A
03xx	00
0309	9A
04xx	9I
0800	00
1100	10
34xx	9G
80xx	30
81xx	92
82xx	9C
83xx	9N
83E0	9K

Example Programs

The programs presented in this section are:

- Example I (Batch Data Transfer)

  Figure 10-1 shows a batch data transfer program that reads a database file and sends the data to a remote system. When the source program finishes sending its records, it sends an indication that it is done sending records to the target program. The target program then starts sending its records until it reaches end-of-file. At end-of-file, the target program sends a detach indication to the source program. The two programs end their sessions.

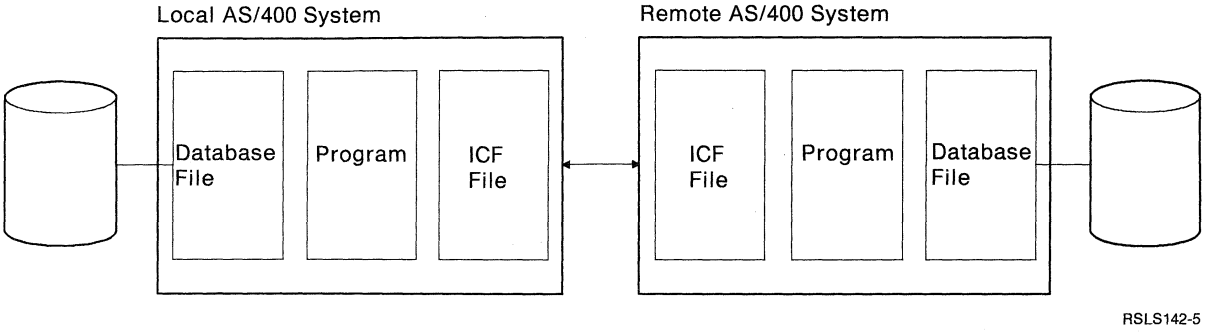


Figure 10-1. Batch Data Transfer

- Example II (Multiple-Session Inquiry)

  Figure 10-2 on page 10-4 shows an inquiry program that accepts inquiries from a display device, sends the request to one of four remote AS/400 systems, and waits for a response to the inquiry. Based on the input received from the display device, the program determines the target program to which it sends the inquiry request. The same program resides in each of the remote systems.

  Figure 10-2 on page 10-4 contains a display device and four ICF communication program devices.

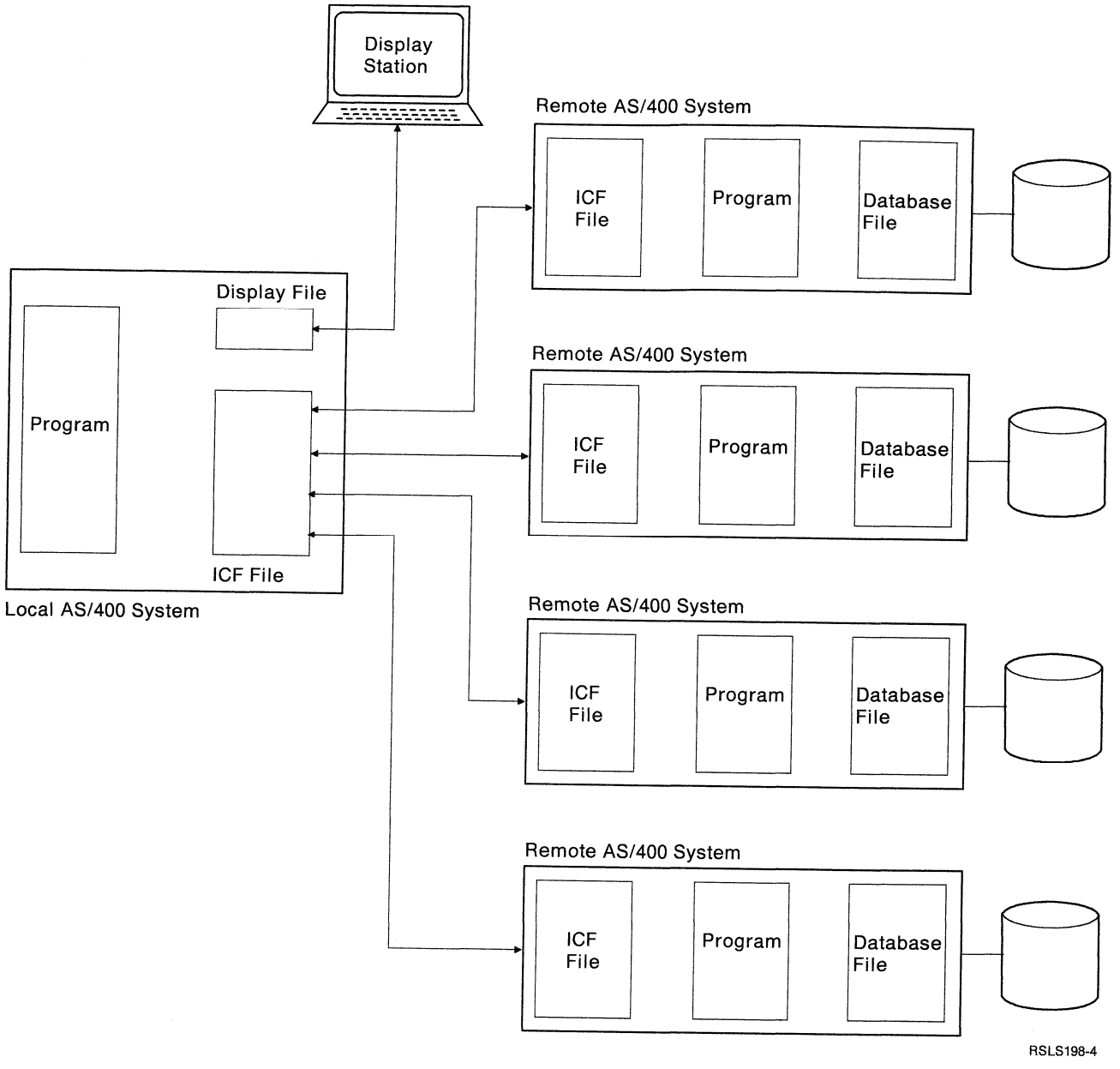


Figure 10-2. Multiple-Session Inquiry

The rest of this chapter discusses the details of the two application examples. The DDS file, program listings, and an explanation of the programs are included.

Batch Data Transfer (Example I)

The following figures show a batch data transfer program. A source AS/400 system program communicates with a target program on another AS/400 system using the ICF support. The source program starts a target program on a remote AS/400 system and transfers a file to that target program.

The target program responds, after receiving an indication that the source is done sending, by reading its own file and then sends the records to the source program until it reaches end of file. At end-of-file, the target program sends a detach request to the source program and ends its session.

Both the source program and the target program are described.

Transaction Flow of the Batch Data Transfer (Example I)

In Figure 10-3, the source program issues an evoke to start a program at the remote AS/400 system.

Note: An acquire operation is not necessary since the device was acquired during the open operation. The device was acquired during the open operation because the ACQPGMDEV parameter was used when the ICF file was created.

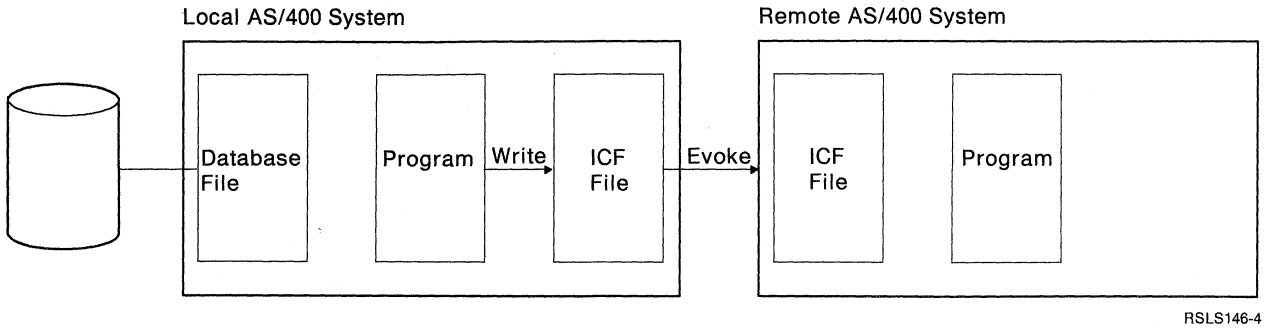


Figure 10-3. Evoke Request Starts a Target Program

After issuing the evoke request, the source program sends a database file to the target program, which prints the records as shown in Figure 10-4.

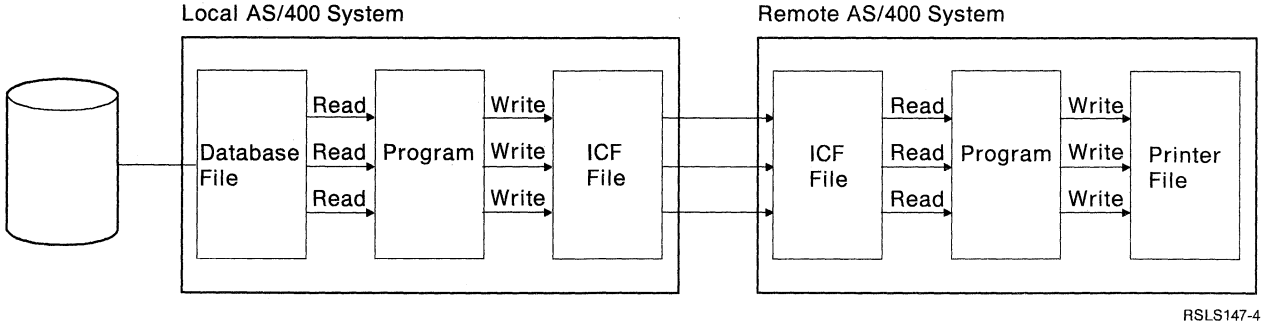


Figure 10-4. Target Program Prints Records

After the target program receives and prints the file, a database file is sent to the source program. The source program prints the records as they are received, as shown in Figure 10-5.

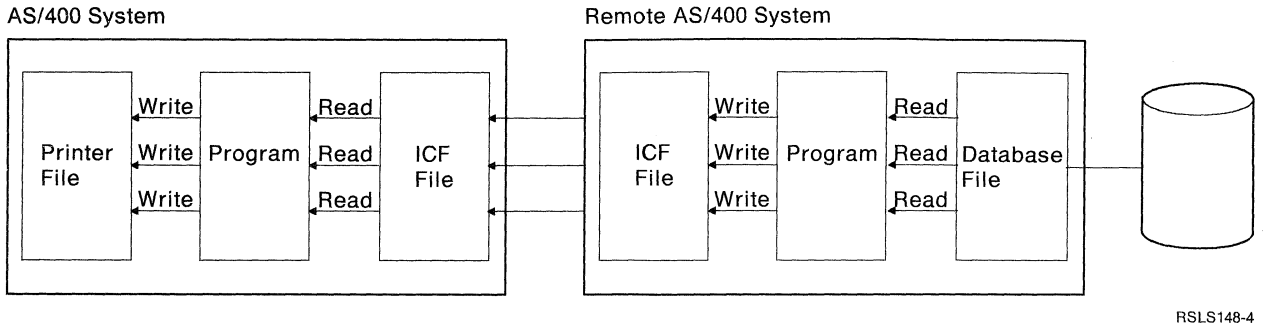


Figure 10-5. Source Program Prints the Received Records

Once all the records have been sent by the target program, the target program issues a detach to the source program to end the transaction, as shown in Figure 10-6.

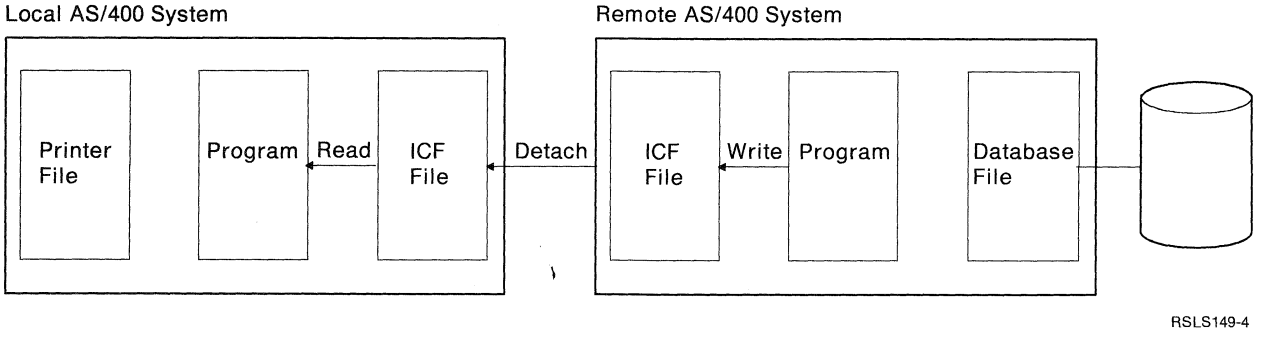


Figure 10-6. Target Program Ends the Transaction

Source Program Batch Transfer (Example I)

The following describes the COBOL/400 batch data transfer source program.

Program Files: The COBOL/400 batch transfer source program uses the following files:

SRCICF	An ICF file used to send records to and receive records from the target program.
DBFILE	A database file that contains the records to be sent to the target program.
QPRINT	A printer file used to print the records received from the target program.

DDS Source: The DDS source used in the ICF file is illustrated in Figure 10-7 on page 10-7. The other files (DBFILE and QPRINT) are program-described and therefore require no DDS.

```
A*************************************************************
A*                                                           *
A*                          ICF FILE                         *
A*              USED IN BATCH DATA TRANSFER PROGRAM          *
A*                                                           *
A*************************************************************
A*
A*  FILE LEVEL INDICATORS:
A*
A                                     INDARA
A*
A                                     RCVTRNRND(15 'END OF DATA')
A*
A  30                                 DETACH
A*
A                                     INDTXT(30 '30->DETACH TARG-
A                                     ET PROGRAM.')
A*
A                                     RCVDETACH(35 'RECEIVED    -
A                                     DETACHED.')
A*
A*
A*************************************************************
A*                      ICF RECORD FORMATS                   *
A*************************************************************
          R RCVDATA
            RCVFLD        80A
          R SNDDATA
            SNDFLD        80A
          R EVOKPGM
A  50                                 EVOKE(&LIB/&PGMID)
A  50                                 SECURITY(2 'PASSWRD' +
                                              3 'USERID')
A           PGMID         10A  P
A           LIB           10A  P
A         R ENDREC
A         R INVITE
A  45                                 INVITE
```

Figure 10-7. DDS Source for ICF File Used in Batch Data Transfer Program

ICF File Creation and Program Device Entry Definition: The command needed to create the ICF file is:

```
CRTICFF  FILE(ICFLIB/SRCICF)  SRCFILE(ICFLIB/QICFPUB)  SRCMBR(SRCICF)
   ACQPGMDEV(PGMDEVA)  TEXT('ICF FILE FOR BATCH DATA TRANSFER')
```

The command needed to define the program device entry is:

```
ADDICFDEVE FILE(ICFLIB/SRCICF) PGMDEV(PGMDEVA) RMTLOCNAME(CHICAGO)
```

Program Explanation: The following describes the structure of the program examples illustrated in Figure 10-8 on page 10-11 and Figure 10-9 on page 10-17. The ICF file used in the first example id defined by the user, and uses externally described data formats (DDS). The second example uses the same file, but uses program-described data and system-supplied formats. The reference numbers in the explanation below correspond to the numbers in the following program examples.

Although the basic structure of the two examples provided is the same, there are differences because of the use of user-defined formats and system-supplied formats.

Differences between the first and second example are documented as notes in each of the descriptions.

1 This section identifies the files used in the program. SRCICF is the ICF file used to send records to the target program.

This section also contains declarations of I/O variables, work areas and constants needed. MAJ-MIN contains the major/minor return code from the I/O feedback area.

Note: In the program using system-supplied formats, the input records for SRCICF are explicitly coded since SRCICF is now treated as a program-described file. The system-supplied file, QICDMF, can be used instead of SRCICF. This can be done by specifying QICDMF in the file specification, or by using an override ICF file (OVRICFF) command to change the file name from SRCICF to QICDMF. The OVRICFF command can also be used to change the ACQPGMDEV parameter of the file.

2 This section defines the error handling for the program. The major/minor return code is checked to determine whether the error is recoverable. If the error is recoverable (major/minor code 83xx or 03xx), it sets a flag (ERR-SW) to 1 and returns to the program.

If any other error has occurred, the program writes the feedback data to a file (DFILE), calls a program to print DFILE, and then ends.

3 The program opens the files that are going to be used as follows:

DBFILE	The database file used as input for transmitting to the target program.
QPRINT	The printer file used as output for records received
SRCICF	The ICF file used to send data to and receive data from the target program

Because the ICF file was created using the ACQPGMDEV parameter, the program device is automatically acquired when the file is opened. Therefore, no acquire operation is coded in the program.

4 If the ERR-SW flag is set to 1, indicating that a recoverable error has occurred, the program determines whether the open retry count limit (9) has been exceeded. If it has, the program goes to **11**. If the limit count is less than 9, one is added to the count and control passes to **10**. Control then passes to **3** to attempt to open the file.

5 The evoke request is built and sent to the remote system. Because the DDS for the record format only specifies the field identifiers with the record, the code in this section of the program moves the literal value CTDBATCL to the field *PGMID*, and ICFLIB to the field *LIB*. Indicator 50 is set to indicate that the program start request is to be sent.

When the program start request is received at the remote system, ICFLIB is searched for CTDBATCL and that program is then started. CTDBATCL is a control language (CL) program that contains the following statements:

```
ADDLIBLE ICFLIB
CALL ICFLIB/CTDBAT
```

Note: In the program using system-supplied formats, the library and program (ICFLIB/CTFBATCL) are specified as part of the $$EVOKNI

format. CTFBATCL is a CL program that contains the following statements:

```
ADDLIBLE ICFLIB
CALL ICFLIB/CTFBAT
```

6 Data is read from the database file. If the record read is end-of-file, the program sets the EOF-DBFILE-SW, performs routine 12 to invite the program device, and goes to 7.

If it is not the last record, the data is moved to the output buffer and the program goes to 9 to write the record to the program device. When control returns from 9, the process is repeated (DBFILE read) until end-of-file.

7 Data is read from the ICF file (SRCICF). If the operation was successful, and data was received (major return code not = '03'), then the data is written to the printer file (QPRINT). If an error occurs on the read, control passes to 10 to attempt recovery. When control returns from 10, control passes to 3 to attempt to open the files.

Data is read until the detach indication is received from the target program. When detach is received, indicator 35 is set on, as defined by the RCVDETACH keyword in the DDS for the ICF file. Note that RCVDETACH is a file-level keyword.

Note: In the program using system-supplied formats, the minor return code of '08' is checked to verify whether detach is received.

8 After the detach request has been received, this routine writes the following message to the print file:

```
CSDBAT HAS COMPLETED NORMALLY
```

The files are closed. The program device is automatically released as a result of the close operation.

Note: In the program using system-supplied formats, the program name is CSFBAT.

9 This routine is performed to write data to the ICF file using the format SNDDATA. If an error occurs on the write, control passes to 10 and then finally to 3.

Note: In the program using system-supplied formats, the $$SENDNI format is used instead of the user-defined SNDDATA format.

10 This routine is performed when a recoverable session error has occurred. It closes the files (SRCICF, QPRINT and DBFILE) and sets the error switch (ERR-SW) to 0. Control then returns to the statement immediately following the PERFORM statement that passed control to this routine.

11 This routine is performed when the application receives a major/minor return code of 03xx or 83xx after it has already attempted to recover 9 times from session errors that caused the error. The program is designed to tolerate only nine failures. It writes the following message to the print file:

```
CSDBAT HAS COMPLETED ABNORMALLY
```

The session is ended.

Note: In the program using system-supplied formats, the program name is CSFBAT.

12 This routine is performed to issue an invite request to the program device using format INVITE. If an error occurs, control passes to **10** and then finally to **3**.

Note: In the program using system-supplied formats, the $$SEND format is used instead of the user-defined INVITE format.

```
5728CB1 R01 M02 881028              IBM AS/400  COBOL/400          ICFLIB/CSDBAT          03/01/89 13:44:14       Page    1
Program name . . . . . . . . . . . . : CSDBAT in ICFLIB
Source file  . . . . . . . . . . . . : QICFPUB in ICFLIB     Member - CSDBAT     02/28/89 13:51:38
Compiler option  . . . . . . . . . . : *NONE
Code generation option . . . . . . . : *NONE
Code generation severity level . . . : 29
Print file . . . . . . . . . . . . . : QSYSPRT in *LIBL
FIPS flagging option . . . . . . . . : *NOFIPS *NOSEG *NODEB *NOOBSOLETE
Text not available for message LBX6039 file QLBLMSG.
Flagging level . . . . . . . . . . . : 0
Replace existing program . . . . . . : *YES
User profile . . . . . . . . . . . . : *USER
Authority  . . . . . . . . . . . . . : *CHANGE
Text . . . . . . . . . . . . . . . . : cobol batch file transfer using dds (source)
Compiler . . . . . . . . . . . . . . : IBM AS/400  COBOL/400
5728CB1 R01 M02 881028              COBOL SOURCE LISTING          ICFLIB/CSDBAT          03/01/89 13:44:14       Page    2
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
    1  000100 IDENTIFICATION DIVISION.                                                              02/21/89
    2  000200 PROGRAM-ID.                CSDBAT.                                                    02/21/89
       000300*************************************************************************            02/21/89
       000400*  THIS IS A BATCH FILE TRANSFER PROGRAM THAT READS A SEQUENTIAL   *                   02/21/89
       000500*  FILE AND SENDS THE RECORDS TO THE REMOTE SYSTEM UNTIL THE END   *                   02/21/89
       000600*  OF FILE IS REACHED. AT THIS TIME, THE PROGRAM STOPS SENDING     *                   02/21/89
       000700*  AND STARTS RECEIVING RECORDS FROM THE REMOTE SYSTEM UNTIL A     *                   02/21/89
       000800*  A DETACH INDICATION IS RECEIVED.                                *                   02/21/89
       000900*************************************************************************            02/21/89
    3  001000 ENVIRONMENT DIVISION.                                                              10/16/87
    4  001100 CONFIGURATION SECTION.                                                             10/16/87
    5  001200 SOURCE-COMPUTER.           IBM-AS400.                                              02/21/89
    6  001300 OBJECT-COMPUTER.           IBM-AS400.                                              02/21/89
    7  001400 SPECIAL-NAMES.             I-O-FEEDBACK IS FEEDBACK-AREA                           10/16/87
    8  001500                            OPEN-FEEDBACK IS OPEN-FBA.                              10/16/87
    9  001600 INPUT-OUTPUT SECTION.                                                              10/16/87
   10  001700 FILE-CONTROL.                                                                      10/16/87
       001800* 1                                                                                 10/16/87
   11  001900     SELECT DBFILE ASSIGN TO DATABASE-DBFILE.                                       10/16/87
   12  002000     SELECT SRCICF ASSIGN TO WORKSTATION-SRCICF-SI                                  10/16/87
   13  002100         ORGANIZATION IS TRANSACTION                                                10/16/87
   14  002200         FILE STATUS IS STATUS-IND MAJ-MIN.                                         10/16/87
   15  002300     SELECT DFILE ASSIGN TO DATABASE-HEXDUMP.                                       10/16/87
   16  002400     SELECT QPRINT ASSIGN TO PRINTER-QSYSPRT.                                       10/16/87
   17  002500 DATA DIVISION.                                                                     10/16/87
   18  002600 FILE SECTION.                                                                      10/16/87
   19  002700 FD  DBFILE                                                                         10/16/87
   20  002800     LABEL RECORDS ARE STANDARD.                                                    10/16/87
   21  002900 01  DBREC.                                                                         10/16/87
   22  003000     05  DBREC-DATA                  PIC X(80).                                     10/16/87
   23  003100 FD  SRCICF                                                                         10/16/87
   24  003200     LABEL RECORDS ARE STANDARD.                                                    10/16/87
   25  003300 01  ICFREC.                                                                       10/16/87
   26  003400     COPY DDS-ALL-FORMATS OF SRCICF.                                                10/16/87
   27 +000001       05  SRCICF-RECORD PIC X(80).                                 <-ALL-FMTS
      +000002*  INPUT FORMAT:RCVDATA    FROM FILE SRCICF     OF LIBRARY ICFLIB   <-ALL-FMTS
      +000003*                                                                   <-ALL-FMTS
   28 +000004       05  RCVDATA-I     REDEFINES SRCICF-RECORD.                   <-ALL-FMTS
   29 +000005           06 RCVFLD           PIC X(80).                          <-ALL-FMTS
      +000006* OUTPUT FORMAT:RCVDATA    FROM FILE SRCICF     OF LIBRARY ICFLIB   <-ALL-FMTS
      +000007*                                                                   <-ALL-FMTS
   30 +000008       05  RCVDATA-O     REDEFINES SRCICF-RECORD.                   <-ALL-FMTS
   31 +000009           06 RCVFLD           PIC X(80).                          <-ALL-FMTS
      +000010*  INPUT FORMAT:SNDDATA    FROM FILE SRCICF     OF LIBRARY ICFLIB   <-ALL-FMTS
      +000011*                                                                   <-ALL-FMTS
   32 +000012       05  SNDDATA-I     REDEFINES SRCICF-RECORD.                   <-ALL-FMTS
   33 +000013           06 SNDFLD           PIC X(80).                          <-ALL-FMTS
```

Figure 10-8 (Part 1 of 6). Source Program Example—CSDBAT (User-Defined Formats)

```
      +000014* OUTPUT FORMAT:SNDDATA    FROM FILE SRCICF     OF LIBRARY ICFLIB           <-ALL-FMTS
      +000015*                                                                          <-ALL-FMTS
   34 +000016       05  SNDDATA-O     REDEFINES SRCICF-RECORD.                       <-ALL-FMTS
   35 +000017           06 SNDFLD            PIC X(80).                              <-ALL-FMTS
      +000018*  INPUT FORMAT:EVOKPGM    FROM FILE SRCICF     OF LIBRARY ICFLIB           <-ALL-FMTS
      +000019*                                                                          <-ALL-FMTS
      +000020*      05  EVOKPGM-I     REDEFINES SRCICF-RECORD.                       <-ALL-FMTS
      +000021* OUTPUT FORMAT:EVOKPGM    FROM FILE SRCICF     OF LIBRARY ICFLIB           <-ALL-FMTS
5728CB1 R01 M02 881028                COBOL SOURCE LISTING              ICFLIB/CSDBAT        03/01/89 13:44:14       Page     3
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
      +000022*                                                                          <-ALL-FMTS
   36 +000023       05  EVOKPGM-O     REDEFINES SRCICF-RECORD.                       <-ALL-FMTS
   37 +000024           06 PGMID             PIC X(10).                              <-ALL-FMTS
   38 +000025           06 LIB               PIC X(10).                              <-ALL-FMTS
      +000026*  INPUT FORMAT:ENDREC     FROM FILE SRCICF     OF LIBRARY ICFLIB           <-ALL-FMTS
      +000027*                                                                          <-ALL-FMTS
      +000028*      05  ENDREC-I      REDEFINES SRCICF-RECORD.                       <-ALL-FMTS
      +000029* OUTPUT FORMAT:ENDREC     FROM FILE SRCICF     OF LIBRARY ICFLIB           <-ALL-FMTS
      +000030*                                                                          <-ALL-FMTS
      +000031*      05  ENDREC-O      REDEFINES SRCICF-RECORD.                       <-ALL-FMTS
      +000032*  INPUT FORMAT:INVITE     FROM FILE SRCICF     OF LIBRARY ICFLIB           <-ALL-FMTS
      +000033*                                                                          <-ALL-FMTS
      +000034*      05  INVITE-I      REDEFINES SRCICF-RECORD.                       <-ALL-FMTS
      +000035* OUTPUT FORMAT:INVITE     FROM FILE SRCICF     OF LIBRARY ICFLIB           <-ALL-FMTS
      +000036*                                                                          <-ALL-FMTS
      +000037*      05  INVITE-O      REDEFINES SRCICF-RECORD.                       <-ALL-FMTS
   39  003500 FD  DFILE                                                                        10/16/87
   40  003600     LABEL RECORDS ARE STANDARD.                                                  10/16/87
   41  003700 01  DUMPREC.                                                                    10/16/87
   42  003800     05  DUMP-MAJ-MIN                PIC X(4).                                   10/16/87
   43  003900     05  DUMP-RECORD                 PIC X(400).                                 10/16/87
   44  004000 FD  QPRINT                                                                       10/16/87
   45  004100     LABEL RECORDS ARE OMITTED.                                                   10/16/87
   46  004200 01  PRINTREC             PIC X(132).                                           10/16/87
   47  004300 WORKING-STORAGE SECTION.                                                         10/16/87
   48  004400 77  STATUS-IND                  PIC X(2).                                       10/16/87
   49  004500 77  MAJ-MIN-SAV                 PIC X(4).                                       10/16/87
   50  004600 77  EOF-DBFILE-SW               PIC X VALUE "0".                                10/16/87
   51  004700 77  ERR-SW                      PIC X VALUE "0".                                10/16/87
   52  004800 77  INDON                       PIC 1 VALUE B"1".                               10/16/87
   53  004900 77  INDOFF                      PIC 1 VALUE B"0".                               10/16/87
   54  005000 77  OPEN-COUNT                  PIC 9(1) VALUE 0.                               10/16/87
   55  005100 77  LEN                         PIC 9(10)V9(5) COMP.                            10/16/87
       005200                                                                                  10/16/87
   56  005300 77  CMD2                        PIC X(31)                                       10/16/87
   57  005400       VALUE "CPYF HEXDUMP *LIST PRTFMT(*HEX)".                                   10/16/87
   58  005500 01  CMNF-INDIC-AREA.                                                           10/16/87
   59  005600     05  CMNF-INDIC              PIC 1 OCCURS 99 TIMES                           10/16/87
   60  005700                                  INDICATOR 1.                                    10/16/87
   61  005800 01  OPEN-FBA.                                                                  10/16/87
   62  005900     05  FILLER                  PIC X(75).                                      10/16/87
   63  006000     05  RECS-IN-DB              PIC 9(09) COMP-4.                               10/16/87
   64  006100     05  FILLER                  PIC X(45).                                      10/16/87
   65  006200 01  MAJ-MIN.                                                                   10/16/87
   66  006300     05  MAJ                     PIC X(2).                                       10/16/87
   67  006400     05  MIN                     PIC X(2).                                       10/16/87
5728CB1 R01 M02 881028                COBOL SOURCE LISTING              ICFLIB/CSDBAT        03/01/89 13:44:14       Page     4
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
      006500/                                                                             10/16/87
```

Figure 10-8 (Part 2 of 6). Source Program Example—CSDBAT (User-Defined Formats)

```
 68  006600 PROCEDURE DIVISION.                                                    10/16/87
     006700 DECLARATIVES.                                                          10/16/87
     006800 ERR-SECTION SECTION.                                                   10/16/87
     006900* 2                                                                    10/16/87
     007000     USE AFTER STANDARD ERROR PROCEDURE ON SRCICF.                      10/16/87
     007100 SRCICF-EXCEPTION.                                                      10/16/87
     007200*                                                                       10/16/87
     007300* CHECK THE MAJOR/MINOR CODES AND TAKE APPROPRIATE ACTION               10/16/87
     007400*                                                                       10/16/87
     007500* RECOVERABLE SESSION ERROR.  CLOSE ICF FILE.                           10/16/87
 69  007600     IF MAJ = "03" OR MAJ = "83"                                       02/21/89
 70  007700        MOVE "PROGRAM STARTED AGAIN DUE TO SESSION ERROR"              10/16/87
     007800        TO PRINTREC                                                    10/16/87
 71  007900        WRITE PRINTREC                                                 10/16/87
 72  008000        MOVE "1" TO ERR-SW                                             10/16/87
 73  008100        GO TO EXIT-DECLARATIVES.                                       10/16/87
     008200*                                                                       10/16/87
     008300*************************************************************************02/21/89
     008400* WHEN THERE IS A PERMANENT SESSION ERROR DETECTED, THE MAJOR-     *    02/21/89
     008500* MINOR CODE IS PLACED INTO A DATA BASE FILE AND THE FILE IS       *    02/21/89
     008600* PRINTED IN HEX USING COPYFILE.                                   *    02/21/89
     008700*************************************************************************02/21/89
     008800*                                                                       10/16/87
     008900 GETFBA.                                                                10/16/87
 74  009000     OPEN OUTPUT DFILE.                                                10/16/87
 75  009100     MOVE MAJ-MIN TO DUMP-MAJ-MIN.                                     10/16/87
 76  009200     MOVE ICFREC TO DUMP-RECORD.                                       10/16/87
 77  009300     WRITE DUMPREC.                                                    10/16/87
 78  009400     CLOSE DFILE.                                                      10/16/87
 79  009500     MOVE 31 TO LEN.                                                   10/16/87
 80  009600     CALL "QCAEXEC" USING CMD2 LEN.                                    10/16/87
 81  009700     MOVE "PROGRAM TERMINATED DUE TO ERROR IN ICF FILE"                10/16/87
     009800        TO PRINTREC.                                                   10/16/87
 82  009900     WRITE PRINTREC.                                                   10/16/87
 83  010000     STOP RUN.                                                         10/16/87
     010100*                                                                      10/16/87
     010200 EXIT-DECLARATIVES.                                                     10/16/87
     010300     EXIT.                                                             02/28/89
     010400*                                                                      10/16/87
 84  010500 END DECLARATIVES.                                                     10/16/87
5728CB1 R01 M02 881028              COBOL SOURCE LISTING          ICFLIB/CSDBAT          03/01/89 13:44:14       Page    5
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
     010600/                                                                       10/16/87
     010700 START-PROGRAM  SECTION.                                                10/16/87
     010800*                                                                       10/16/87
     010900 START-PROGRAM-PARAGRAPH.                                               10/16/87
     011000* 3                                                                    10/16/87
 85  011100     OPEN INPUT  DBFILE                                                10/16/87
     011200          I-O    SRCICF                                                10/16/87
     011300          OUTPUT QPRINT.                                                10/16/87
     011400*                                                                       10/16/87
     011500*************************************************************************02/21/89
     011600* THE FOLLOWING TEST IS TO ATTEMPT RECOVERY IF AN ERROR OCCURS     *    02/21/89
     011700* WHEN OPENING THE ICF FILE.                                       *    02/21/89
     011800*************************************************************************02/21/89
     011900* 4                                                                    10/16/87
 86  012000     IF ERR-SW = "1"                                                   10/16/87
 87  012100         THEN IF OPEN-COUNT IS = 9                                     10/16/87
 88  012200              THEN GO TO ABNORMAL-TERMINATION                          10/16/87
     012300              ELSE                                                     10/16/87
 89  012400              ADD 1 TO OPEN-COUNT                                      10/16/87
     012500              PERFORM ERROR-RECOVERY-RTN                               10/16/87
 90  012600              GO TO START-PROGRAM-PARAGRAPH                            10/16/87
     012700         ELSE                                                          10/16/87
```

Figure 10-8 (Part 3 of 6). Source Program Example—CSDBAT (User-Defined Formats)

```
 92  012800         MOVE 0 TO OPEN-COUNT.                                          10/16/87
     012900*                                                                       10/16/87
     013000*************************************************************************02/21/89
     013100* EVOKE THE PROGRAM "CTDBATCL" ON THE REMOTE SYSTEM IN LIBRARY     *    02/21/89
     013200* ICFLIB. INDICATOR IN50 IS THE EVOKE KEYWORD.                     *    02/21/89
     013300*************************************************************************02/21/89
     013400* 5                                                                     10/16/87
 93  013500     MOVE "CTDBATCL" TO PGMID OF EVOKPGM-O.                            10/16/87
 94  013600     MOVE "ICFLIB" TO LIB  OF EVOKPGM-O.                               10/16/87
 95  013700     MOVE INDON TO CMNF-INDIC(50).                                     10/16/87
 96  013800     WRITE ICFREC FORMAT IS "EVOKPGM"                                  10/16/87
     013900        INDICATORS ARE CMNF-INDIC-AREA.                                10/16/87
 97  014000     MOVE INDOFF TO CMNF-INDIC(50)                                     10/16/87
 98  014100     IF ERR-SW = "1"                                                   10/16/87
 99  014200         PERFORM ERROR-RECOVERY-RTN                                    10/16/87
100  014300         GO TO START-PROGRAM-PARAGRAPH.                                10/16/87
     014400*                                                                      10/16/87
     014500*************************************************************************02/21/89
     014600* WHEN THE EVOKE OPERATION IS SUCCESSFUL, A RECORD FROM THE DATA-  *    02/21/89
     014700* BASE FILE IS READ AND THEN SENT TO THE TARGET SYSTEM. THIS WILL  *    02/21/89
     014800* BE REPEATED UNTIL THE END OF FILE IS REACHED ON THE DATABASE     *    02/21/89
     014900* FILE. AT END OF FILE, THE PROGRAM DEVICE IS INVITED AND THE READ *    02/21/89
     015000* OPERATION IS ISSUED TO GET THE DATA FROM THE REMOTE SYSTEM.      *    02/21/89
     015100*************************************************************************02/21/89
     015200*                                                                      10/16/87
     015300 SEND-DATA.                                                          10/16/87
     015400* 6                                                                    10/16/87
101  015500     READ DBFILE                                                       10/16/87
     015600         AT END                                                        10/16/87
102  015700         PERFORM INVITE-TO-SEND.                                       10/16/87
103  015800     IF EOF-DBFILE-SW NOT = "1"                                        10/16/87
104  015900         MOVE DBREC-DATA TO SNDFLD OF SNDDATA-O                        10/16/87
105  016000         PERFORM WRITE-SRCICF-RTN                                      10/16/87
5728CB1 R01 M02 881028              COBOL SOURCE LISTING          ICFLIB/CSDBAT      03/01/89 13:44:14      Page     6
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
106  016100         GO TO SEND-DATA.                                              10/16/87
     016200*                                                                      10/16/87
     016300*************************************************************************02/21/89
     016400* A READ OPERATION IS ISSUED TO PROGRAM DEVICE TO CONTINUE         *    02/21/89
     016500* RECEIVING DATA FROM THE REMOTE SYSTEM UNTIL THE RCVDETACH INDI-  *    02/21/89
     016600* CATOR IS SET. EACH RECORD RECEIVED WILL BE PRINTED ON THE PRINT  *    02/21/89
     016700* FILE.                                                            *    02/21/89
     016800*************************************************************************02/21/89
     016900*                                                                      10/16/87
     017000 RECEIVE-DATA.                                                       10/16/87
     017100* 7                                                                    10/16/87
107  017200     READ SRCICF INDICATORS ARE CMNF-INDIC-AREA.                       10/16/87
108  017300     IF ERR-SW = "1"                                                   10/16/87
109  017400         PERFORM ERROR-RECOVERY-RTN                                    10/16/87
110  017500         GO TO START-PROGRAM-PARAGRAPH.                                10/16/87
111  017600     IF MAJ NOT = "03"                                                 10/16/87
112  017700        MOVE ICFREC TO PRINTREC                                        10/16/87
113  017800        WRITE PRINTREC.                                                10/16/87
114  017900     IF CMNF-INDIC(35) NOT = INDON                                     10/16/87
115  018000         GO TO RECEIVE-DATA.                                           10/16/87
     018100*                                                                      10/16/87
     018200*************************************************************************02/21/89
     018300* WHEN PROCESSING IS COMPLETED, END OF JOB MESSAGE IS PRINTED.     *    02/21/89
     018400* FILES ARE CLOSED AND THE SESSION IS RELEASED.                    *    02/21/89
     018500*************************************************************************02/21/89
     018600*                                                                      10/16/87
     018700* 8                                                                    10/16/87
116  018800     MOVE "CSDBAT HAS COMPLETED NORMALLY" TO PRINTREC.                 10/16/87
117  018900     WRITE PRINTREC.                                                   10/16/87
```

Figure 10-8 (Part 4 of 6). Source Program Example—CSDBAT (User-Defined Formats)

```
 118  019000     CLOSE DBFILE                                                                  10/16/87
      019100           SRCICF                                                                  10/16/87
      019200           QPRINT.                                                                 10/16/87
 119  019300     STOP RUN.                                                                     10/16/87
      019400*                                                                                  10/16/87
      019500*************************************************************************         02/21/89
      019600*   THIS SUBROUTINE SENDS DATA TO THE REMOTE SYSTEM                     *         02/21/89
      019700*************************************************************************         02/21/89
      019800 WRITE-SRCICF-RTN.                                                                10/16/87
      019900* 9                                                                              10/16/87
 120  020000     WRITE ICFREC FORMAT IS "SNDDATA"                                            10/16/87
      020100        INDICATORS ARE CMNF-INDIC-AREA.                                          10/16/87
 121  020200     IF ERR-SW = "1"                                                             10/16/87
 122  020300         PERFORM ERROR-RECOVERY-RTN                                              10/16/87
 123  020400         GO TO START-PROGRAM-PARAGRAPH.                                          10/16/87
      020500*                                                                                10/16/87
      020600 ERROR-RECOVERY-RTN.                                                              10/16/87
      020700* 10                                                                             10/16/87
 124  020800     CLOSE SRCICF                                                                10/16/87
      020900           DBFILE                                                                10/16/87
      021000           QPRINT.                                                               10/16/87
 125  021100     MOVE "0" TO ERR-SW.                                                         10/16/87
      021200*                                                                                10/16/87
      021300*************************************************************************         02/21/89
      021400*   WHEN AN ERROR OCCURS ON AN ICF SESSION, INFORMATION ABOUT THE  *             02/21/89
      021500*   ERROR IS PRINTED.                                              *             02/21/89
5728CB1 R01 M02 881028              COBOL SOURCE LISTING            ICFLIB/CSDBAT          03/01/89 13:44:14       Page     7
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
      021600*************************************************************************         02/21/89
      021700 ABNORMAL-TERMINATION.                                                           10/16/87
      021800* 11                                                                             10/16/87
 126  021900     MOVE "CSDBAT HAS COMPLETED ABNORMALLY"                                      10/16/87
      022000        TO PRINTREC.                                                            10/16/87
 127  022100     WRITE PRINTREC.                                                             10/16/87
 128  022200     STOP RUN.                                                                   10/16/87
      022300*                                                                                10/16/87
      022400*************************************************************************         02/21/89
      022500* WHEN END OF FILE IS DETECTED, AN INVITE OPERATION IS ISSUED TO   *             02/21/89
      022600* NOTIFY THE TARGET THAT IT CAN START SENDING DATA.                *             02/21/89
      022700*************************************************************************         02/21/89
      022800 INVITE-TO-SEND.                                                                 10/16/87
      022900* 12                                                                             10/16/87
 129  023000     MOVE INDON TO CMNF-INDIC(45).                                               10/16/87
 130  023100     MOVE "1" TO EOF-DBFILE-SW.                                                  10/16/87
 131  023200     WRITE ICFREC FORMAT IS "INVITE"                                             10/16/87
      023300        INDICATORS ARE CMNF-INDIC-AREA.                                         10/16/87
 132  023400     IF ERR-SW = "1"                                                             10/16/87
 133  023500         PERFORM ERROR-RECOVERY-RTN                                              10/16/87
 134  023600         GO TO START-PROGRAM-PARAGRAPH.                                          10/16/87
                      * * * * *  E N D  O F  S O U R C E  * * * * *
5728CB1 R01 M02 881028                COBOL MESSAGES                ICFLIB/CSDBAT          03/01/89 13:44:14       Page     8
 STMT
*   19  MSGID: LBL0650  SEVERITY: 00  SEQNBR:  002700
       Message . . . . :   Blocking/Deblocking for file 'DBFILE' will
         be performed by compiler-generated code.
*   26  MSGID: LBL0600  SEVERITY: 10  SEQNBR:  003400
       Message . . . . :   No INPUT fields found for format EVOKPGM.
*   26  MSGID: LBL0600  SEVERITY: 10  SEQNBR:  003400
       Message . . . . :   No INPUT fields found for format ENDREC.
*   26  MSGID: LBL0600  SEVERITY: 10  SEQNBR:  003400
       Message . . . . :   No OUTPUT fields found for format ENDREC.
*   26  MSGID: LBL0600  SEVERITY: 10  SEQNBR:  003400
       Message . . . . :   No INPUT fields found for format INVITE.
*   26  MSGID: LBL0600  SEVERITY: 10  SEQNBR:  003400
       Message . . . . :   No OUTPUT fields found for format INVITE.
```

Figure 10-8 (Part 5 of 6). Source Program Example—CSDBAT (User-Defined Formats)

```
*   39  MSGID: LBL0650  SEVERITY: 00  SEQNBR:  003500
        Message . . . . :   Blocking/Deblocking for file 'DFILE' will be
          performed by compiler-generated code.
                                        MESSAGE SUMMARY
 TOTAL     INFO(0-4)     WARNING(5-19)     ERROR(20-29)     SEVERE(30-39)     TERMINAL(40-99)
     7           2                 5                0                 0                   0
                    * * * * *  E N D  O F  C O B O L  M E S S A G E S  * * * * *
236 source records read
37 copy records read
1 copy members processed
0 sequence errors
10 was the highest severity message issued
 LBL0901 00  Program CSDBAT created in library ICFLIB.
                         * * * * *  E N D  O F  C O M P I L A T I O N  * * * * *
```

Figure 10-8 (Part 6 of 6). Source Program Example—CSDBAT (User-Defined Formats)

```
5728CB1 R01 M02 881028              IBM AS/400  COBOL/400          ICFLIB/CSFBAT          03/01/89 13:45:26       Page    1
Program name . . . . . . . . . . . . : CSFBAT in ICFLIB
Source file  . . . . . . . . . . . . : QICFPUB in ICFLIB     Member - CSFBAT     02/28/89 13:55:59
Compiler option  . . . . . . . . . . : *NONE
Code generation option . . . . . . . : *NONE
Code generation severity level . . . : 29
Print file . . . . . . . . . . . . . : QSYSPRT in *LIBL
FIPS flagging option . . . . . . . . : *NOFIPS *NOSEG *NODEB *NOOBSOLETE
Text not available for message LBX6039 file QLBLMSG.
Flagging level . . . . . . . . . . . : 0
Replace existing program . . . . . . : *YES
User profile . . . . . . . . . . . . : *USER
Authority  . . . . . . . . . . . . . : *CHANGE
Text . . . . . . . . . . . . . . . . : cobol batch file transfer using $$FORMAT (source)
Compiler . . . . . . . . . . . . . . : IBM AS/400  COBOL/400
5728CB1 R01 M02 881028              COBOL SOURCE LISTING          ICFLIB/CSFBAT          03/01/89 13:45:26       Page    2
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
    1  000100 IDENTIFICATION DIVISION.                                                                  02/21/89
    2  000200 PROGRAM-ID.                CSFBAT.                                                        02/21/89
       000300*************************************************************************                02/21/89
       000400*  THIS IS A BATCH FILE TRANSFER PROGRAM THAT READS A SEQUENTIAL   *                       02/21/89
       000500*  FILE AND SENDS THE RECORDS TO THE REMOTE SYSTEM UNTIL THE END   *                       02/21/89
       000600*  OF FILE IS REACHED. AT THIS TIME, THE PROGRAM STOPS SENDING     *                       02/21/89
       000700*  AND STARTS RECEIVING RECORDS FROM THE REMOTE SYSTEM UNTIL A     *                       02/21/89
       000800*  A DETACH INDICATION IS RECEIVED.                                *                       02/21/89
       000900*************************************************************************                02/21/89
    3  001000 ENVIRONMENT DIVISION.                                                                     10/16/87
    4  001100 CONFIGURATION SECTION.                                                                    10/16/87
    5  001200 SOURCE-COMPUTER.           IBM-AS400.                                                     02/21/89
    6  001300 OBJECT-COMPUTER.           IBM-AS400.                                                     02/21/89
    7  001400 SPECIAL-NAMES.             I-O-FEEDBACK IS FEEDBACK-AREA                                  10/16/87
    8  001500                            OPEN-FEEDBACK IS OPEN-FBA.                                     10/16/87
    9  001600 INPUT-OUTPUT SECTION.                                                                     10/16/87
   10  001700 FILE-CONTROL.                                                                             10/16/87
       001800* 1                                                                                        10/16/87
   11  001900     SELECT DBFILE ASSIGN TO DATABASE-DBFILE.                                              10/16/87
   12  002000     SELECT SRCICF ASSIGN TO WORKSTATION-SRCICF-SI                                         10/16/87
   13  002100         ORGANIZATION IS TRANSACTION                                                       10/16/87
   14  002200         FILE STATUS IS STATUS-IND MAJ-MIN.                                                10/16/87
   15  002300     SELECT DFILE ASSIGN TO DATABASE-HEXDUMP.                                              10/16/87
   16  002400     SELECT QPRINT ASSIGN TO PRINTER-QSYSPRT.                                              10/16/87
   17  002500 DATA DIVISION.                                                                            10/16/87
   18  002600 FILE SECTION.                                                                             10/16/87
   19  002700 FD  DBFILE                                                                                10/16/87
   20  002800     LABEL RECORDS ARE STANDARD.                                                           10/16/87
   21  002900 01  DBREC.                                                                               10/16/87
   22  003000     05  DBREC-DATA                  PIC X(80).                                           10/16/87
   23  003100 FD  SRCICF                                                                                10/16/87
   24  003200     LABEL RECORDS ARE STANDARD.                                                           10/16/87
   25  003300 01  ICFREC.                                                                              10/16/87
   26  003400     05  EVOKPGM-O.                                                                       10/16/87
   27  003500         10  PGMID                   PIC X(8).                                            10/16/87
   28  003600         10  PASSWD                  PIC X(8).                                            11/16/88
   29  003700         10  USERID                  PIC X(8).                                            11/16/88
   30  003800         10  LIB                     PIC X(8).                                            10/16/87
   31  003900         10  FILLER                  PIC X(52).                                           10/16/87
   32  004000     05  SNDDATA-O REDEFINES EVOKPGM-O.                                                   10/16/87
   33  004100         10  SNDLENGTH               PIC 9(4).                                            10/16/87
   34  004200         10  SNDFIELD                PIC X(80).                                           10/16/87
   35  004300     05  INVITE-O REDEFINES EVOKPGM-O.                                                    02/27/89
   36  004400         10  INVLENGTH               PIC 9(4).                                            10/16/87
   37  004500         10  INVFIELD                PIC X(80).                                           10/16/87
   38  004600 FD  DFILE                                                                                 10/16/87
   39  004700     LABEL RECORDS ARE STANDARD.                                                           10/16/87
   40  004800 01  DUMPREC.                                                                             10/16/87
```

Figure 10-9 (Part 1 of 5). Source Program Example — CSFBAT (System-Supplied Formats)

```
   41  004900     05  DUMP-MAJ-MIN                PIC X(4).                                    10/16/87
   42  005000     05  DUMP-RECORD                 PIC X(400).                                  10/16/87
   43  005100 FD  QPRINT                                                                       10/16/87
   44  005200     LABEL RECORDS ARE OMITTED.                                                   10/16/87
   45  005300 01  PRINTREC            PIC X(132).                                             10/16/87
   46  005400 WORKING-STORAGE SECTION.                                                         10/16/87
   47  005500 77  STATUS-IND                      PIC X(2).                                    10/16/87
5728CB1 R01 M02 881028               COBOL SOURCE LISTING            ICFLIB/CSFBAT          03/01/89 13:45:26        Page     3
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
   48  005600 77  MAJ-MIN-SAV                  PIC X(4).                                    10/16/87
   49  005700 77  EOF-DBFILE-SW                PIC X VALUE "0".                             10/16/87
   50  005800 77  ERR-SW                       PIC X VALUE "0".                             10/16/87
   51  005900 77  OPEN-COUNT                   PIC 9(1) VALUE 0.                            10/16/87
   52  006000 77  LEN                          PIC 9(10)V9(5) COMP.                         10/16/87
       006100                                                                               10/16/87
   53  006200 77  CMD2                         PIC X(31)                                    10/16/87
   54  006300       VALUE "CPYF HEXDUMP *LIST PRTFMT(*HEX)".                                10/16/87
   55  006400 01  OPEN-FBA.                                                                  10/16/87
   56  006500     05  FILLER                   PIC X(75).                                   10/16/87
   57  006600     05  RECS-IN-DB               PIC 9(09) COMP-4.                            10/16/87
   58  006700     05  FILLER                   PIC X(45).                                   10/16/87
   59  006800 01  MAJ-MIN.                                                                   10/16/87
   60  006900     05  MAJ                      PIC X(2).                                    10/16/87
   61  007000     05  MIN                      PIC X(2).                                    10/16/87
5728CB1 R01 M02 881028               COBOL SOURCE LISTING            ICFLIB/CSFBAT          03/01/89 13:45:26        Page     4
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
       007100/                                                                              10/16/87
   62  007200 PROCEDURE DIVISION.                                                           10/16/87
       007300 DECLARATIVES.                                                                 10/16/87
       007400 ERR-SECTION SECTION.                                                          10/16/87
       007500* 2                                                                           10/16/87
       007600     USE AFTER STANDARD ERROR PROCEDURE ON SRCICF.                             10/16/87
       007700 SRCICF-EXCEPTION.                                                             10/16/87
       007800*                                                                             10/16/87
       007900* CHECK THE MAJOR/MINOR CODES AND TAKE APPROPRIATE ACTION                     10/16/87
       008000*                                                                             10/16/87
       008100* RECOVERABLE SESSION ERROR.  CLOSE ICF FILE.                               10/16/87
   63  008200     IF MAJ = "03" OR MAJ = "83"                                              10/16/87
   64  008300        MOVE "PROGRAM STARTED AGAIN DUE TO SESSION ERROR"                     10/16/87
       008400        TO PRINTREC                                                           10/16/87
   65  008500        WRITE PRINTREC                                                        10/16/87
   66  008600        MOVE "1" TO ERR-SW                                                    10/16/87
   67  008700        GO TO EXIT-DECLARATIVES.                                              10/16/87
       008800*                                                                             10/16/87
       008900*************************************************************************  02/21/89
       009000* WHEN THERE IS A PERMANENT SESSION ERROR DETECTED, THE MAJOR-MINOR*          02/21/89
       009100* CODE IS PLACED INTO A DATA BASE FILE AND THE FILE IS PRINTED IN  *          02/21/89
       009200* HEX USING COPYFILE.                                              *          02/21/89
       009300*************************************************************************  02/21/89
       009400*                                                                             10/16/87
       009500 GETFBA.                                                                       10/16/87
   68  009600     OPEN OUTPUT DFILE.                                                       10/16/87
   69  009700     MOVE MAJ-MIN TO DUMP-MAJ-MIN.                                            10/16/87
   70  009800     MOVE ICFREC TO DUMP-RECORD.                                              10/16/87
   71  009900     WRITE DUMPREC.                                                           10/16/87
   72  010000     CLOSE DFILE.                                                             10/16/87
   73  010100     MOVE 31 TO LEN.                                                          10/16/87
   74  010200     CALL "QCAEXEC" USING CMD2 LEN.                                           10/16/87
   75  010300     MOVE "PROGRAM TERMINATED DUE TO ERROR IN ICF FILE"                       10/16/87
       010400        TO PRINTREC.                                                          10/16/87
   76  010500     WRITE PRINTREC.                                                          10/16/87
   77  010600     STOP RUN.                                                                10/16/87
       010700*                                                                             10/16/87
       010800 EXIT-DECLARATIVES.                                                            10/16/87
       010900     EXIT.                                                                    02/28/89
```

Figure 10-9 (Part 2 of 5). Source Program Example — CSFBAT (System-Supplied Formats)

```
     011000*                                                                                10/16/87
  78 011100 END DECLARATIVES.                                                               10/16/87
5728CB1 R01 M02 881028              COBOL SOURCE LISTING            ICFLIB/CSFBAT          03/01/89 13:45:26      Page     5
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
     011200/                                                                                10/16/87
     011300 START-PROGRAM  SECTION.                                                         10/16/87
     011400*                                                                                10/16/87
     011500 START-PROGRAM-PARAGRAPH.                                                        10/16/87
     011600* 3                                                                             10/16/87
  79 011700     OPEN INPUT DBFILE                                                         10/16/87
     011800          I-O   SRCICF                                                        10/16/87
     011900          OUTPUT QPRINT.                                                        10/16/87
     012000*                                                                               10/16/87
     012100*************************************************************************     02/21/89
     012200* THE FOLLOWING TEST IS TO ATTEMPT RECOVERY IF AN ERROR OCCURS     *           02/21/89
     012300* WHEN OPENING THE ICF FILE.                                       *           02/21/89
     012400*************************************************************************     02/21/89
     012500* 4                                                                             10/16/87
  80 012600     IF ERR-SW = "1"                                                           10/16/87
  81 012700         THEN IF OPEN-COUNT IS = 9                                             10/16/87
  82 012800              THEN GO TO ABNORMAL-TERMINATION                                  10/16/87
     012900              ELSE                                                             10/16/87
  83 013000              ADD 1 TO OPEN-COUNT                                              10/16/87
     013100              PERFORM ERROR-RECOVERY-RTN                                       10/16/87
  84 013200              GO TO START-PROGRAM-PARAGRAPH                                    10/16/87
     013300         ELSE                                                                  10/16/87
  86 013400         MOVE 0 TO OPEN-COUNT.                                                 10/16/87
     013500*                                                                              10/16/87
     013600*************************************************************************     02/21/89
     013700* EVOKE THE PROGRAM "CTDBATCL" ON THE REMOTE SYSTEM IN LIBRARY     *           02/21/89
     013800* ICFLIB. INDICATOR IN50 IS THE EVOKE KEYWORD.                     *           02/21/89
     013900*************************************************************************     02/21/89
     014000* 5                                                                             10/16/87
  87 014100     MOVE SPACES TO EVOKPGM-O.                                                 10/16/87
  88 014200     MOVE "CTFBATCL" TO PGMID OF EVOKPGM-O.                                    10/16/87
  89 014300     MOVE "QSECOFR" TO PASSWD OF EVOKPGM-O.                                    11/16/88
  90 014400     MOVE "QSECOFR" TO USERID OF EVOKPGM-O.                                    11/16/88
  91 014500     MOVE "ICFLIB" TO LIB  OF EVOKPGM-O.                                       10/16/87
  92 014600     WRITE ICFREC FORMAT IS "$$EVOKNI".                                        10/16/87
  93 014700     IF ERR-SW = "1"                                                           10/16/87
  94 014800         PERFORM ERROR-RECOVERY-RTN                                            10/16/87
  95 014900         GO TO START-PROGRAM-PARAGRAPH.                                        10/16/87
     015000*                                                                              10/16/87
     015100*************************************************************************     02/21/89
     015200* WHEN THE EVOKE OPERATION IS SUCCESSFUL, A RECORD FROM THE DATA-  *           02/21/89
     015300* BASE FILE IS READ AND THEN SENT TO THE TARGET SYSTEM. THIS WILL  *           02/21/89
     015400* BE REPEATED UNTIL THE END OF FILE IS REACHED ON THE DATABASE     *           02/21/89
     015500* FILE. AT END OF FILE, THE PROGRAM DEVICE IS INVITED AND THE READ *           02/21/89
     015600* OPERATION IS ISSUED TO GET THE DATA FROM THE REMOTE SYSTEM.      *           02/21/89
     015700*************************************************************************     02/21/89
     015800*                                                                              10/16/87
     015900 SEND-DATA.                                                                    10/16/87
     016000* 6                                                                            10/16/87
  96 016100     READ DBFILE                                                               10/16/87
     016200         AT END                                                                10/16/87
  97 016300         PERFORM INVITE-TO-SEND.                                               10/16/87
  98 016400     IF EOF-DBFILE-SW NOT = "1"                                                10/16/87
  99 016500         MOVE DBREC-DATA TO SNDFIELD OF SNDDATA-O                              10/16/87
 100 016600         MOVE +80 TO SNDLENGTH OF SNDDATA-O                                    10/16/87
5728CB1 R01 M02 881028              COBOL SOURCE LISTING            ICFLIB/CSFBAT          03/01/89 13:45:26      Page     6
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
 101 016700         PERFORM WRITE-SRCICF-RTN                                              10/16/87
 102 016800         GO TO SEND-DATA.                                                      10/16/87
     016900*                                                                              10/16/87
```

Figure 10-9 (Part 3 of 5). Source Program Example — CSFBAT (System-Supplied Formats)

```
      017000*****************************************************************                02/21/89
      017100* A READ OPERATION IS ISSUED TO PROGRAM DEVICE TO CONTINUE       *                02/21/89
      017200* RECEIVING DATA FROM THE REMOTE SYSTEM UNTIL THE RCVDETACH      *                02/21/89
      017300* INDICATOR IS SET. EACH RECORD RECEIVED WILL BE PRINTED TO THE  *                02/21/89
      017400* PRINT FILE.                                                    *                02/21/89
      017500*****************************************************************                02/21/89
      017600*                                                                                10/16/87
      017700 RECEIVE-DATA.                                                                    10/16/87
      017800* 7                                                                              10/16/87
  103 017900     READ SRCICF.                                                                10/16/87
  104 018000     IF ERR-SW = "1"                                                             10/16/87
  105 018100         PERFORM ERROR-RECOVERY-RTN                                              10/16/87
  106 018200         GO TO START-PROGRAM-PARAGRAPH.                                          10/16/87
  107 018300     IF MAJ NOT = "03"                                                           10/16/87
  108 018400        MOVE ICFREC TO PRINTREC                                                  10/16/87
  109 018500        WRITE PRINTREC.                                                          10/16/87
  110 018600     IF MIN NOT = "08"                                                           10/16/87
  111 018700         GO TO RECEIVE-DATA.                                                     10/16/87
      018800*                                                                               10/16/87
      018900*****************************************************************                02/21/89
      019000* WHEN PROCESSING IS COMPLETED, END OF JOB MESSAGE IS PRINTED.   *               02/21/89
      019100* FILES ARE CLOSED AND THE SESSION IS ENDED.                     *               02/21/89
      019200*****************************************************************                02/21/89
      019300*                                                                                10/16/87
      019400* 8                                                                              10/16/87
  112 019500     MOVE "CSFBAT HAS COMPLETED NORMALLY" TO PRINTREC.                           10/16/87
  113 019600     WRITE PRINTREC.                                                             10/16/87
  114 019700     CLOSE DBFILE                                                                10/16/87
      019800           SRCICF                                                                10/16/87
      019900           QPRINT.                                                               10/16/87
  115 020000     STOP RUN.                                                                   10/16/87
      020100*                                                                               10/16/87
      020200*****************************************************************                02/21/89
      020300*   THIS SUBROUTINE SENDS DATA TO THE REMOTE SYSTEM              *               02/21/89
      020400*****************************************************************                02/21/89
      020500 WRITE-SRCICF-RTN.                                                                10/16/87
      020600* 9                                                                              10/16/87
  116 020700     WRITE ICFREC FORMAT IS "$$SENDNI".                                          10/16/87
  117 020800     IF ERR-SW = "1"                                                             10/16/87
  118 020900         PERFORM ERROR-RECOVERY-RTN                                              10/16/87
  119 021000         GO TO START-PROGRAM-PARAGRAPH.                                          10/16/87
      021100*                                                                               10/16/87
      021200 ERROR-RECOVERY-RTN.                                                              10/16/87
      021300* 10                                                                             10/16/87
  120 021400     CLOSE SRCICF                                                                10/16/87
      021500           DBFILE                                                                10/16/87
      021600           QPRINT.                                                               10/16/87
  121 021700     MOVE "0" TO ERR-SW.                                                         10/16/87
      021800*                                                                               10/16/87
      021900*****************************************************************                02/21/89
      022000*   WHEN AN ERROR OCCURS ON AN ICF SESSION, INFORMATION ABOUT    *               02/21/89
      022100*   THE ERROR IS PRINTED.                                        *               02/21/89
5728CB1 R01 M02 881028              COBOL SOURCE LISTING          ICFLIB/CSFBAT        03/01/89 13:45:26       Page     7
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
      022200*****************************************************************                02/21/89
      022300 ABNORMAL-TERMINATION.                                                            10/16/87
      022400* 11                                                                             10/16/87
  122 022500     MOVE "CSFBAT HAS COMPLETED ABNORMALLY"                                      10/16/87
      022600        TO PRINTREC.                                                             10/16/87
  123 022700     WRITE PRINTREC.                                                             10/16/87
  124 022800     STOP RUN.                                                                   10/16/87
      022900*                                                                               10/16/87
```

Figure 10-9 (Part 4 of 5). Source Program Example — CSFBAT (System-Supplied Formats)

```
      023000*****************************************************************                02/21/89
      023100* WHEN END OF FILE IS DETECTED, AN INVITE OPERATION IS ISSUED TO   *              02/21/89
      023200* NOTIFY THE TARGET THAT IT CAN START SENDING DATA.                *              02/21/89
      023300*****************************************************************                02/21/89
      023400 INVITE-TO-SEND.                                                                  10/16/87
      023500* 12                                                                             10/16/87
  125 023600     MOVE "1" TO EOF-DBFILE-SW.                                                  10/16/87
  126 023700     MOVE +0 TO INVLENGTH OF INVITE-O.                                           10/16/87
  127 023800     WRITE ICFREC FORMAT IS "$$SEND".                                            10/16/87
  128 023900     IF ERR-SW = "1"                                                             10/16/87
  129 024000         PERFORM ERROR-RECOVERY-RTN                                              10/16/87
  130 024100         GO TO START-PROGRAM-PARAGRAPH.                                          10/16/87
                         * * * * *  E N D  O F  S O U R C E  * * * * *
5728CB1 R01 M02 881028                COBOL MESSAGES              ICFLIB/CSFBAT          03/01/89 13:45:26       Page    8
 STMT
*   19  MSGID: LBL0650  SEVERITY: 00  SEQNBR:  002700
      Message . . . . :   Blocking/Deblocking for file 'DBFILE' will
        be performed by compiler-generated code.
*   38  MSGID: LBL0650  SEVERITY: 00  SEQNBR:  004600
      Message . . . . :   Blocking/Deblocking for file 'DFILE' will be
        performed by compiler-generated code.
                                    MESSAGE SUMMARY
 TOTAL    INFO(0-4)    WARNING(5-19)    ERROR(20-29)    SEVERE(30-39)    TERMINAL(40-99)
    2          2              0               0               0                0
               * * * * *  E N D  O F  C O B O L  M E S S A G E S  * * * * *
241 source records read
0 copy records read
0 copy members processed
0 sequence errors
0 was the highest severity message issued
 LBL0901 00  Program CSFBAT created in library ICFLIB.
                     * * * * *  E N D  O F  C O M P I L A T I O N  * * * * *
```

Figure 10-9 (Part 5 of 5). Source Program Example — CSFBAT (System-Supplied Formats)

Target Program Batch Transfer (Example I)

The following describes a COBOL target program batch transfer program.

Program Files: The COBOL batch transfer target program uses the following files:

TGTICF An ICF file used to send records to and receive records from the source program.

DBFILE A database file that contains the records to be sent to the source program.

QPRINT A printer file used to print the records received from the source program.

DDS Source: The DDS source used in the ICF file is illustrated in the following example. The other files (DBFILE and QPRINT) are program-described and therefore requires no DDS.

```
A*************************************************************
A*                                                           *
A*                          ICF FILE                         *
A*              USED IN BATCH DATA TRANSFER PROGRAM          *
A*                                                           *
A*************************************************************
A*
A*  FILE LEVEL INDICATORS:
A*
A                                     INDARA
A*
A                                     RCVTRNRND(15 'END OF DATA')
A*
A  30                                 DETACH
A*
A                                     INDTXT(30 '30->DETACH TARG-
A                                     GET PROGRAM.')
A*
A                                     RCVDETACH(35 'RECEIVED    -
A                                     DETACHED.')
A*
A*
A*************************************************************
A*                       ICF RECORD FORMATS                  *
A*************************************************************
          R RCVDATA
            RCVFLD        80A
          R SNDDATA
            SNDFLD        80A
          R EVOKPGM
A  50                                 EVOKE(&LIB/&PGMID)
A  50                                 SECURITY(2 'PASSWRD' +
                                              3 'USERID')
A           PGMID         10A  P
A           LIB           10A  P
A         R ENDREC
A         R INVITE
A  45                                 INVITE
```

Figure 10-10. DDS Source for ICF File Used in Target Program Batch Transfer

ICF File Creation and Program Device Entry Definition: The command needed to create the ICF file is:

```
CRTICFF  FILE(ICFLIB/TGTICF)  SRCFILE(ICFLIB/QICFPUB)  SRCMBR(TGTICF)
   ACQPGMDEV(PGMDEVB)  TEXT('TARGET ICF FILE FOR BATCH DATA TRANSFER')
```

The command needed to define the program device entry is:

```
ADDICFDEVE FILE(ICFLIB/TGTICF) PGMDEV(PGMDEVB) RMTLOCNAME(*REQUESTER)
```

This example acquires all sessions at the beginning of the program. For performance considerations, you may not want to acquire sessions until they are actually needed in the program.

Program Explanation: The following describes the structure of the program examples illustrated in Figure 10-11 on page 10-25 and Figure 10-12 on page 10-29. The ICF file used in the first example is defined by the user, and uses externally described data formats (DDS). The second example uses the same file, but uses program-described data and system-supplied formats. The reference numbers in the explanation below correspond to the numbers in the following program examples.

Although the basic structure of the two examples provided is the same, there are differences because of the use of user-defined formats and system-supplied formats.

Differences between the first and second example are documented as notes in each of the descriptions.

1 This section identifies the files used in the program. TGTICF is the ICF file used to send records to the source program.

This section also contains declarations of I/O variables, work areas, and constants needed. MAJ-MIN contains the major/minor return code from the I/O feedback area.

Note: In the program using system-supplied formats, the input records for TGTICF are explicitly coded in the program since TGTICF is now treated as a program-described file. The system-supplied file, QICDMF, can be used instead of TGTICF. Using the system-supplied file can be done by specifying QICDMF in the file specification, or by using an OVRICFF command to change the file name from TGTICF to QICDMF. The OVRICFF command can also be used to change the ACQPGMDEV parameter of the file.

2 The program defines the error handling for the program. If an error occurs, the program writes the following message to the print file:

```
CTDBAT HAS COMPLETED ABNORMALLY
```

The session is released.

Note: In the program using system-supplied formats, the program name is CTFBAT.

3 The program opens the files that are going to be used as follows:

DBFILE — The database file used as input for transmitting to the source program

QPRINT — The printer file used as output for records received

TGTICF — The ICF file used to receive data from and send data to the source program

Because the ICF file was created using the ACQPGMDEV parameter, the program device is automatically acquired when the file is opened. Therefore, no acquire operation is coded in the program.

4 Data is read from the program device (TGTICF file).

If an error occurs on the read (major return code greater than '03'), control passes to 2. Otherwise if data was received (major return code not = '03'), then the data is written to the printer file (QPRINT).

5 Data records are read until the change-direction indicator is received from the source program. When change direction is received, indicator 15 is set on, as defined by the RCVTRNRND keyword in the DDS for the ICF file, and control is passed to 6.

Note: In the program using system-supplied formats, the minor return code of '00' is checked to verify whether change direction is received.

6 The database file is read and the records sent to the source program until the end of the database file. At end-of-file, the program passes control to 7.

If it is not the last record, then the data is written to the program device using the format SNDDATA, and the next database record is is read. If an error occurs on the write operation, the program goes to 2 and a message is printed.

Note: In the program using system-supplied formats, the $$SENDNI format is used instead of the user-defined SNDDATA format.

7 This routine issues a detach request to the program device using format ENDREC. Indicator 30 is associated with the DETACH keyword. If an error occurs, the program goes to 2 and a message is printed.

Note: In the program using system-supplied formats, the $$SENDET format is used instead of the user-defined ENDREC format.

8 After the detach request has been sent, the following message is written to the print file:

```
CTDBAT HAS COMPLETED NORMALLY
```

The files are closed. The program device is automatically released as a result of the close operation, and the program ends.

Note: In the program using system-supplied formats, the program name is CTFBAT.

```
5728CB1 R01 M02 881028              IBM AS/400  COBOL/400           ICFLIB/CTDBAT          03/01/89 13:44:52       Page    1
Program name . . . . . . . . . . . . : CTDBAT in ICFLIB
Source file  . . . . . . . . . . . . : QICFPUB in ICFLIB     Member - CTDBAT     02/21/89 17:51:20
Compiler option  . . . . . . . . . . : *NONE
Code generation option . . . . . . . : *NONE
Code generation severity level . . . : 29
Print file . . . . . . . . . . . . . : QSYSPRT in *LIBL
FIPS flagging option . . . . . . . . : *NOFIPS *NOSEG *NODEB *NOOBSOLETE
Text not available for message LBX6039 file QLBLMSG.
Flagging level . . . . . . . . . . . : 0
Replace existing program . . . . . . : *YES
User profile . . . . . . . . . . . . : *USER
Authority  . . . . . . . . . . . . . : *CHANGE
Text . . . . . . . . . . . . . . . . : cobol batch file transfer using dds (target)
Compiler . . . . . . . . . . . . . . : IBM AS/400  COBOL/400
5728CB1 R01 M02 881028              COBOL SOURCE LISTING           ICFLIB/CTDBAT          03/01/89 13:44:52       Page    2
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
    1  000100 IDENTIFICATION DIVISION.                                                              02/21/89
    2  000200 PROGRAM-ID.              CTDBAT.                                                      02/21/89
       000300*****************************************************************                      02/21/89
       000400*   THIS TARGET PROGRAM IS EVOKED BY THE SOURCE PROGRAM AND       *                      02/21/89
       000500*   RECEIVES RECORDS FROM IT. WHEN THE SOURCE PROGRAM IS DONE     *                      02/21/89
       000600*   SENDING DATA, THIS PROGRAM SENDS ITS OWN RECORDS UNTIL IT IS  *                      02/21/89
       000700*   DONE.  WHEN IT IS DONE, IT SENDS A DETACH REQUEST TO THE SOURCE*                     02/21/89
       000800*   PROGRAM AND ENDS ITS SESSION AND JOB.                         *                      02/21/89
       000900*****************************************************************                      02/21/89
    3  001000 ENVIRONMENT DIVISION.                                                                 10/16/87
    4  001100 CONFIGURATION SECTION.                                                                10/16/87
    5  001200 SOURCE-COMPUTER.         IBM-AS400.                                                   02/21/89
    6  001300 OBJECT-COMPUTER.         IBM-AS400.                                                   02/21/89
    7  001400 SPECIAL-NAMES.           I-O-FEEDBACK IS FEEDBACK-AREA                                10/16/87
    8  001500                          OPEN-FEEDBACK IS OPEN-FBA.                                   10/16/87
    9  001600 INPUT-OUTPUT SECTION.                                                                 10/16/87
       001700* 1                                                                                    10/16/87
   10  001800 FILE-CONTROL.                                                                         10/16/87
   11  001900     SELECT DBFILE ASSIGN TO DATABASE-DBFILE.                                          10/16/87
   12  002000     SELECT TGTICF ASSIGN TO WORKSTATION-TGTICF-SI                                     10/16/87
   13  002100         ORGANIZATION IS TRANSACTION                                                   10/16/87
   14  002200         FILE STATUS IS STATUS-IND MAJ-MIN.                                            10/16/87
   15  002300     SELECT DFILE ASSIGN TO DATABASE-HEXDUMP.                                          10/16/87
   16  002400     SELECT QPRINT ASSIGN TO PRINTER-QSYSPRT.                                          10/16/87
   17  002500 DATA DIVISION.                                                                        10/16/87
   18  002600 FILE SECTION.                                                                         10/16/87
   19  002700 FD  DBFILE                                                                            10/16/87
   20  002800     LABEL RECORDS ARE STANDARD.                                                       10/16/87
   21  002900 01  DBREC.                                                                           10/16/87
   22  003000     05  DBREC-DATA                  PIC X(80).                                       10/16/87
   23  003100 FD  TGTICF                                                                            10/16/87
   24  003200     LABEL RECORDS ARE STANDARD.                                                       10/16/87
   25  003300 01  ICFREC.                                                                          10/16/87
   26  003400     COPY DDS-ALL-FORMATS OF TGTICF.                                                   10/16/87
   27 +000001       05  TGTICF-RECORD PIC X(80).                                 <-ALL-FMTS
      +000002*  INPUT FORMAT:RCVDATA    FROM FILE TGTICF     OF LIBRARY ICFLIB   <-ALL-FMTS
      +000003*                                                                   <-ALL-FMTS
   28 +000004       05  RCVDATA-I     REDEFINES TGTICF-RECORD.                  <-ALL-FMTS
   29 +000005           06 RCVFLD           PIC X(80).                          <-ALL-FMTS
      +000006* OUTPUT FORMAT:RCVDATA    FROM FILE TGTICF     OF LIBRARY ICFLIB   <-ALL-FMTS
      +000007*                                                                   <-ALL-FMTS
   30 +000008       05  RCVDATA-O     REDEFINES TGTICF-RECORD.                  <-ALL-FMTS
   31 +000009           06 RCVFLD           PIC X(80).                          <-ALL-FMTS
```

Figure 10-11 (Part 1 of 4). Target Program Example — CTDBAT (User-Defined Formats)

```
     +000010*  INPUT FORMAT:SNDDATA    FROM FILE TGTICF     OF LIBRARY ICFLIB           <-ALL-FMTS
     +000011*                                                                          <-ALL-FMTS
  32 +000012       05  SNDDATA-I     REDEFINES TGTICF-RECORD.                        <-ALL-FMTS
  33 +000013           06 SNDFLD            PIC X(80).                               <-ALL-FMTS
     +000014* OUTPUT FORMAT:SNDDATA    FROM FILE TGTICF     OF LIBRARY ICFLIB           <-ALL-FMTS
     +000015*                                                                          <-ALL-FMTS
  34 +000016       05  SNDDATA-O     REDEFINES TGTICF-RECORD.                        <-ALL-FMTS
  35 +000017           06 SNDFLD            PIC X(80).                               <-ALL-FMTS
     +000018*  INPUT FORMAT:EVOKPGM    FROM FILE TGTICF     OF LIBRARY ICFLIB           <-ALL-FMTS
     +000019*                                                                          <-ALL-FMTS
     +000020*      05  EVOKPGM-I     REDEFINES TGTICF-RECORD.                        <-ALL-FMTS
     +000021* OUTPUT FORMAT:EVOKPGM    FROM FILE TGTICF     OF LIBRARY ICFLIB           <-ALL-FMTS
5728CB1 R01 M02 881028               COBOL SOURCE LISTING            ICFLIB/CTDBAT          03/01/89 13:44:52       Page     3
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
     +000022*                                                                          <-ALL-FMTS
  36 +000023       05  EVOKPGM-O     REDEFINES TGTICF-RECORD.                        <-ALL-FMTS
  37 +000024           06 PGMID             PIC X(10).                               <-ALL-FMTS
  38 +000025           06 LIB               PIC X(10).                               <-ALL-FMTS
     +000026*  INPUT FORMAT:ENDREC     FROM FILE TGTICF     OF LIBRARY ICFLIB           <-ALL-FMTS
     +000027*                                                                          <-ALL-FMTS
     +000028*      05  ENDREC-I      REDEFINES TGTICF-RECORD.                        <-ALL-FMTS
     +000029* OUTPUT FORMAT:ENDREC     FROM FILE TGTICF     OF LIBRARY ICFLIB           <-ALL-FMTS
     +000030*                                                                          <-ALL-FMTS
     +000031*      05  ENDREC-O      REDEFINES TGTICF-RECORD.                        <-ALL-FMTS
     +000032*  INPUT FORMAT:INVITE     FROM FILE TGTICF     OF LIBRARY ICFLIB           <-ALL-FMTS
     +000033*                                                                          <-ALL-FMTS
     +000034*      05  INVITE-I      REDEFINES TGTICF-RECORD.                        <-ALL-FMTS
     +000035* OUTPUT FORMAT:INVITE     FROM FILE TGTICF     OF LIBRARY ICFLIB           <-ALL-FMTS
     +000036*                                                                          <-ALL-FMTS
     +000037*      05  INVITE-O      REDEFINES TGTICF-RECORD.                        <-ALL-FMTS
  39  003500 FD  DFILE                                                                        10/16/87
  40  003600     LABEL RECORDS ARE STANDARD.                                                  10/16/87
  41  003700 01  DUMPREC.                                                                    10/16/87
  42  003800     05  DUMP-MAJ-MIN                PIC X(4).                                   10/16/87
  43  003900     05  DUMP-RECORD                 PIC X(80).                                  10/16/87
  44  004000 FD  QPRINT                                                                       10/16/87
  45  004100     LABEL RECORDS ARE OMITTED.                                                   10/16/87
  46  004200 01  PRINTREC             PIC X(132).                                           10/16/87
  47  004300 WORKING-STORAGE SECTION.                                                         10/16/87
  48  004400 77  MAJ-MIN-SAV                     PIC X(4).                                   10/16/87
  49  004500 77  STATUS-IND                      PIC X(2).                                   10/16/87
  50  004600 77  INDON                           PIC 1 VALUE B"1".                           10/16/87
  51  004700 77  INDOFF                          PIC 1 VALUE B"0".                           10/16/87
  52  004800 77  LEN                             PIC 9(10)V9(5) COMP                         10/16/87
  53  004900                                     VALUE 0.                                    10/16/87
  54  005000 77  CMD2                            PIC X(31)                                   10/16/87
  55  005100       VALUE "CPYF HEXDUMP *LIST PRTFMT(*HEX)".                                   10/16/87
  56  005200 01  CMNF-INDIC-AREA.                                                            10/16/87
  57  005300     05  CMNF-INDIC                  PIC 1 OCCURS 99 TIMES                       10/16/87
  58  005400                                     INDICATOR 1.                                10/16/87
  59  005500 01  OPEN-FBA.                                                                  10/16/87
  60  005600     05  FILLER                      PIC X(75).                                  10/16/87
  61  005700     05  RECS-IN-DB                  PIC 9(09) COMP-4.                           10/16/87
  62  005800     05  FILLER                      PIC X(45).                                  10/16/87
  63  005900 01  MAJ-MIN.                                                                    10/16/87
  64  006000     05  MAJ                         PIC X(2).                                   10/16/87
  65  006100     05  MIN                         PIC X(2).                                   10/16/87
5728CB1 R01 M02 881028               COBOL SOURCE LISTING            ICFLIB/CTDBAT          03/01/89 13:44:52       Page     4
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
      006200/                                                                             10/16/87
```

Figure 10-11 (Part 2 of 4). Target Program Example — CTDBAT (User-Defined Formats)

```
 66  006300 PROCEDURE DIVISION.                                                    10/16/87
     006400 DECLARATIVES.                                                          10/16/87
     006500 ERR-SECTION SECTION.                                                   10/16/87
     006600* 2                                                                    10/16/87
     006700*                                                                       10/16/87
     006800     USE AFTER STANDARD ERROR PROCEDURE ON TGTICF.                      10/16/87
     006900 TGTICF-EXCEPTION.                                                      10/16/87
     007000*                                                                       10/16/87
     007100*************************************************************************02/21/89
     007200* GET INFORMATION FROM THE MAJOR-MINOR CODE AND PLACE IT INTO      *    02/21/89
     007300* A DATA BASE FILE.  THEN PRINT THE FILE IN HEX USING COPYFILE.    *    02/21/89
     007400*************************************************************************02/21/89
     007500*                                                                       10/16/87
     007600 GETFBA.                                                                10/16/87
 67  007700     MOVE "CTDBAT HAS COMPLETED ABNORMALLY" TO PRINTREC.                10/16/87
 68  007800     WRITE PRINTREC.                                                    10/16/87
 69  007900     OPEN OUTPUT DFILE.                                                 10/16/87
 70  008000     MOVE MAJ-MIN TO DUMP-MAJ-MIN.                                      10/16/87
 71  008100     MOVE ICFREC TO DUMP-RECORD.                                        10/16/87
 72  008200     WRITE DUMPREC.                                                     10/16/87
 73  008300     CLOSE DFILE.                                                       10/16/87
 74  008400     MOVE 31 TO LEN.                                                    10/16/87
 75  008500     CALL "QCAEXEC" USING CMD2 LEN.                                     10/16/87
 76  008600     STOP RUN.                                                          10/16/87
     008700*                                                                       10/16/87
     008800 EXIT-DECLARATIVES.                                                     10/16/87
     008900     EXIT.                                                              10/16/87
     009000*                                                                       10/16/87
 77  009100 END DECLARATIVES.                                                      10/16/87
5728CB1 R01 M02 881028              COBOL SOURCE LISTING          ICFLIB/CTDBAT          03/01/89 13:44:52       Page     5
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
     009200/                                                                       10/16/87
     009300 START-PROGRAM  SECTION.                                                10/16/87
     009400 START-PROGRAM-PARAGRAPH.                                               10/16/87
     009500* 3                                                                    10/16/87
 78  009600     OPEN OUTPUT QPRINT                                                 10/16/87
     009700          I-O    TGTICF                                                 10/16/87
     009800          INPUT  DBFILE.                                                10/16/87
     009900*                                                                       10/16/87
     010000*************************************************************************02/21/89
     010100* DATA CONTINUES TO BE RECEIVED FROM THE PROGRAM DEVICE UNTIL THE  *    02/21/89
     010200* RCVTRNRND INDICATOR IS SET. EACH RECORD RECEIVED IS PRINTED TO   *    02/21/89
     010300* THE PRINT FILE.                                                  *    02/21/89
     010400*************************************************************************02/21/89
     010500*                                                                       10/16/87
     010600 RECEIVE-DATA.                                                          10/16/87
     010700* 4                                                                    10/16/87
 79  010800     READ TGTICF INDICATORS ARE CMNF-INDIC-AREA.                        10/16/87
 80  010900     IF MAJ NOT = "03"                                                  10/16/87
 81  011000        MOVE ICFREC TO PRINTREC                                         10/16/87
 82  011100        WRITE PRINTREC.                                                 10/16/87
     011200* 5                                                                    10/16/87
 83  011300     IF CMNF-INDIC(15) NOT = INDON                                      10/16/87
 84  011400         GO TO RECEIVE-DATA.                                            10/16/87
     011500*                                                                       10/16/87
     011600*************************************************************************02/21/89
     011700* RECORD IS READ FROM THE DATABASE FILE AND SENT TO THE SOURCE     *    02/21/89
     011800* PROGRAM. DATA TRANSMISSION CONTINUES UNTIL END OF FILE IS        *    02/21/89
     011900* DETECTED ON THE DATABASE FILE.  AT THIS TIME, A DETACH REQUEST   *    02/21/89
     012000* IS SENT TO THE SOURCE PROGRAM.                                   *    02/21/89
```

Figure 10-11 (Part 3 of 4). Target Program Example — CTDBAT (User-Defined Formats)

```
       012100*****************************************************************                  02/21/89
       012200*                                                                                 10/16/87
       012300 SEND-DATA.                                                                       10/16/87
       012400* 6                                                                              10/16/87
    85 012500     READ DBFILE AT END GO TO SIGNAL-DETACH.                                     10/16/87
    87 012600     WRITE ICFREC FROM DBREC FORMAT IS "SNDDATA"                                 10/16/87
       012700         INDICATORS ARE CMNF-INDIC-AREA.                                         10/16/87
    88 012800     GO TO SEND-DATA.                                                            10/16/87
       012900*                                                                                10/16/87
       013000*****************************************************************                  02/21/89
       013100* SIGNAL DETACH TO THE SOURCE PROGRAM.                          *                  02/21/89
       013200*****************************************************************                  02/21/89
       013300*                                                                                 10/16/87
       013400 SIGNAL-DETACH.                                                                   10/16/87
       013500* 7                                                                              10/16/87
    89 013600     MOVE INDON TO CMNF-INDIC(30).                                               10/16/87
    90 013700     WRITE ICFREC FORMAT IS "ENDREC"                                             10/16/87
       013800        INDICATORS ARE CMNF-INDIC-AREA.                                          10/16/87
       013900*****************************************************************                  02/21/89
       014000*   WHEN THE END OF FILE IS REACHED, AN EOJ MESSAGE IS PRINTED    *                  02/21/89
       014100*   AND THE PROGRAM ENDS.                                         *                  02/21/89
       014200*****************************************************************                  02/21/89
       014300* 8                                                                              10/16/87
    91 014400     MOVE "CTDBAT HAS COMPLETED NORMALLY" TO PRINTREC                            10/16/87
    92 014500     WRITE PRINTREC.                                                             10/16/87
    93 014600     CLOSE DBFILE                                                                10/16/87
 5728CB1 R01 M02 881028              COBOL SOURCE LISTING            ICFLIB/CTDBAT          03/01/89 13:44:52        Page     6
  STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
       014700          TGTICF                                                                 10/16/87
       014800          QPRINT.                                                                10/16/87
    94 014900     STOP RUN.                                                                   10/16/87
                        * * * * *  E N D  O F  S O U R C E  * * * * *
 5728CB1 R01 M02 881028                COBOL MESSAGES                ICFLIB/CTDBAT          03/01/89 13:44:52        Page     7
  STMT
*   19  MSGID: LBL0650  SEVERITY: 00  SEQNBR:  002700
       Message . . . . :   Blocking/Deblocking for file 'DBFILE' will
         be performed by compiler-generated code.
*   26  MSGID: LBL0600  SEVERITY: 10  SEQNBR:  003400
       Message . . . . :   No INPUT fields found for format EVOKPGM.
*   26  MSGID: LBL0600  SEVERITY: 10  SEQNBR:  003400
       Message . . . . :   No INPUT fields found for format ENDREC.
*   26  MSGID: LBL0600  SEVERITY: 10  SEQNBR:  003400
       Message . . . . :   No OUTPUT fields found for format ENDREC.
*   26  MSGID: LBL0600  SEVERITY: 10  SEQNBR:  003400
       Message . . . . :   No INPUT fields found for format INVITE.
*   26  MSGID: LBL0600  SEVERITY: 10  SEQNBR:  003400
       Message . . . . :   No OUTPUT fields found for format INVITE.
*   39  MSGID: LBL0650  SEVERITY: 00  SEQNBR:  003500
       Message . . . . :   Blocking/Deblocking for file 'DFILE' will be
         performed by compiler-generated code.
*   67  MSGID: LBL0335  SEVERITY: 00  SEQNBR:  007600
       Message . . . . :   Empty paragraph or section precedes 'GETFBA'
         paragraph or section.
                                    MESSAGE SUMMARY
  TOTAL    INFO(0-4)    WARNING(5-19)    ERROR(20-29)    SEVERE(30-39)    TERMINAL(40-99)
     8          3              5               0               0                0
                   * * * * *  E N D  O F  C O B O L  M E S S A G E S  * * * * *
 149 source records read
 37 copy records read
 1 copy members processed
 0 sequence errors
 10 was the highest severity message issued
  LBL0901 00  Program CTDBAT created in library ICFLIB.
                        * * * * *  E N D  O F  C O M P I L A T I O N  * * * * *
```

Figure 10-11 (Part 4 of 4). Target Program Example — CTDBAT (User-Defined Formats)

```
5728CB1 R01 M02 881028              IBM AS/400  COBOL/400          ICFLIB/CTFBAT          03/01/89 13:46:01       Page    1
Program name . . . . . . . . . . . . : CTFBAT in ICFLIB
Source file  . . . . . . . . . . . . : QICFPUB in ICFLIB     Member - CTFBAT     02/27/89 09:38:46
Compiler option  . . . . . . . . . . : *NONE
Code generation option . . . . . . . : *NONE
Code generation severity level . . . : 29
Print file . . . . . . . . . . . . . : QSYSPRT in *LIBL
FIPS flagging option . . . . . . . . : *NOFIPS *NOSEG *NODEB *NOOBSOLETE
Text not available for message LBX6039 file QLBLMSG.
Flagging level . . . . . . . . . . . : 0
Replace existing program . . . . . . : *YES
User profile . . . . . . . . . . . . : *USER
Authority  . . . . . . . . . . . . . : *CHANGE
Text . . . . . . . . . . . . . . . . : cobol batch file transfer using $$FORMAT (target)
Compiler . . . . . . . . . . . . . . : IBM AS/400  COBOL/400
5728CB1 R01 M02 881028              COBOL SOURCE LISTING          ICFLIB/CTFBAT          03/01/89 13:46:01       Page    2
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
    1  000100 IDENTIFICATION DIVISION.                                                                 02/21/89
    2  000200 PROGRAM-ID.                CTFBAT.                                                       02/21/89
       000300*************************************************************************               02/22/89
       000400*   THIS TARGET PROGRAM IS EVOKED BY THE SOURCE PROGRAM AND        *                      02/22/89
       000500*   RECEIVES RECORDS FROM IT. WHEN THE SOURCE PROGRAM IS DONE      *                      02/22/89
       000600*   SENDING DATA, THIS PROGRAM SENDS ITS OWN RECORDS UNTIL IT IS   *                      02/22/89
       000700*   DONE.  WHEN THIS PROGRAM IS DONE, IT SENDS A DETACH REQUEST TO *                      02/22/89
       000800*   THE SOURCE PROGRAM AND ENDS ITS SESSION AND JOB.               *                      02/22/89
       000900*************************************************************************               02/22/89
    3  001000 ENVIRONMENT DIVISION.                                                                    10/16/87
    4  001100 CONFIGURATION SECTION.                                                                   10/16/87
    5  001200 SOURCE-COMPUTER.           IBM-AS400.                                                    02/21/89
    6  001300 OBJECT-COMPUTER.           IBM-AS400.                                                    02/21/89
    7  001400 SPECIAL-NAMES.             I-O-FEEDBACK IS FEEDBACK-AREA                                 10/16/87
    8  001500                            OPEN-FEEDBACK IS OPEN-FBA.                                    10/16/87
    9  001600 INPUT-OUTPUT SECTION.                                                                    10/16/87
       001700* 1                                                                                       10/16/87
   10  001800 FILE-CONTROL.                                                                            10/16/87
   11  001900     SELECT DBFILE ASSIGN TO DATABASE-DBFILE.                                             10/16/87
   12  002000     SELECT TGTICF ASSIGN TO WORKSTATION-TGTICF-SI                                        10/16/87
   13  002100         ORGANIZATION IS TRANSACTION                                                      10/16/87
   14  002200         FILE STATUS IS STATUS-IND MAJ-MIN.                                               10/16/87
   15  002300     SELECT DFILE ASSIGN TO DATABASE-HEXDUMP.                                             10/16/87
   16  002400     SELECT QPRINT ASSIGN TO PRINTER-QSYSPRT.                                             10/16/87
   17  002500 DATA DIVISION.                                                                           10/16/87
   18  002600 FILE SECTION.                                                                            10/16/87
   19  002700 FD  DBFILE                                                                               10/16/87
   20  002800     LABEL RECORDS ARE STANDARD.                                                          10/16/87
   21  002900 01  PREC.                                                                               10/16/87
   22  003000     05  PREC-DATA                  PIC X(80).                                           10/16/87
   23  003100 FD  TGTICF                                                                               10/16/87
   24  003200     LABEL RECORDS ARE STANDARD.                                                          10/16/87
   25  003300 01  ICFREC.                                                                             10/16/87
   26  003400     05  SNDDATA-O.                                                                      10/16/87
   27  003500         10  SNDLENGTH              PIC 9(4).                                            10/16/87
   28  003600         10  SNDFIELD               PIC X(80).                                           10/16/87
   29  003700     05  ENDREC-O REDEFINES SNDDATA-O.                                                   10/16/87
   30  003800         10  ENDLENGTH              PIC 9(4).                                            10/16/87
   31  003900         10  FILLER                 PIC X(80).                                           10/16/87
   32  004000 FD  DFILE                                                                                10/16/87
   33  004100     LABEL RECORDS ARE STANDARD.                                                          10/16/87
   34  004200 01  DUMPREC.                                                                            10/16/87
   35  004300     05  DUMP-MAJ-MIN               PIC X(4).                                            10/16/87
```

Figure 10-12 (Part 1 of 4). Target Program Example—CTFBAT (System-Supplied Formats)

```
    36  004400     05  DUMP-RECORD                PIC X(80).                                    10/16/87
    37  004500 FD  QPRINT                                                                       10/16/87
    38  004600     LABEL RECORDS ARE OMITTED.                                                   10/16/87
    39  004700 01  PRINTREC            PIC X(132).                                            10/16/87
    40  004800 WORKING-STORAGE SECTION.                                                         10/16/87
    41  004900 77  MAJ-MIN-SAV                   PIC X(4).                                     10/16/87
    42  005000 77  STATUS-IND                    PIC X(2).                                     10/16/87
    43  005100 77  INDON                         PIC 1 VALUE B"1".                             10/16/87
    44  005200 77  INDOFF                        PIC 1 VALUE B"0".                             10/16/87
    45  005300 77  LEN                           PIC 9(10)V9(5) COMP                           10/16/87
    46  005400                                   VALUE 0.                                      10/16/87
    47  005500 77  CMD2                          PIC X(31)                                     10/16/87
5728CB1 R01 M02 881028              COBOL SOURCE LISTING           ICFLIB/CTFBAT          03/01/89 13:46:01        Page     3
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
    48  005600       VALUE "CPYF HEXDUMP *LIST PRTFMT(*HEX)".                                  10/16/87
    49  005700 01  OPEN-FBA.                                                                   10/16/87
    50  005800     05  FILLER                   PIC X(75).                                    10/16/87
    51  005900     05  RECS-IN-DB               PIC 9(09) COMP-4.                             10/16/87
    52  006000     05  FILLER                   PIC X(45).                                    10/16/87
    53  006100 01  MAJ-MIN.                                                                    10/16/87
    54  006200     05  MAJ                      PIC X(2).                                     10/16/87
    55  006300     05  MIN                      PIC X(2).                                     10/16/87
5728CB1 R01 M02 881028              COBOL SOURCE LISTING           ICFLIB/CTFBAT          03/01/89 13:46:01        Page     4
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
        006400/                                                                               10/16/87
    56  006500 PROCEDURE DIVISION.                                                            10/16/87
        006600 DECLARATIVES.                                                                  10/16/87
        006700 ERR-SECTION SECTION.                                                           10/16/87
        006800* 2                                                                            10/16/87
        006900*                                                                               10/16/87
        007000     USE AFTER STANDARD ERROR PROCEDURE ON TGTICF.                              10/16/87
        007100 TGTICF-EXCEPTION.                                                              10/16/87
        007200*                                                                               02/27/89
        007300*****************************************************************            02/27/89
        007400* GET INFORMATION FROM THE MAJOR-MINOR CODE AND PLACE IT INTO      *           02/27/89
        007500* A DATA BASE FILE.  THEN PRINT THE FILE IN HEX USING COPYFILE.    *           02/27/89
        007600*****************************************************************            02/27/89
        007700*                                                                               02/27/89
        007800 GETFBA.                                                                        10/16/87
    57  007900     MOVE "CTFBAT HAS COMPLETED ABNORMALLY" TO PRINTREC.                        10/16/87
    58  008000     WRITE PRINTREC.                                                            10/16/87
    59  008100     OPEN OUTPUT DFILE.                                                         10/16/87
    60  008200     MOVE MAJ-MIN TO DUMP-MAJ-MIN.                                              10/16/87
    61  008300     MOVE ICFREC TO DUMP-RECORD.                                                10/16/87
    62  008400     WRITE DUMPREC.                                                             10/16/87
    63  008500     CLOSE DFILE.                                                               10/16/87
    64  008600     MOVE 31 TO LEN.                                                            10/16/87
    65  008700     CALL "QCAEXEC" USING CMD2 LEN.                                             10/16/87
    66  008800     STOP RUN.                                                                  10/16/87
        008900*                                                                               10/16/87
        009000 EXIT-DECLARATIVES.                                                             10/16/87
        009100     EXIT.                                                                      10/16/87
        009200*                                                                               10/16/87
    67  009300 END DECLARATIVES.                                                             10/16/87
5728CB1 R01 M02 881028              COBOL SOURCE LISTING           ICFLIB/CTFBAT          03/01/89 13:46:01        Page     5
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
        009400/                                                                               10/16/87
        009500 START-PROGRAM  SECTION.                                                        10/16/87
        009600 START-PROGRAM-PARAGRAPH.                                                       10/16/87
        009700* 3                                                                            10/16/87
    68  009800     OPEN OUTPUT QPRINT                                                         10/16/87
        009900          I-O   TGTICF                                                         10/16/87
        010000          INPUT DBFILE.                                                        10/16/87
        010100*                                                                               10/16/87
```

Figure 10-12 (Part 2 of 4). Target Program Example—CTFBAT (System-Supplied Formats)

```
    010200*****************************************************************          02/22/89
    010300* DATA CONTINUES TO BE RECEIVED FROM THE PROGRAM DEVICE UNTIL THE  *          02/22/89
    010400* RCVTRNRND INDICATOR IS SET. EACH RECORD RECEIVED IS PRINTED TO   *          02/22/89
    010500* THE PRINT FILE.                                                  *          02/22/89
    010600*****************************************************************          02/22/89
    010700*                                                                          10/16/87
    010800 RECEIVE-DATA.                                                             10/16/87
    010900* 4                                                                       10/16/87
 69 011000     READ TGTICF.                                                         10/16/87
 70 011100     IF MAJ NOT = "03"                                                    10/16/87
 71 011200        MOVE ICFREC TO PRINTREC                                           10/16/87
 72 011300        WRITE PRINTREC.                                                   10/16/87
    011400* 5                                                                       10/16/87
 73 011500     IF MIN = "01" THEN                                                   10/16/87
 74 011600        GO TO RECEIVE-DATA.                                               10/16/87
    011700*                                                                         10/16/87
    011800*****************************************************************          02/22/89
    011900* RECORD IS READ FROM THE DATABASE FILE AND IS SENT TO THE SOURCE  *          02/22/89
    012000* PROGRAM. DATA TRANSMISSION CONTINUES UNTIL END OF FILE IS        *          02/22/89
    012100* DETECTED ON THE DATABASE FILE. AT THIS TIME, A DETACH SIGNAL IS  *          02/22/89
    012200* IS SENT TO THE SOURCE PROGRAM.                                   *          02/22/89
    012300*****************************************************************          02/22/89
    012400*                                                                          10/16/87
    012500 SEND-DATA.                                                                10/16/87
    012600* 6                                                                       10/16/87
 75 012700     READ DBFILE AT END GO TO SIGNAL-DETACH.                              10/16/87
 77 012800     MOVE PREC TO SNDFIELD OF SNDDATA-O.                                  10/16/87
 78 012900     MOVE +80 TO SNDLENGTH OF SNDDATA-O.                                  10/16/87
 79 013000     WRITE ICFREC FORMAT IS "$$SENDNI".                                   10/16/87
 80 013100     GO TO SEND-DATA.                                                     10/16/87
    013200*                                                                         10/16/87
    013300*****************************************************************          02/22/89
    013400* SIGNAL DETACH TO THE SOURCE PROGRAM.                             *          02/22/89
    013500*****************************************************************          02/22/89
    013600*                                                                          10/16/87
    013700 SIGNAL-DETACH.                                                            10/16/87
    013800* 7                                                                       10/16/87
 81 013900     MOVE SPACES TO ENDREC-O.                                             10/16/87
 82 014000     MOVE +0 TO ENDLENGTH OF ENDREC-O.                                    10/16/87
 83 014100     WRITE ICFREC FORMAT IS "$$SENDET".                                   10/16/87
    014200*****************************************************************          02/22/89
    014300*   WHEN THE END OF FILE IS REACHED, AN EOJ MESSAGE IS PRINTED AND *          02/22/89
    014400*   THE PROGRAM ENDS.                                              *          02/22/89
    014500*****************************************************************          02/22/89
    014600* 8                                                                       10/16/87
 84 014700     MOVE "CTFBAT HAS COMPLETED NORMALLY" TO PRINTREC                     10/16/87
 85 014800     WRITE PRINTREC.                                                      10/16/87
5728CB1 R01 M02 881028              COBOL SOURCE LISTING          ICFLIB/CTFBAT          03/01/89 13:46:01       Page     6
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
 86 014900     CLOSE DBFILE                                                         10/16/87
    015000           TGTICF                                                        10/16/87
    015100           QPRINT.                                                       10/16/87
 87 015200     STOP RUN.                                                            10/16/87
                     * * * * *  E N D  O F  S O U R C E  * * * * *
5728CB1 R01 M02 881028                COBOL MESSAGES              ICFLIB/CTFBAT          03/01/89 13:46:01       Page     7
 STMT
```

Figure 10-12 (Part 3 of 4). Target Program Example—CTFBAT (System-Supplied Formats)

```
*   19  MSGID: LBL0650  SEVERITY: 00  SEQNBR:  002700
        Message . . . . :   Blocking/Deblocking for file 'DBFILE' will
          be performed by compiler-generated code.
*   32  MSGID: LBL0650  SEVERITY: 00  SEQNBR:  004000
        Message . . . . :   Blocking/Deblocking for file 'DFILE' will be
          performed by compiler-generated code.
*   57  MSGID: LBL0335  SEVERITY: 00  SEQNBR:  007800
        Message . . . . :   Empty paragraph or section precedes 'GETFBA'
          paragraph or section.
                                        MESSAGE SUMMARY
  TOTAL    INFO(0-4)    WARNING(5-19)    ERROR(20-29)    SEVERE(30-39)    TERMINAL(40-99)
      3            3                0               0               0                  0
                    * * * * *  E N D  O F  C O B O L  M E S S A G E S  * * * * *
 152 source records read
 0 copy records read
 0 copy members processed
 0 sequence errors
 0 was the highest severity message issued
  LBL0901 00  Program CTFBAT created in library ICFLIB.
                        * * * * *  E N D  O F  C O M P I L A T I O N  * * * * *
```

Figure 10-12 (Part 4 of 4). Target Program Example—CTFBAT (System-Supplied Formats)

Multiple-Session Inquiry (Example II)

This example illustrates an interactive inquiry application that communicates with multiple ICF sessions. A source AS/400 system program accepts inquiries from a display device and sends a request to one of four AS/400 systems. The source program communicates with the display device through a display file, and with the four remote systems through a single ICF file.

The purpose of this example is to show multiple sessions from a single ICF file. The source program communicates with four sessions. From the viewpoint of each of the four target programs, there is only one session (with the requesting program device). Therefore, the target programs do not require any unique logic to support the multiple-session source.

Both the source program and the target program are described. The same target program is evoked in each of the four separate remote systems. Therefore, only one target program is shown in the programming example.

Transaction Flow of the Multiple-Session Inquiry (Example II)

The program shown in Figure 10-13 is started from a display station, and both the display and the ICF file are opened. CIWS00 is the *REQUESTER device, acquired when the display file opens. CIWS00 is acquired because DEV(*REQUESTER) was specified when the display file was created. Since the ICF file was created with ACQPGMDEV(*NONE), no ICF program devices are acquired during open processing.

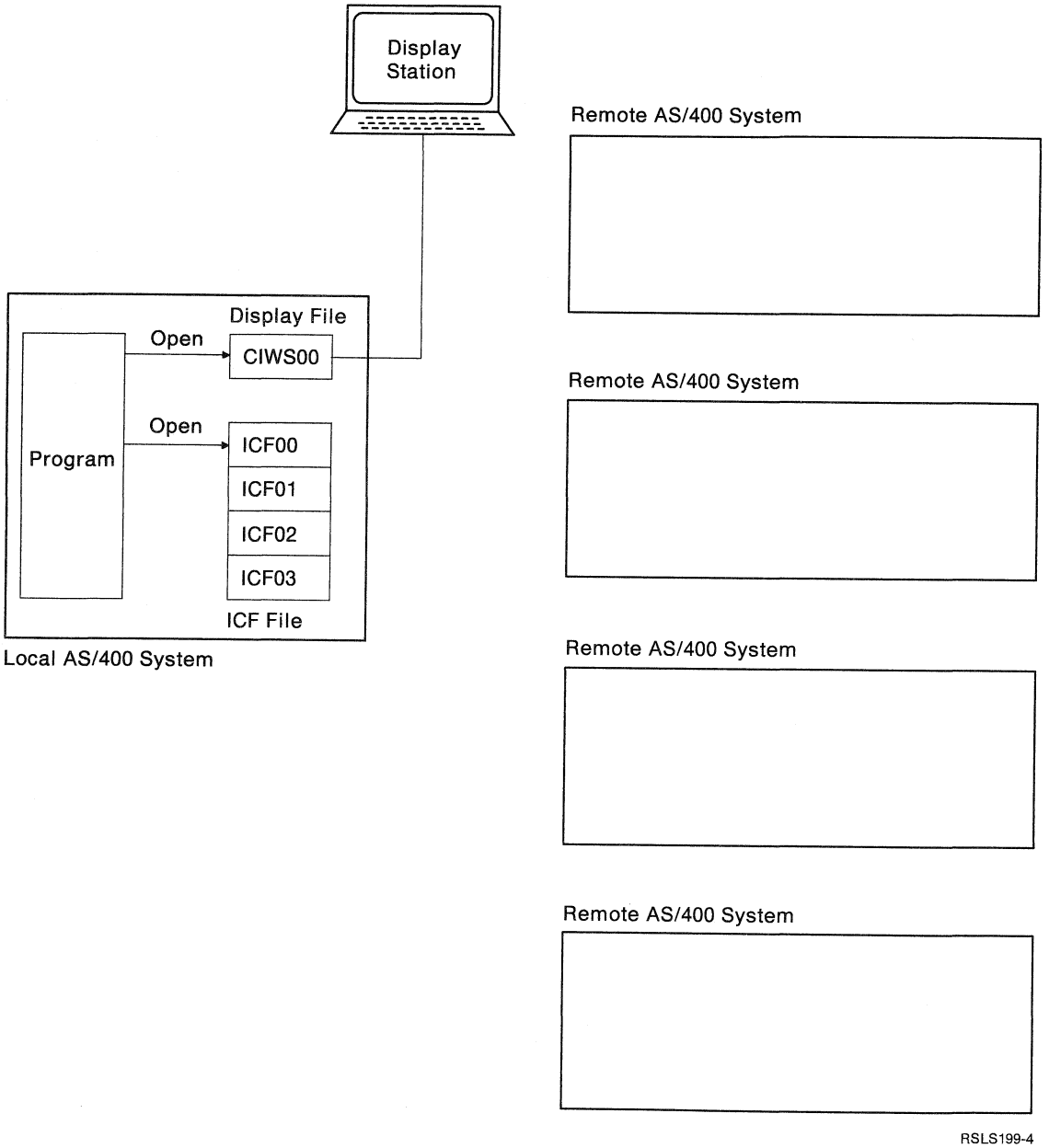


Figure 10-13. Program Starts at Display Station

All other program devices must be explicitly acquired by the program, as shown in Figure 10-14.

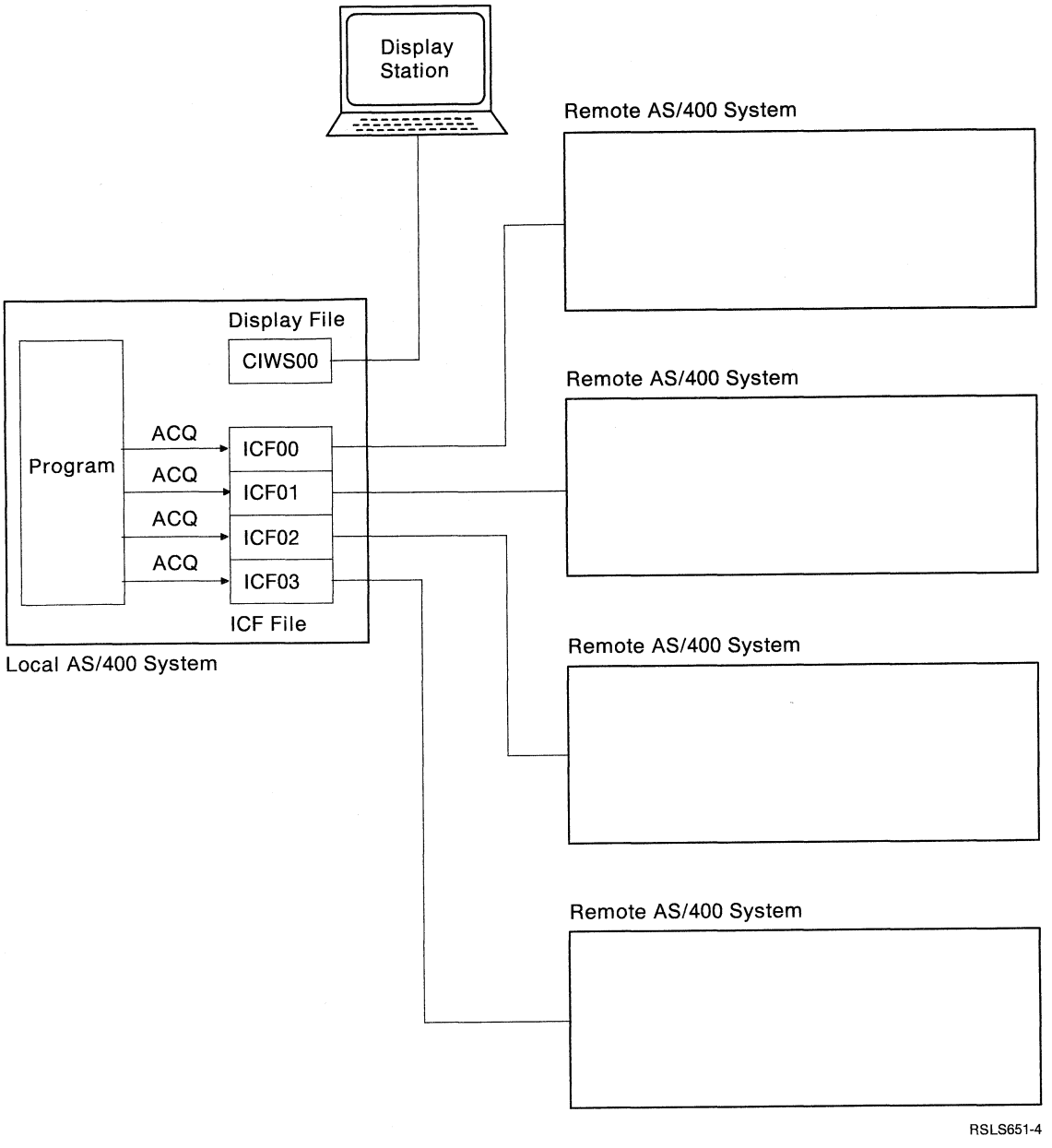


Figure 10-14. Program Devices Explicitly Acquired

All target programs are started with the evoke, as shown in Figure 10-15.

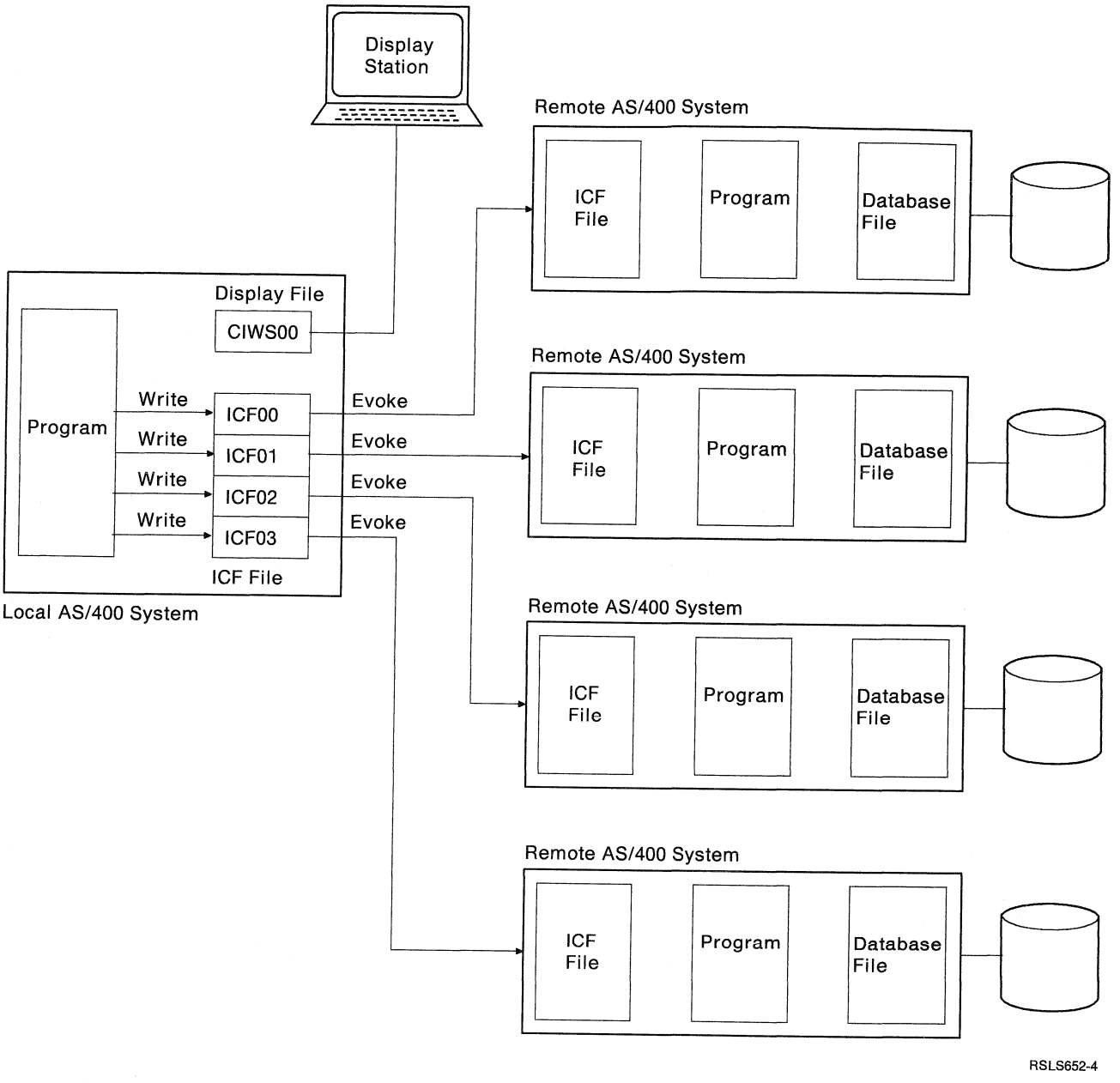


Figure 10-15. Evoke Starts Target Programs

The source program uses a specific program device name. Each target program uses an ICF file with a program device name that is associated with the requesting program device. The target program's only session is the one used to communicate with the source program. The ICF file on the remote system must be opened by the COBOL language support using the open operation. Since the file was created with the requesting program device specified on the ACQPGMDEV parameter, the requesting program device is acquired when the file is opened.

The main menu is written to the display station on the local system, and the program waits for a request from the display station, as shown in Figure 10-16.

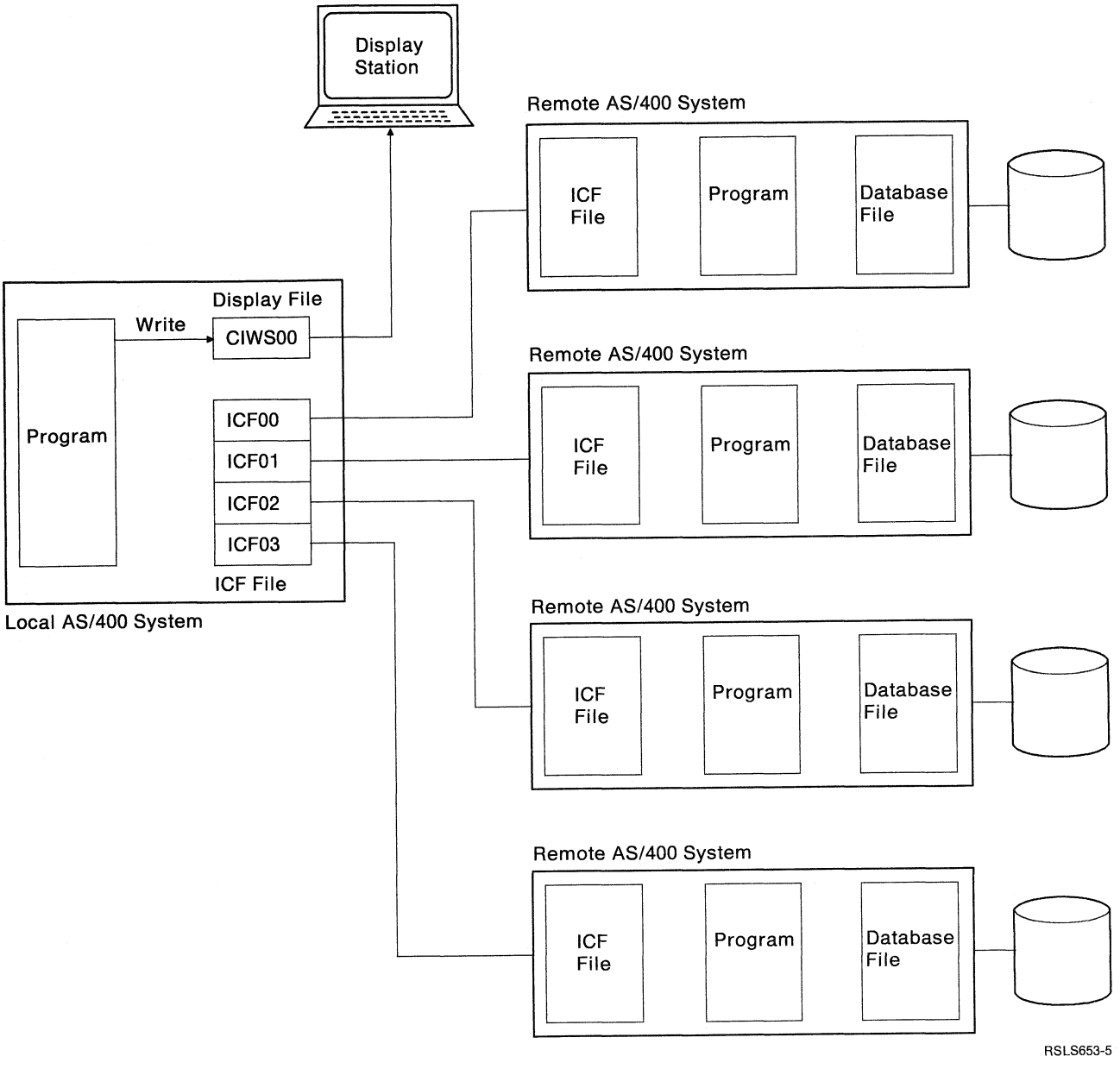


Figure 10-16. Main Menu Written to Display Station

The source program sends an inquiry request to one of the remote systems based on the request made from the display station, as shown in Figure 10-17.

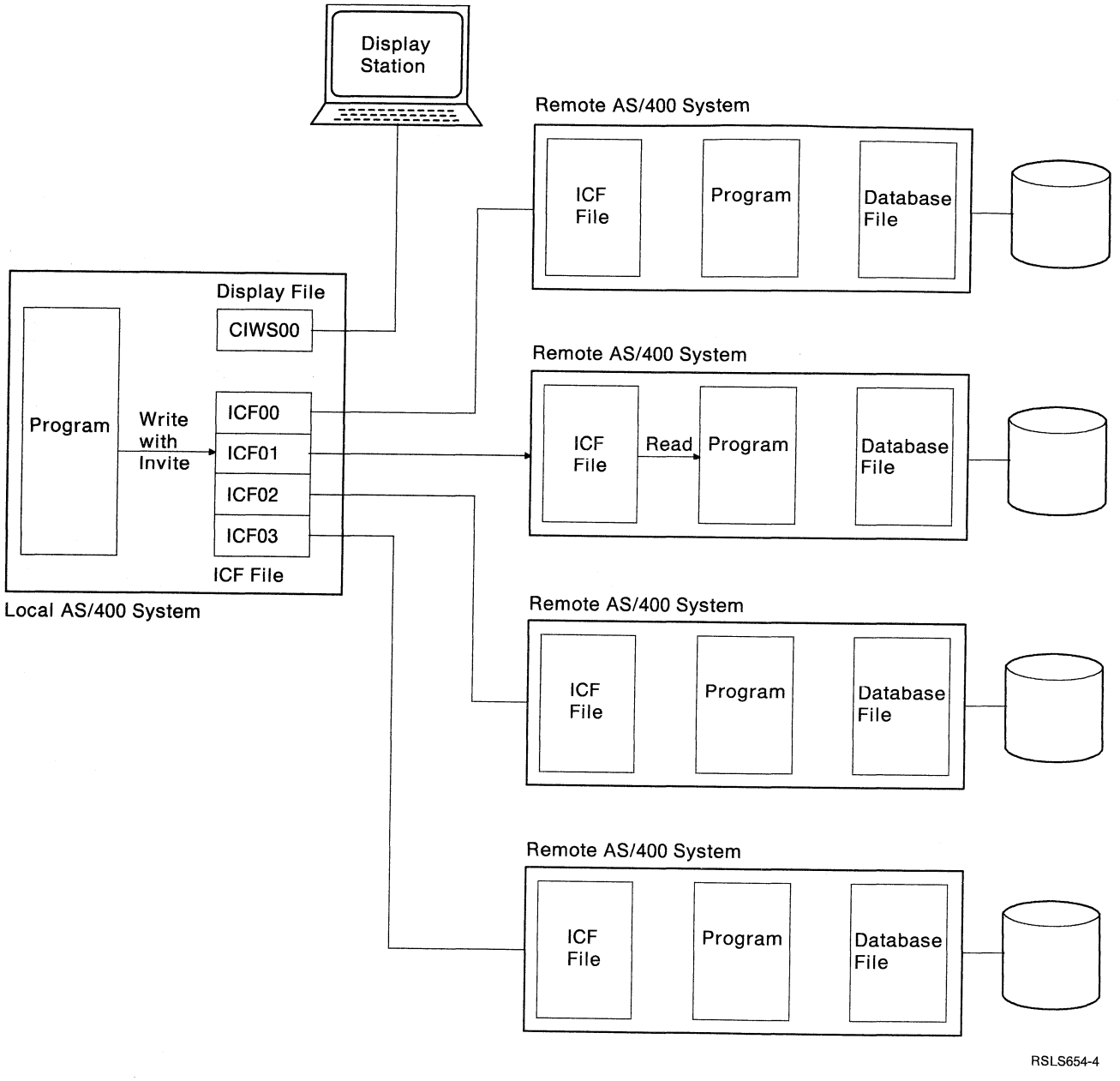


Figure 10-17. Program Sends Inquiry Request to Remote System

The target program responds to the inquiry by sending a reply, as shown in Figure 10-18.

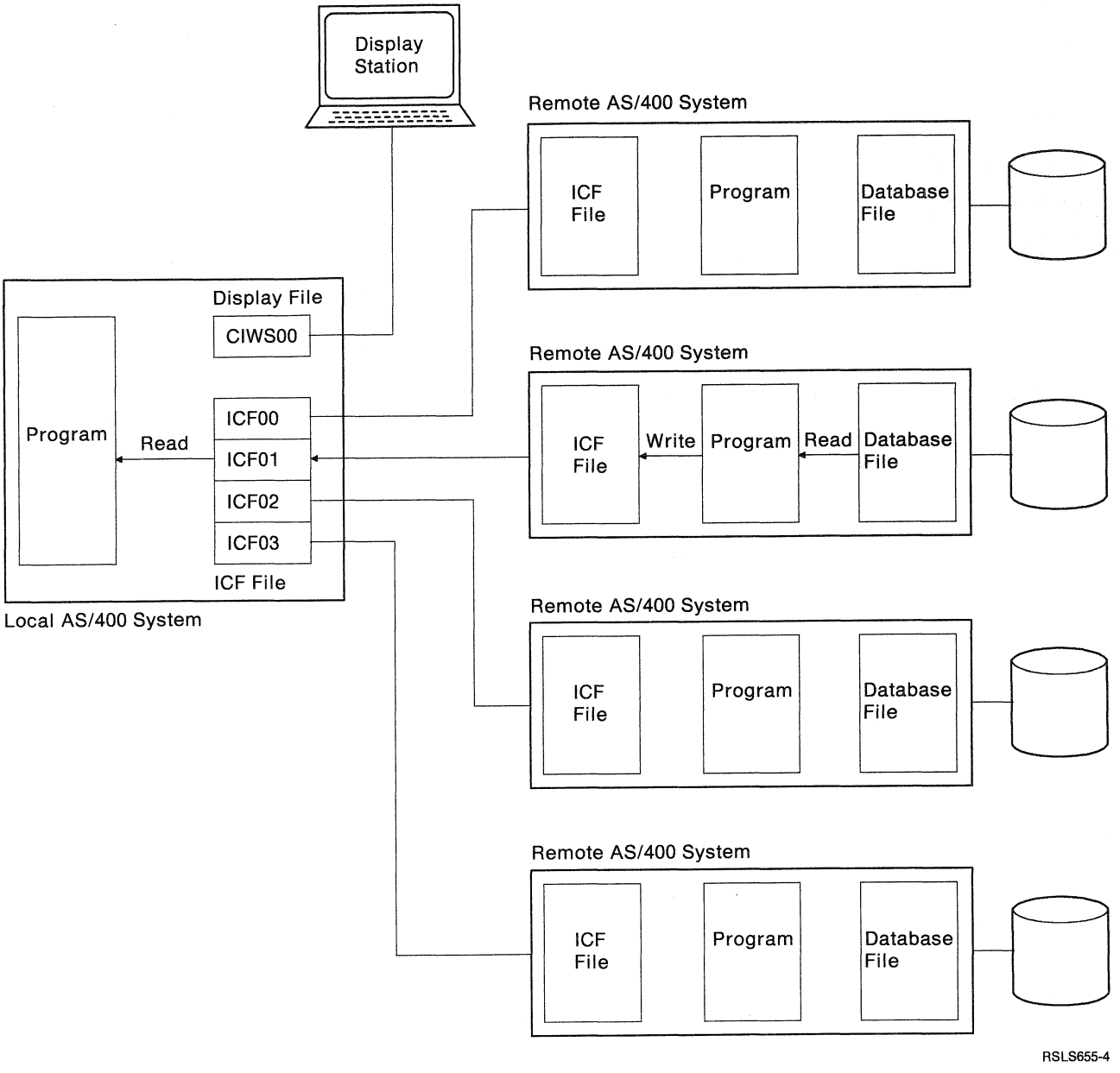


Figure 10-18. Target Program Sends a Reply

The program sends a detach request and ends the session when function key 1 is pressed (while the main inquiry menu is present), as shown in Figure 10-19.

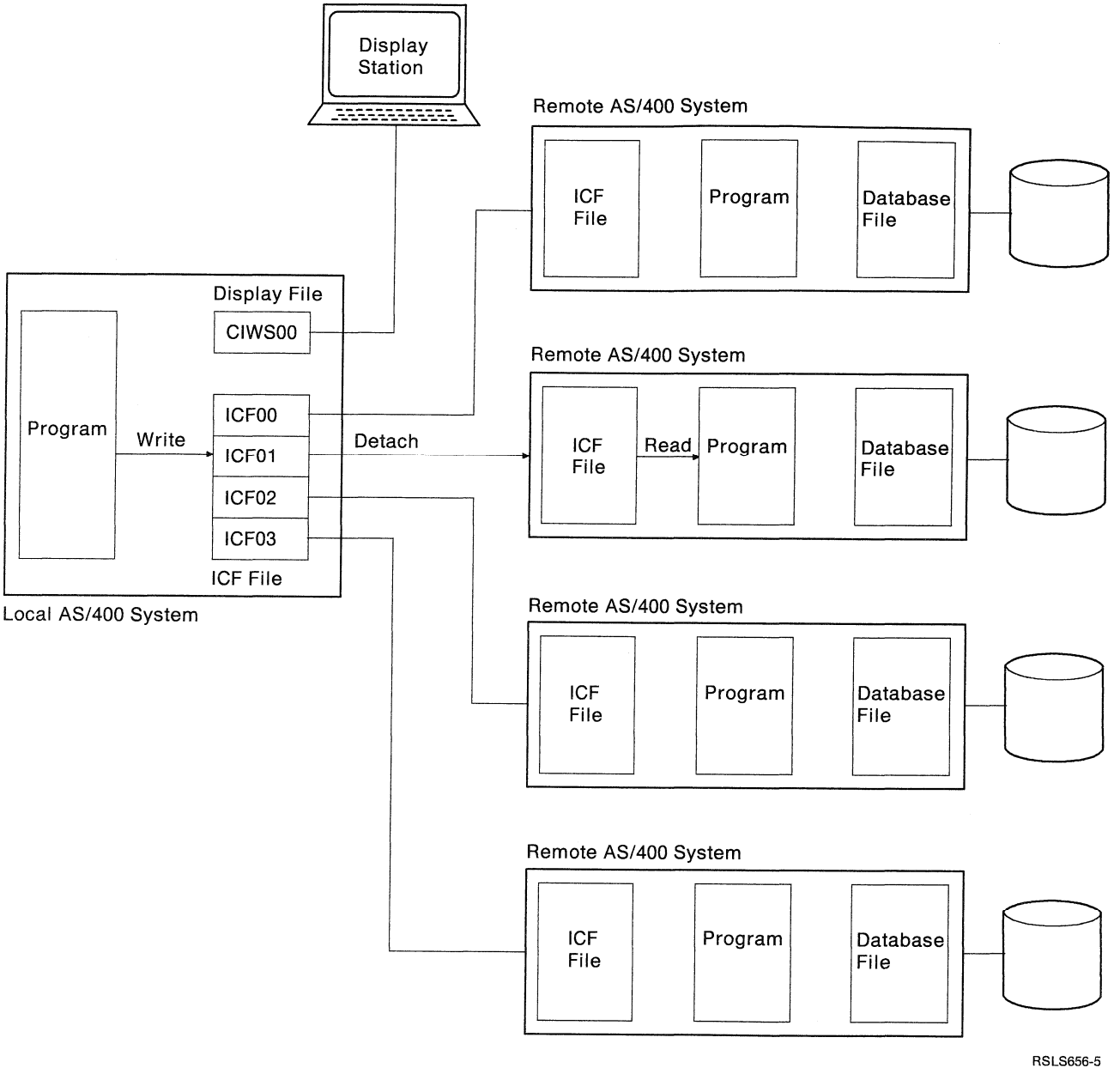


Figure 10-19. Program Ends the Session

Source Program Multiple-Session Inquiry (Example II)

The following describes a COBOL source program multiple-session inquiry.

Program Files: The COBOL multiple session source program uses the following files:

CMNFIL	An ICF file used to send records to and receive records from the target program.
DSPFIL	A display file used to enter requests that are to be sent to the target program.
QPRINT	A printer file used to print error messages resulting from communications errors.

DDS Source: The DDS for the ICF file (CMNFIL) is illustrated in Figure 10-20.

```
5714PW1 R01M00  880301                SEU SOURCE LISTING                      10/14/87 17:20:41          PAGE 1
SOURCE FILE . . . . . .  QICFPUB/ICFLIB
MEMBER  . . . . . . . .  CMNFIL
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
  100                                                                                                  10/06/87
  200                                                                                                  10/06/87
  300      A*************************************************************                             10/14/87
  400      A*                                                           *                             10/14/87
  500      A*                        ICF FILE                           *                             10/14/87
  600      A*        USED IN SOURCE MULTIPLE SESSION PROGRAM            *                             10/14/87
  700      A*                                                           *                             10/14/87
  800      A*************************************************************                             10/14/87
  900      A                                     INDARA                                                10/07/87
 1000      A          R ITMRSP                                                                        10/06/87
 1100      A                                     RECID(1 'I')                                          10/06/87
 1200      A            RECITM        1                                                              10/06/87
 1300      A            ITEMNO        6  0                                                           10/13/87
 1400      A            DESC         30                                                              10/06/87
 1500      A            QTYLST        7  0                                                           10/06/87
 1600      A            QTYOH         7  0                                                           10/06/87
 1700      A            QTYOO         7  0                                                           10/06/87
 1800      A            QTYBO         7  0                                                           10/06/87
 1900      A            UNITQ         2                                                              10/06/87
 2000      A            PR01          7  2                                                           10/06/87
 2100      A            PR05          7  0                                                           10/06/87
 2200      A            UFRT          5  2                                                           10/06/87
 2300      A            SLSTM         9  2                                                           10/06/87
 2400      A            SLSTY        11  2                                                           10/06/87
 2500      A            CSTTM         9  2                                                           10/06/87
 2600      A            CSTTY        11  2                                                           10/06/87
 2700      A            PRO           5  2                                                           10/06/87
 2800      A            LOS           9  2                                                           10/06/87
 2900      A            FILL1        56                                                              10/06/87
 3000      A          R DTLRSP                                                                        10/06/87
 3100      A                                     RECID(1 'C')                                          10/06/87
 3200      A                                     RCVTRNRND(90)                                         10/06/87
 3300      A            RECCUS        1                                                              10/06/87
 3400      A            CUSTNO        6  0                                                           10/13/87
 3500      A            DNAME        30                                                              10/06/87
 3600      A            DLSTOR        6  0                                                           10/06/87
 3700      A            DSLSTM        9  0                                                           10/06/87
 3800      A            DSPM01        9  0                                                           10/06/87
 3900      A            DSPM02        9  0                                                           10/06/87
 4000      A            DSPM03        9  0                                                           10/06/87
 4100      A            DSTTYD       11  0                                                           10/06/87
 4200      A            IDEPT         3  0                                                           10/06/87
 4300      A            FILL2        57                                                              10/06/87
 4400      A          R DETACH                                                                        10/06/87
 4500      A                                     DETACH                                                10/06/87
 4600      A          R EOS                                                                           10/06/87
 4700      A                                     EOS                                                   10/06/87
 4800      A          R EVKREQ                                                                        10/06/87
 4900      A                                     EVOKE(&LIB/&PGMID)                                    10/12/87
 5000      A            PGMID        10A  P                                                          10/06/87
 5100      A            LIB          10A  P                                                          10/06/87
 5200      A          R ITMREQ                                                                        10/06/87
 5300      A                                     INVITE                                                10/06/87
5714PW1 R01M00  880301                SEU SOURCE LISTING                      10/14/87 17:20:41          PAGE 2
SOURCE FILE . . . . . .  QICFPUB/ICFLIB
MEMBER  . . . . . . . .  CMNFIL
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
 5400      A            ITEMNO        6  0                                                           10/13/87
 5500      A          R DTLREQ                                                                        10/06/87
 5600      A                                     INVITE                                                10/06/87
 5700      A            CUSTNO        6  0                                                           10/13/87
                      * * * *  E N D  O F  S O U R C E  * * * *
```

Figure 10-20. DDS for Source Program Multiple Session Inquiry Using CMNFIL

The DDS for the display file (DSPFIL) is illustrated in Figure 10-21.

```
000100871007   A*****************************************************************
000200871007   A*                                                               *
000300871007   A*                       DISPLAY FILE                            *
000400871007   A*              USED IN SOURCE MULTIPLE SESSION PROGRAM          *
000500871007   A*                                                               *
000600871007   A*****************************************************************
000700871008   A*  BEGINNING MENU
000800871008   A*******************
000900871007   A                                           DSPSIZ(*DS3)
001000871007   A                                           CF01(99) CF02(98) CF03(97)
001100871007   A          R CIMENU                         TEXT('MENU FOR INQUIRY')
001200871007   A                                       1 34'INQUIRY MENU'
001300871007   A                                       3  1'Select one of the following:'
001400871007   A                                       4  3'1. Item inquiry'
001500871007   A                                       5  3'2. Customer inquiry'
001600871007   A                                      11  1'Option:'
001700871007   A            OPTION         1N  I 11  9VALUES('1' '2')
001800871008   A                                      19  5DFT('CMD KEY 1 - END ')
001900871008   A          R DTLMNU                         TEXT('CUSTOMER INQUIRY SCREEN 1')
002000871007   A                                       2  2DFT('ENTER CUSTOMER')
002100871013   A            CUSTNO         6N 0I  2 20
002200871008   A                                      19  5DFT('CMD KEY 1 - END ')
002300871008   A                                      19 23DFT(' 2 - MAIN MENU ')
002400871008   A*
002500871008   A****************************
002600871007   A*  CUSTOMER INQUIRY SCREEN
002700871008   A****************************
002800871007   A          R DTLSCR                         TEXT('CUSTOMER INQUIRY SCR. #2')
002900871007   A                                       1  3DFT('CUST  DPT LAST ORD   & THIS   +
003000871007   A                                            $MTH1      &MTH2      $MTH3      THIS+
003100871008   A                                            YTD NAME')
003200871008   A            CUSTN          6N     2  2
003300871007   A            DEPT           3N 0   2  9
003400871007   A            DLSTR          6N 0   2 13
003500871007   A            DSLSM          9N 0   2 22
003600871007   A            DSPM1          9N 0   2 32
003700871007   A            DSPM2          9N 0   2 42
003800871007   A            DSPM3          9N 0   2 52
003900871007   A            DSTYD         11N 0   2 62
004000890321   A            CNAME          5      2 74
004100871008   A                                      19  5DFT('CMD KEY 1 - END ')
004200871008   A                                      19 23DFT(' 2 - MAIN MENU ')
004300871007   A*
004400871008   A************************
004500871007   A*  ITEM INQUIRY SCREEN
004600871008   A************************
004700871007   A          R ITMMNU                         TEXT('ITEM INQUIRY SCREEN ONE')
004800871008   A                                       2  2DFT('ENTER ITEM NUMBER')
004900871013   A            ITEMNO         6N 0I  2 20
005000871008   A                                      19  5DFT('CMD KEY 1 - END ')
005100871008   A                                      19 23DFT(' 2 - MAIN MENU ')
005200871008   A************************
005300871008   A*  ITEM DISPLAY
005400871008   A************************
005500871007   A          R ITMSC2                         TEXT('ITEM INQUIRY SCREEN TWO') OVE+
005600871007   A                                           RLAY
005700871007   A                                       4  2DFT('DESC-')
005800871007   A            DSC           30          4  8
005900871007   A                                       5  2DFT('QUANTITY AVAILABLE')
006000871007   A            QAVAIL         7N 0       5 25
006100871007   A                                       6 11DFT('ON HAND')
006200871007   A            QTYH           7N 0       6 25
006300871007   A                                       7 11DFT('ON ORDER')
006400871007   A            QTYO           7N 0       7 25
006500871007   A                                       8 11DFT('BACK ORDER')
006600871007   A            QTYB           7N 0       8 25
006700871007   A                                       9  2DFT('UNIT OF MEASURE')
006800871007   A            UNT            2          9 30
```

Figure 10-21 (Part 1 of 2). DDS for Source Program Multiple Session Inquiry Using DSPFIL

```
006900871007     A                                      10  2DFT('PRICE PER UNIT')
007000871007     A            PR1            7Y 2  10 24EDTCDE(3)
007100871007     A                                   11  8DFT('QUANTITY')
007200871007     A            PR5            7Y 0  11 25EDTCDE(3)
007300871007     A                                   12  8DFT('FREIGHT')
007400871007     A            UFR            5Y 2  12 26EDTCDE(3)
007500871008     A                                   13 32DFT('MORE... ')
007600871008     A                                   19  5DFT('CMD KEY 1 - END ')
007700871008     A                                   19 23DFT(' 2 - MAIN MENU ')
007800871008     A                                   19 40DFT(' 3 - ITEM MENU ')
007900871008     A*****************************
008000871008     A*  ITEM ADDITIONAL DISPLAY
008100871008     A*****************************
008200871007     A          R ITMSC3                      TEXT('ITEM INQUIRY SCREEN 3  ') OVE+
008300871007     A                                        RLAY
008400871007     A                                    5  2DFT('SALES MONTH')
008500871007     A            SLSM           9Y 2   5 16EDTCDE(1)
008600871007     A                                   6  8DFT('Y-T-D')
008700871007     A            SLSY          11Y 2   6 14EDTCDE(1)
008800871007     A                                   7  2DFT('COSTS MONTH')
008900871007     A            CSTM           9Y 2   7 16EDTCDE(1)
009000871007     A                                   8  8DFT('Y-T-D')
009100871007     A            CSTY          11Y 2   8 14EDTCDE(1)
009200871007     A                                   9  2DFT('PROFIT PCT')
009300871007     A            PROFIT         5Y 2   9 22EDTCDE(1)
009400871007     A                                  10  2DFT('LOST SALES')
009500871007     A            LOSTS          9Y 2  10 16EDTCDE(1)
009600871008     A                                  19  5DFT('CMD KEY 1 - END ')
009700871008     A                                  19 23DFT(' 2 - MAIN MENU ')
009800871008     A************************
009900871007     A*   TIMOUT SCREEN.
010000871008     A************************
010100871007     A          R TIMOUT                      TEXT('TIME OUT SCREEN')         OVE+
010200871007     A                                        RLAY
010300871007     A                                   20  2DFT('REMOTE SYSTEM TIMED OUT. ENTER
010400871007     A                                        1 TO TRY AGAIN OR 2 TO END.')
010500871007     A            TIMRSP         1   I 20 61
```

Figure 10-21 (Part 2 of 2). DDS for Source Program Multiple Session Inquiry Using DSPFIL

ICF File Creation and Program Device Entry Definition: The command needed to create the ICF file is:

```
CRTICFF  FILE(ICFLIB/CMNFIL)  SRCFILE(ICFLIB/QICFPUB)  SRCMBR(CMNFIL)
   ACQPGMDEV(*NONE)  MAXPGMDEV(4)  WAITRCD(30)
   TEXT("SOURCE ICF FILE FOR MULTIPLE SESSION PROGRAM")
```

The commands needed to define the four program device entries are:

```
OVRICFDEVE PGMDEV(ICF00) RMTLOCNAME(CHICAGO) FMTSLT(*RECID)

OVRICFDEVE PGMDEV(ICF01) RMTLOCNAME(NEWYORK) FMTSLT(*RECID)

OVRICFDEVE PGMDEV(ICF02) RMTLOCNAME(DETROIT) FMTSLT(*RECID)

OVRICFDEVE PGMDEV(ICF03) RMTLOCNAME(MADISON) FMTSLT(*RECID)
```

Program Explanation: The following explains the structure of the program examples illustrated in Figure 10-22 on page 10-49 and Figure 10-23 on page 10-63. The ICF file used in the first example is defined by the user, and uses externally described data formats (DDS). The second example uses the same file, but uses program-described data and system-supplied formats. The reference numbers in the explanation below correspond to the numbers in the following program examples.

In the following examples, the ICF file used in the first example is externally described, whereas the ICF file used in the second example is a program-described file.

Although the basic structure of the two examples provided is the same, there are differences because of the way the user-defined formats and the system-supplied formats are used. All output operations to the ICF file in the first example are done using the WRITE statement with the record format name coded as an operand. The output operations to the ICF file in the second example using system-supplied formats are issued with the system-supplied format coded as literal operand.

Differences between the first and second example are described in each of the following descriptions as necessary.

1 This section defines the ICF file (CMNFIL) and the display file (DSPFIL) used in the program.

CMNFIL is the ICF file used to send records to and receive records from each of the four target programs. CMNFIL is implemented with the file-level keyword, INDARA, indicating a separate indicator area is used.

DSPFIL is the display file used to receive user's requests and to report the information received based on the request.

The control area clause in the select statements of CMNFIL and DSPFIL is used to define the I/O feedback area. Information from the I/O feedback is used to determine the major/minor return code, record format, and command key pressed.

Note: In the program using system-supplied formats, the input records for CMNFIL are explicitly coded in the program since CMNFIL is now treated as a program-described file. The system-supplied file, QICDMF, could have been used instead of CMNFIL. Using the system-supplied file can be done by specifying QICDMF in the file specification, or by using an OVRICFF command to change the name from CMNFIL to QICDMF.

2 DSP-ERROR SECTION and CMN-ERROR SECTION define the error handling procedures for I/O errors on the DSPFIL and CMNFIL. A DSPFIL I/O error causes the program to end, and an error message to be sent to the printfile. The section for CMNFIL file I/O errors checks the major/minor return code to determine if the error is recoverable. If the error is recoverable (major code 83), it sets a flag (ERR-SW) to 1 and returns to the program. Furthermore, when major/minor code 3431 (input data truncated) is received, it is saved but not considered as an error, and takes an exit.

3 The program opens the files to be used and initializes the ICF file separate indicator area.

4 If the ERR-SW switch is set to 1, indicating that a recoverable error has occurred, the program determines whether the open retry count limit nine has been exceeded. If it has, the program goes to **19** and then ends. If the limit count is less than nine, one is added to the count and control passes to **17** and then to **3** to try to open the file.

5 The four program devices used by the program are explicitly acquired.

The device for the work station is implicitly acquired when the DSPFIL file is opened.

Also, the evoke requests are issued to the remote systems by passing control to 16.

When control returns from 16, the main menu (record format CIMENU) is then written to the work station.

6 A read operation is issued to the display device, and the program waits for an input request from the user. When a record is returned, the last record format used (as specified in the RCD-FMT field in the I/O control area) is checked. Based on the value in RCD-FMT, the program branches to the appropriate routine.

If a match is not found for the display record format, the main menu (CIMENU) is written to the work station and control is returned to 6.

7 This routine is called if the request is made from the main menu (CIMENU). If the CMD-KEY variable is set to '01', indicating that the operator pressed command key 1, the four transactions and sessions are ended and the program ends. If the operator entered option 1, the program writes the Item Inquiry menu (ITMMNU) to the work station and returns to 6.

If the option is not 1, the Customer Inquiry menu (DTLMNU) is written to the work station and control is passed to 6.

The rest of this chapter discusses the details of the control is passed to 6.

8 This routine is called when the user is requesting an item inquiry (record format ITMMNU). If command key 1 (CMD-KEY = '01') is pressed, control passes to 19, and then to 20, the four transactions end, and the program ends. If command key 2 is pressed, the inquiry request is canceled, the main menu (CMENU) is written to the work station, and the program returns to 16.

The item number read from the work station is checked for value range. If the range is from 0 to 399999, then the request is sent to the target program on program device ICF01.

If the range is from 400000 to 699999, the request is sent to the target program on program device ICF02.

If the range is from 700000 to 899999, the request is sent to the target program on program device ICF03.

The request is sent to the appropriate target program by writing data to the program device using format ITMREQ. The INVITE keyword is specified as part of the ITMREQ format to give the target program permission to send.

A read request is issued to the program device to receive the response to the inquiry.

The read is an implied read-from-invited-program-devices because no record format is specified in the read statement.

Control goes to 9 to process the item information based on the input data received and the result written to the screen using format ITMSC2.

After returning from 9, the program returns to 6.

Note: In the program using system-supplied formats, the $$SEND format is used as a literal instead of the user-defined ITMREQ record format name.

9 This routine is called when the target program responds to a request for an item record. If the returned item number is 0 or less, the request is invalid and a new Item Inquiry menu (ITMMNU) is written to the work station.

The program then performs the calculations to set the quantity fields and writes the result to the requesting work station using record format ITMSC2.

The program then returns to the calling routine.

10 This routine is called to process the next user request. If command key 1 (CMD-KEY = '01') is pressed, the transactions and session are ended 19, and control goes to 20 to end the program.

If command key 2 is pressed, the main menu (CMENU) is written to the work station. If command key 3 is pressed, the Item Inquiry menu is written to the work station, and the program returns to 6. By pressing Enter, the profit and loss figures are calculated and written to the work station before returning control to 6.

11 This routine calculates the profit and loss figures for the second screen of the requested item number.

12 This routine is called when a request is read from the Customer Inquiry menu (DTLMNU). If command key 1 (CMD-KEY = '01') is pressed, the transactions and sessions are ended. If command key 2 (CMD-KEY = '02') is pressed, the main menu (CIMENU) is written to the work station and the program returns to 6.

The customer inquiry request is sent to the target program by writing data to the program device ICF00 using format DTLREQ. The INVITE keyword is specified as part of the DLTREQ format to give the target program permission to send.

Control goes to 13 to retrieve the customer detail information.

Routine 14 is called to continue the customer information processing.

The program returns to 6.

Note: In the program using system-supplied formats, the write operation is issued with the $$SEND format specified as literal, in the user-supplied format, WRITE was issued with a record format name DTLREQ.

13 The information supplied by the target program in response to a request for a customer detail is processed in this routine. If the customer number is 0 or less, the request is invalid and the main menu (record format CIMENU) is written to the work station. The program then returns to 6.

Control goes to 15 to retrieve the customer detail information, and the result is written to the work station using record format DTLSCR.

The program then returns to 6.

14 This routine is called from 6, and handles the user's request following the display of the customer information. Command key 1 ends the job, command key 2 displays the main menu (CMENU), and pressing Enter displays the Customer Inquiry menu (DTLMNU). Then, control returns to 6.

15 This routine issues the read operation to the program device.

This read is an implied read-from-invited-program-devices because no record format is specified on the read statement.

A check is made of the MAJ-MIN return code for possible error conditions on a successful return (control is automatically passed to 2 for non-successful I/O operations). A 0310 major-minor return code means the remote system has timed out. (The wait time was specified on the CRTICFF command.) If no data was received (MAJ-MIN - 03xx), the request is sent again to the remote system. Finally, if the data returns in the wrong format, Control is passed to 17.

The customer information received from the target program is processed, and the result is written to the user work station using screen format DTLBLK.

Control returns to the calling routine.

Note: The program using system-supplied formats issues the READ statement with the file name CMNFIL specified in the operand without a record format name.

16 This routine builds the evoke requests to send to the remote systems. Because the DDS keyword for the record format only specifies the field identifiers with the record, this code moves the literal value CTDMULCL to the field *PGMID*, and ICFLIB to the field *LIB*.

When the program start request is received at the remote system, ICFLIB is searched for CTDMULCL and that program then starts. CTDMULCL is a CL program that contains the following statements:

```
ADDLIBLE ICFLIB
CALL ICFLIB/CTDMUL
```

Note: In the program using system-supplied formats, the evoke request is issued using the WRITE statement followed with a $$EVOKNI format coded as literal in the operand.

The library and program (ICFLIB/CTFMULCL) are specified as part of the $$EVOKNI format. CTFMULCL is a CL program that contains the following statements:

```
ADDLIBLE ICFLIB
CALL ICFLIB/CTFMUL
```

17 This routine ends the transactions and closes the files. The ERR-SW indicator is set again, and control returns to the calling routine.

18 This routine is executed when the program detects data in an incorrect record format. It writes an error message to the printfile, ends the program, and implicitly ends the session.

19 This routine issues the detach function to the ICF file for each of the four program devices. In the program using the system-supplied format, the write operation is issued using $$SENDET format, but in the program using the user-supplied format, it is the record format name DETACH.

20 This routine releases the program devices and close the files. The program ends.

```
5728CB1 R01 M02 881028              IBM AS/400  COBOL/400          ICFLIB/CSDMUL          03/21/89 08:20:30       Page    1
Program name . . . . . . . . . . . . : CSDMUL in ICFLIB
Source file  . . . . . . . . . . . . : QICFPUB in ICFLIB     Member - CSDMUL     03/21/89 08:16:34
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
User profile . . . . . . . . . . . . : *USER
Authority  . . . . . . . . . . . . . : *CHANGE
Text . . . . . . . . . . . . . . . . : CBL Multiple Session Inquiry - Source DDS
Compiler . . . . . . . . . . . . . . : IBM AS/400  COBOL/400
5728CB1 R01 M02 881028               COBOL SOURCE LISTING          ICFLIB/CSDMUL          03/21/89 08:20:30       Page    2
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
    1  000100 IDENTIFICATION DIVISION.                                                                 09/30/87
    2  000200 PROGRAM-ID.                CSDMUL.                                                       09/30/87
       000300*************************************************************************               09/30/87
       000400*  THIS PROGRAM ASSIGNS FOUR SESSIONS AS FOLLOWS:                      *                  09/30/87
       000500*    'ICF00' TO INQUIRE ABOUT A CUSTOMER ACCOUNT BEFORE AN             *                  09/30/87
       000600*            ORDER IS PROCESSED.                                       *                  09/30/87
       000700*    'ICF01' TO INQUIRE ABOUT THE INVENTORY STATUS OF AN ITEM          *                  09/30/87
       000800*            BEING ORDERED (ITEM 000001 THRU 399999).                  *                  09/30/87
       000900*    'ICF02' TO INQUIRE ABOUT THE INVENTORY STATUS OF AN ITEM          *                  09/30/87
       001000*            BEING ORDERED (ITEM 400000 THRU 699999).                  *                  09/30/87
       001100*    'ICF03' TO INQUIRE ABOUT THE INVENTORY STATUS OF AN ITEM          *                  09/30/87
       001200*            BEING ORDERED (ITEM 700000 THRU 999999).                  *                  09/30/87
       001300*  A DISPLAY DEVICE IS USED TO ENTER THE REQUEST ( USING A             *                  09/30/87
       001400*  CUSTOMER AND AN ITEM MENU) THAT IS SENT TO THE REMOTE               *                  09/30/87
       001500*  SYSTEM.                                                             *                  10/15/87
       001600*************************************************************************               09/30/87
    3  001700 ENVIRONMENT DIVISION.                                                                  09/30/87
    4  001800 CONFIGURATION SECTION.                                                                 09/30/87
    5  001900 SOURCE-COMPUTER.           IBM-AS400.                                                  01/15/88
    6  002000 OBJECT-COMPUTER.           IBM-AS400.                                                  01/15/88
    7  002100 SPECIAL-NAMES.             I-O-FEEDBACK IS IO-FEEDBACK                                 09/30/87
    8  002200                            OPEN-FEEDBACK IS OPEN-FBA.                                  09/30/87
    9  002300 INPUT-OUTPUT SECTION.                                                                  09/30/87
   10  002400 FILE-CONTROL.                                                                          09/30/87
       002500* 1                                                                                     09/30/87
       002600*************************************************************************               09/30/87
       002700*                                                                     *                  09/30/87
       002800*                F I L E   S P E C I F I C A T I O N S                *                  09/30/87
       002900*                                                                     *                  09/30/87
       003000*    CMNFIL :   ICF FILE USED TO SEND A REQUEST TO ONE                *                  01/15/88
       003100*               OF FOUR DIFFERENT TARGET PROGRAMS.  MULTIPLE          *                  09/30/87
       003200*               SESSIONS ARE ACTIVE CONCURRENTLY.                     *                  09/30/87
       003300*                                                                     *                  09/30/87
       003400*    DSPFIL :   DISPLAY FILE USED TO ENTER A REQUEST TO BE            *                  09/30/87
       003500*               SENT TO A REMOTE SYSTEM.                              *                  09/30/87
       003600*                                                                     *                  09/30/87
       003700*************************************************************************               09/30/87
   11  003800     SELECT CMNFIL ASSIGN TO WORKSTATION-CMNFIL-SI                                      10/13/87
   12  003900         ORGANIZATION IS TRANSACTION                                                    09/30/87
   13  004000         CONTROL-AREA IS TR-CTL-AREA                                                    09/30/87
   14  004100         FILE STATUS IS STATUS-IND MAJ-MIN.                                             09/30/87
   15  004200     SELECT DSPFIL ASSIGN TO WORKSTATION-DSPFIL                                         09/30/87
   16  004300         ORGANIZATION IS TRANSACTION                                                    09/30/87
   17  004400         CONTROL-AREA IS DISPLAY-FEEDBACK                                               09/30/87
   18  004500         FILE STATUS IS STATUS-DSP.                                                     09/30/87
   19  004600     SELECT QPRINT ASSIGN TO PRINTER-QSYSPRT.                                           09/30/87
   20  004700 DATA DIVISION.                                                                         09/30/87
   21  004800 FILE SECTION.                                                                          09/30/87
   22  004900 FD  CMNFIL                                                                             09/30/87
   23  005000     LABEL RECORDS ARE STANDARD.                                                        09/30/87
   24  005100 01  CMNREC.                                                                           09/30/87
```

Figure 10-22 (Part 1 of 14). Source Program Example — CSDMUL (User-Defined Formats)

```
  25  005200     COPY DDS-ALL-FORMATS-I-O OF CMNFIL.                                    02/28/89
  26 +000001       05  CMNFIL-RECORD PIC X(196).                                  <-ALL-FMTS
     +000002*    I-O FORMAT:ITMRSP     FROM FILE CMNFIL     OF LIBRARY ICFLIB     <-ALL-FMTS
     +000003*                                                                     <-ALL-FMTS
5728CB1 R01 M02 881028               COBOL SOURCE LISTING            ICFLIB/CSDMUL      03/21/89 08:20:30       Page     3
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
  27 +000004       05  ITMRSP        REDEFINES CMNFIL-RECORD.                    <-ALL-FMTS
  28 +000005           06 RECITM            PIC X(1).                           <-ALL-FMTS
  29 +000006           06 ITEMNO            PIC S9(6).                          <-ALL-FMTS
  30 +000007           06 DESC              PIC X(30).                          <-ALL-FMTS
  31 +000008           06 QTYLST            PIC S9(7).                          <-ALL-FMTS
  32 +000009           06 QTYOH             PIC S9(7).                          <-ALL-FMTS
  33 +000010           06 QTYOO             PIC S9(7).                          <-ALL-FMTS
  34 +000011           06 QTYBO             PIC S9(7).                          <-ALL-FMTS
  35 +000012           06 UNITQ             PIC X(2).                           <-ALL-FMTS
  36 +000013           06 PR01              PIC S9(5)V9(2).                     <-ALL-FMTS
  37 +000014           06 PR05              PIC S9(7).                          <-ALL-FMTS
  38 +000015           06 UFRT              PIC S9(3)V9(2).                     <-ALL-FMTS
  39 +000016           06 SLSTM             PIC S9(7)V9(2).                     <-ALL-FMTS
  40 +000017           06 SLSTY             PIC S9(9)V9(2).                     <-ALL-FMTS
  41 +000018           06 CSTTM             PIC S9(7)V9(2).                     <-ALL-FMTS
  42 +000019           06 CSTTY             PIC S9(9)V9(2).                     <-ALL-FMTS
  43 +000020           06 PRO               PIC S9(3)V9(2).                     <-ALL-FMTS
  44 +000021           06 LOS               PIC S9(7)V9(2).                     <-ALL-FMTS
  45 +000022           06 FILL1             PIC X(56).                          <-ALL-FMTS
     +000023*  INPUT FORMAT:DTLRSP     FROM FILE CMNFIL     OF LIBRARY ICFLIB     <-ALL-FMTS
     +000024*                                                                     <-ALL-FMTS
  46 +000025       05  DTLRSP-I      REDEFINES CMNFIL-RECORD.                    <-ALL-FMTS
  47 +000026           06 RECCUS            PIC X(1).                           <-ALL-FMTS
  48 +000027           06 CUSTNO            PIC S9(6).                          <-ALL-FMTS
  49 +000028           06 DNAME             PIC X(30).                          <-ALL-FMTS
  50 +000029           06 DLSTOR            PIC S9(6).                          <-ALL-FMTS
  51 +000030           06 DSLSTM            PIC S9(9).                          <-ALL-FMTS
  52 +000031           06 DSPM01            PIC S9(9).                          <-ALL-FMTS
  53 +000032           06 DSPM02            PIC S9(9).                          <-ALL-FMTS
  54 +000033           06 DSPM03            PIC S9(9).                          <-ALL-FMTS
  55 +000034           06 DSTTYD            PIC S9(11).                         <-ALL-FMTS
  56 +000035           06 IDEPT             PIC S9(3).                          <-ALL-FMTS
  57 +000036           06 FILL2             PIC X(57).                          <-ALL-FMTS
     +000037* OUTPUT FORMAT:DTLRSP     FROM FILE CMNFIL     OF LIBRARY ICFLIB     <-ALL-FMTS
     +000038*                                                                     <-ALL-FMTS
  58 +000039       05  DTLRSP-O      REDEFINES CMNFIL-RECORD.                    <-ALL-FMTS
  59 +000040           06 RECCUS            PIC X(1).                           <-ALL-FMTS
  60 +000041           06 CUSTNO            PIC S9(6).                          <-ALL-FMTS
  61 +000042           06 DNAME             PIC X(30).                          <-ALL-FMTS
  62 +000043           06 DLSTOR            PIC S9(6).                          <-ALL-FMTS
  63 +000044           06 DSLSTM            PIC S9(9).                          <-ALL-FMTS
  64 +000045           06 DSPM01            PIC S9(9).                          <-ALL-FMTS
  65 +000046           06 DSPM02            PIC S9(9).                          <-ALL-FMTS
  66 +000047           06 DSPM03            PIC S9(9).                          <-ALL-FMTS
  67 +000048           06 DSTTYD            PIC S9(11).                         <-ALL-FMTS
  68 +000049           06 IDEPT             PIC S9(3).                          <-ALL-FMTS
  69 +000050           06 FILL2             PIC X(57).                          <-ALL-FMTS
     +000051*    I-O FORMAT:DETACH     FROM FILE CMNFIL     OF LIBRARY ICFLIB     <-ALL-FMTS
     +000052*                                                                     <-ALL-FMTS
     +000053*       05  DETACH        REDEFINES CMNFIL-RECORD.                    <-ALL-FMTS
     +000054*    I-O FORMAT:EOS          FROM FILE CMNFIL     OF LIBRARY ICFLIB     <-ALL-FMTS
     +000055*                                                                     <-ALL-FMTS
     +000056*       05  EOS           REDEFINES CMNFIL-RECORD.                    <-ALL-FMTS
     +000057*  INPUT FORMAT:EVKREQ     FROM FILE CMNFIL     OF LIBRARY ICFLIB     <-ALL-FMTS
     +000058*                                                                     <-ALL-FMTS
5728CB1 R01 M02 881028               COBOL SOURCE LISTING            ICFLIB/CSDMUL      03/21/89 08:20:30       Page     4
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
     +000059*       05  EVKREQ-I      REDEFINES CMNFIL-RECORD.                    <-ALL-FMTS
     +000060* OUTPUT FORMAT:EVKREQ     FROM FILE CMNFIL     OF LIBRARY ICFLIB     <-ALL-FMTS
     +000061*                                                                     <-ALL-FMTS
  70 +000062       05  EVKREQ-O      REDEFINES CMNFIL-RECORD.                    <-ALL-FMTS
  71 +000063           06 PGMID             PIC X(10).                          <-ALL-FMTS
```

Figure 10-22 (Part 2 of 14). Source Program Example — CSDMUL (User-Defined Formats)

```
 72 +000064           06 LIB              PIC X(10).                                  <-ALL-FMTS
    +000065*    I-O FORMAT:ITMREQ     FROM FILE CMNFIL     OF LIBRARY ICFLIB         <-ALL-FMTS
    +000066*                                                                         <-ALL-FMTS
 73 +000067       05  ITMREQ        REDEFINES CMNFIL-RECORD.                        <-ALL-FMTS
 74 +000068           06 ITEMNO           PIC S9(6).                                  <-ALL-FMTS
    +000069*    I-O FORMAT:DTLREQ     FROM FILE CMNFIL     OF LIBRARY ICFLIB         <-ALL-FMTS
    +000070*                                                                         <-ALL-FMTS
 75 +000071       05  DTLREQ        REDEFINES CMNFIL-RECORD.                        <-ALL-FMTS
 76 +000072           06 CUSTNO           PIC S9(6).                                  <-ALL-FMTS
 77  005300 FD  DSPFIL                                                                        09/30/87
 78  005400     LABEL RECORDS ARE STANDARD.                                                   09/30/87
 79  005500 01  DSPREC.                                                                      09/30/87
 80  005600     COPY DDS-ALL-FORMATS-I-O OF DSPFIL.                                           02/27/89
 81 +000001       05  DSPFIL-RECORD PIC X(79).                                       <-ALL-FMTS
    +000002*  INPUT FORMAT:CIMENU     FROM FILE DSPFIL     OF LIBRARY ICFLIB         <-ALL-FMTS
    +000003*                          MENU FOR INQUIRY                               <-ALL-FMTS
 82 +000004       05  CIMENU-I      REDEFINES DSPFIL-RECORD.                        <-ALL-FMTS
 83 +000005           06 CIMENU-I-INDIC.                                            <-ALL-FMTS
 84 +000006                07 IN99        PIC 1  INDIC 99.                         <-ALL-FMTS
 85 +000007                07 IN98        PIC 1  INDIC 98.                         <-ALL-FMTS
 86 +000008                07 IN97        PIC 1  INDIC 97.                         <-ALL-FMTS
 87 +000009           06 OPTION           PIC X(1).                                   <-ALL-FMTS
    +000010* OUTPUT FORMAT:CIMENU     FROM FILE DSPFIL     OF LIBRARY ICFLIB         <-ALL-FMTS
    +000011*                          MENU FOR INQUIRY                               <-ALL-FMTS
    +000012*      05  CIMENU-O      REDEFINES DSPFIL-RECORD.                        <-ALL-FMTS
    +000013*  INPUT FORMAT:DTLMNU     FROM FILE DSPFIL     OF LIBRARY ICFLIB         <-ALL-FMTS
    +000014*                          CUSTOMER INQUIRY SCREEN 1                      <-ALL-FMTS
 88 +000015       05  DTLMNU-I      REDEFINES DSPFIL-RECORD.                        <-ALL-FMTS
 89 +000016           06 DTLMNU-I-INDIC.                                            <-ALL-FMTS
 90 +000017                07 IN99        PIC 1  INDIC 99.                         <-ALL-FMTS
 91 +000018                07 IN98        PIC 1  INDIC 98.                         <-ALL-FMTS
 92 +000019                07 IN97        PIC 1  INDIC 97.                         <-ALL-FMTS
 93 +000020           06 CUSTNO           PIC S9(6).                                  <-ALL-FMTS
    +000021* OUTPUT FORMAT:DTLMNU     FROM FILE DSPFIL     OF LIBRARY ICFLIB         <-ALL-FMTS
    +000022*                          CUSTOMER INQUIRY SCREEN 1                      <-ALL-FMTS
    +000023*      05  DTLMNU-O      REDEFINES DSPFIL-RECORD.                        <-ALL-FMTS
    +000024*  INPUT FORMAT:DTLSCR     FROM FILE DSPFIL     OF LIBRARY ICFLIB         <-ALL-FMTS
    +000025*                          CUSTOMER INQUIRY SCR. #2                       <-ALL-FMTS
 94 +000026       05  DTLSCR-I      REDEFINES DSPFIL-RECORD.                        <-ALL-FMTS
 95 +000027           06 DTLSCR-I-INDIC.                                            <-ALL-FMTS
 96 +000028                07 IN99        PIC 1  INDIC 99.                         <-ALL-FMTS
 97 +000029                07 IN98        PIC 1  INDIC 98.                         <-ALL-FMTS
 98 +000030                07 IN97        PIC 1  INDIC 97.                         <-ALL-FMTS
    +000031* OUTPUT FORMAT:DTLSCR     FROM FILE DSPFIL     OF LIBRARY ICFLIB         <-ALL-FMTS
    +000032*                          CUSTOMER INQUIRY SCR. #2                       <-ALL-FMTS
 99 +000033       05  DTLSCR-O      REDEFINES DSPFIL-RECORD.                        <-ALL-FMTS
100 +000034           06 CUSTN            PIC X(6).                                   <-ALL-FMTS
101 +000035           06 DEPT             PIC S9(3).                                  <-ALL-FMTS
102 +000036           06 DLSTR            PIC S9(6).                                  <-ALL-FMTS
103 +000037           06 DSLSM            PIC S9(9).                                  <-ALL-FMTS
5728CB1 R01 M02 881028               COBOL SOURCE LISTING            ICFLIB/CSDMUL        03/21/89 08:20:30       Page    5
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
104 +000038           06 DSPM1            PIC S9(9).                                  <-ALL-FMTS
105 +000039           06 DSPM2            PIC S9(9).                                  <-ALL-FMTS
106 +000040           06 DSPM3            PIC S9(9).                                  <-ALL-FMTS
107 +000041           06 DSTYD            PIC S9(11).                                 <-ALL-FMTS
108 +000042           06 CNAME            PIC X(5).                                   <-ALL-FMTS
    +000043*  INPUT FORMAT:ITMMNU     FROM FILE DSPFIL     OF LIBRARY ICFLIB         <-ALL-FMTS
    +000044*                          ITEM INQUIRY SCREEN ONE                        <-ALL-FMTS
109 +000045       05  ITMMNU-I      REDEFINES DSPFIL-RECORD.                        <-ALL-FMTS
110 +000046           06 ITMMNU-I-INDIC.                                            <-ALL-FMTS
111 +000047                07 IN99        PIC 1  INDIC 99.                         <-ALL-FMTS
112 +000048                07 IN98        PIC 1  INDIC 98.                         <-ALL-FMTS
113 +000049                07 IN97        PIC 1  INDIC 97.                         <-ALL-FMTS
114 +000050           06 ITEMNO           PIC S9(6).                                  <-ALL-FMTS
    +000051* OUTPUT FORMAT:ITMMNU     FROM FILE DSPFIL     OF LIBRARY ICFLIB         <-ALL-FMTS
    +000052*                          ITEM INQUIRY SCREEN ONE                        <-ALL-FMTS
    +000053*      05  ITMMNU-O      REDEFINES DSPFIL-RECORD.                        <-ALL-FMTS
```

Figure 10-22 (Part 3 of 14). Source Program Example — CSDMUL (User-Defined Formats)

```
    +000054*  INPUT FORMAT:ITMSC2     FROM FILE DSPFIL     OF LIBRARY ICFLIB           <-ALL-FMTS
    +000055*                          ITEM INQUIRY SCREEN TWO                          <-ALL-FMTS
  115 +000056       05  ITMSC2-I      REDEFINES DSPFIL-RECORD.                        <-ALL-FMTS
  116 +000057           06 ITMSC2-I-INDIC.                                            <-ALL-FMTS
  117 +000058                07 IN99         PIC 1  INDIC 99.                        <-ALL-FMTS
  118 +000059                07 IN98         PIC 1  INDIC 98.                        <-ALL-FMTS
  119 +000060                07 IN97         PIC 1  INDIC 97.                        <-ALL-FMTS
    +000061* OUTPUT FORMAT:ITMSC2     FROM FILE DSPFIL     OF LIBRARY ICFLIB           <-ALL-FMTS
    +000062*                          ITEM INQUIRY SCREEN TWO                          <-ALL-FMTS
  120 +000063       05  ITMSC2-O      REDEFINES DSPFIL-RECORD.                        <-ALL-FMTS
  121 +000064           06 DSC              PIC X(30).                                <-ALL-FMTS
  122 +000065           06 QAVAIL            PIC S9(7).                                <-ALL-FMTS
  123 +000066           06 QTYH              PIC S9(7).                                <-ALL-FMTS
  124 +000067           06 QTYO              PIC S9(7).                                <-ALL-FMTS
  125 +000068           06 QTYB              PIC S9(7).                                <-ALL-FMTS
  126 +000069           06 UNT               PIC X(2).                                 <-ALL-FMTS
  127 +000070           06 PR1               PIC S9(5)V9(2).                           <-ALL-FMTS
  128 +000071           06 PR5               PIC S9(7).                                <-ALL-FMTS
  129 +000072           06 UFR               PIC S9(3)V9(2).                           <-ALL-FMTS
    +000073*  INPUT FORMAT:ITMSC3     FROM FILE DSPFIL     OF LIBRARY ICFLIB           <-ALL-FMTS
    +000074*                          ITEM INQUIRY SCREEN 3                            <-ALL-FMTS
  130 +000075       05  ITMSC3-I      REDEFINES DSPFIL-RECORD.                        <-ALL-FMTS
  131 +000076           06 ITMSC3-I-INDIC.                                            <-ALL-FMTS
  132 +000077                07 IN99         PIC 1  INDIC 99.                        <-ALL-FMTS
  133 +000078                07 IN98         PIC 1  INDIC 98.                        <-ALL-FMTS
  134 +000079                07 IN97         PIC 1  INDIC 97.                        <-ALL-FMTS
    +000080* OUTPUT FORMAT:ITMSC3     FROM FILE DSPFIL     OF LIBRARY ICFLIB           <-ALL-FMTS
    +000081*                          ITEM INQUIRY SCREEN 3                            <-ALL-FMTS
  135 +000082       05  ITMSC3-O      REDEFINES DSPFIL-RECORD.                        <-ALL-FMTS
  136 +000083           06 SLSM             PIC S9(7)V9(2).                           <-ALL-FMTS
  137 +000084           06 SLSY             PIC S9(9)V9(2).                           <-ALL-FMTS
  138 +000085           06 CSTM             PIC S9(7)V9(2).                           <-ALL-FMTS
  139 +000086           06 CSTY             PIC S9(9)V9(2).                           <-ALL-FMTS
  140 +000087           06 PROFIT            PIC S9(3)V9(2).                           <-ALL-FMTS
  141 +000088           06 LOSTS             PIC S9(7)V9(2).                           <-ALL-FMTS
    +000089*  INPUT FORMAT:TIMOUT     FROM FILE DSPFIL     OF LIBRARY ICFLIB           <-ALL-FMTS
    +000090*                          TIME OUT SCREEN                                  <-ALL-FMTS
  142 +000091       05  TIMOUT-I      REDEFINES DSPFIL-RECORD.                        <-ALL-FMTS
  143 +000092           06 TIMOUT-I-INDIC.                                            <-ALL-FMTS
5728CB1 R01 M02 881028              COBOL SOURCE LISTING            ICFLIB/CSDMUL      03/21/89 08:20:30       Page     6
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
  144 +000093               07 IN99         PIC 1  INDIC 99.                        <-ALL-FMTS
  145 +000094               07 IN98         PIC 1  INDIC 98.                        <-ALL-FMTS
  146 +000095               07 IN97         PIC 1  INDIC 97.                        <-ALL-FMTS
  147 +000096           06 TIMRSP           PIC X(1).                                 <-ALL-FMTS
    +000097* OUTPUT FORMAT:TIMOUT     FROM FILE DSPFIL     OF LIBRARY ICFLIB           <-ALL-FMTS
    +000098*                          TIME OUT SCREEN                                  <-ALL-FMTS
    +000099*      05  TIMOUT-O      REDEFINES DSPFIL-RECORD.                        <-ALL-FMTS
  148  005700 FD  QPRINT                                                                    09/30/87
  149  005800     LABEL RECORDS ARE OMITTED.                                                09/30/87
  150  005900 01  PRINTREC.                                                                 01/14/88
  151  006000     05  RC                          PIC 9999.                                 01/15/88
  152  006100     05  ERRMSG                      PIC X(128).                               01/14/88
  153  006200 WORKING-STORAGE SECTION.                                                      09/30/87
  154  006300 77  STATUS-IND                      PIC X(2).                                 09/30/87
  155  006400 77  STATUS-DSP                      PIC X(2).                                 09/30/87
  156  006500 77  MAJ-MIN-SAV                     PIC X(4).                                 09/30/87
  157  006600 77  EOF-PFILE-SW                    PIC X VALUE "0".                          09/30/87
  158  006700 77  ERR-SW                          PIC X VALUE "0".                          09/30/87
  159  006800 77  INDON                           PIC 1 VALUE B"1".                         09/30/87
  160  006900 77  INDOFF                          PIC 1 VALUE B"0".                         09/30/87
  161  007000 77  OPEN-COUNT                      PIC 9(1) VALUE 0.                         09/30/87
  162  007100 77  LEN                             PIC 9(10)V9(5) COMP.                      09/30/87
  163  007200 77  PROFM                           PIC 9(7)V9(2) COMP-4.                     09/30/87
  164  007300 77  CMD2                            PIC X(31)                                 09/30/87
  165  007400       VALUE "CPYF HEXDUMP *LIST PRTFMT(*HEX)".                                09/30/87
  166  007500 01  SUBKEY-VALUE.                                                            09/30/87
  167  007600     05  SUBKEY                      PIC 9(3) VALUE 0.                         09/30/87
```

Figure 10-22 (Part 4 of 14). Source Program Example — CSDMUL (User-Defined Formats)

```
  168  007700 01  TR-CTL-AREA.                                                  09/30/87
  169  007800     05  FILLER                    PIC X(2).                      09/30/87
  170  007900     05  PGM-DEV-NME               PIC X(10).                     09/30/87
  171  008000     05  RCD-FMT-NME               PIC X(10).                     09/30/87
  172  008100 01  CMNF-INDIC-AREA.                                             09/30/87
  173  008200     05  IN90                      PIC 1 INDIC 90.                09/30/87
  174  008300         88 IN90-ON                     VALUE B"1".               09/30/87
  175  008400         88 IN90-OFF                    VALUE B"0".               09/30/87
  176  008500 01  DSPF-INDIC-AREA.                                            09/30/87
  177  008600     05  IN23                      PIC 1 INDIC 23.                09/30/87
  178  008700         88 IN23-ON                   VALUE B"1".                 09/30/87
  179  008800         88 IN23-OFF                  VALUE B"0".                 09/30/87
  180  008900     05  IN97                      PIC 1 INDIC 97.                09/30/87
  181  009000         88 IN97-ON                   VALUE B"1".                 09/30/87
  182  009100         88 IN97-OFF                  VALUE B"0".                 09/30/87
  183  009200     05  IN98                      PIC 1 INDIC 98.                09/30/87
  184  009300         88 IN98-ON                   VALUE B"1".                 09/30/87
  185  009400         88 IN98-OFF                  VALUE B"0".                 09/30/87
  186  009500     05  IN99                      PIC 1 INDIC 99.                09/30/87
  187  009600         88 IN99-ON                   VALUE B"1".                 09/30/87
  188  009700         88 IN99-OFF                  VALUE B"0".                 09/30/87
  189  009800 01  MAJ-MIN.                                                     09/30/87
  190  009900     05  MAJ                       PIC X(2).                      09/30/87
  191  010000     05  MIN                       PIC X(2).                      09/30/87
  192  010100 01  DISPLAY-FEEDBACK.                                           09/30/87
  193  010200     05  CMD-KEY                   PIC X(2).                      09/30/87
  194  010300     05  FILLER                    PIC X(10).                     09/30/87
  195  010400     05  RCD-FMT                   PIC X(10).                     09/30/87
5728CB1 R01 M02 881028              COBOL SOURCE LISTING          ICFLIB/CSDMUL          03/21/89 08:20:30       Page     7
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
      010500/                                                                 09/30/87
  196 010600 PROCEDURE DIVISION.                                              09/30/87
      010700 DECLARATIVES.                                                    09/30/87
      010800* 2                                                              10/14/87
      010900*************************************************************     02/21/89
      011000*                                                           *     02/21/89
      011100* AN ERROR ON THE DISPLAY FILE - DSPFIL - MAKES IT INACTIVE *     02/21/89
      011200* THE JOB IS ENDED.                                         *     02/21/89
      011300*                                                           *     02/21/89
      011400*************************************************************     02/21/89
      011500 DSP-ERROR SECTION.                                               10/05/87
      011600     USE AFTER STANDARD ERROR PROCEDURE ON DSPFIL.                 10/05/87
      011700*                                                                 10/05/87
      011800 DSPFIL-EXCEPTION.                                                10/05/87
  197 011900     MOVE "DISPLAY ERROR. JOB TERMINATED" TO ERRMSG.              02/21/89
  198 012000     WRITE PRINTREC.                                              02/21/89
  199 012100     CLOSE CMNFIL DSPFIL QPRINT.                                  02/21/89
  200 012200     STOP RUN.                                                    02/21/89
      012300*                                                                 10/13/87
      012400*************************************************************     02/21/89
      012500*                                                           *     02/21/89
      012600* THIS SECTION HANDLES ERRORS ON THE CMNFIL. A PERMANENT    *     02/21/89
      012700* SESSION ERROR WILL END THE JOB.                           *     02/21/89
      012800*                                                           *     02/21/89
      012900*************************************************************     02/21/89
      013000 CMN-ERROR SECTION.                                               10/14/87
      013100     USE AFTER STANDARD ERROR PROCEDURE ON CMNFIL.                 09/30/87
      013200 CMNFIL-EXCEPTION.                                                09/30/87
      013300*************************************************************     02/21/89
      013400* CHECK THE MAJOR/MINOR CODES AND TAKE APPROPRIATE ACTION   *     02/21/89
      013500* MAJOR CODE 34 - INPUT EXCEPTION.                          *     02/21/89
      013600*************************************************************     02/21/89
  201 013700     IF MAJ-MIN = "3431"                                          09/30/87
      013800* DATA TRUNCATED IN INPUT AREA. SAVE RETURN CODE.                 02/22/89
  202 013900           MOVE MAJ-MIN TO MAJ-MIN-SAV                            09/30/87
  203 014000           GO TO EXIT-DECLARATIVES.                               10/13/87
      014100* RECOVERABLE SESSION ERROR.  CLOSE ICF FILE.                     01/21/88
  204 014200     IF MAJ = "83"                                                09/30/87
```

Figure 10-22 (Part 5 of 14). Source Program Example — CSDMUL (User-Defined Formats)

```
  205  014300        MOVE MAJ-MIN TO RC                                                  01/14/88
  206  014400        MOVE "PROGRAM STARTED AGAIN DUE TO SESSION ERROR"                   09/30/87
       014500        TO ERRMSG                                                           01/14/88
  207  014600        WRITE PRINTREC                                                      09/30/87
  208  014700        MOVE "1" TO ERR-SW                                                  09/30/87
  209  014800        GO TO EXIT-DECLARATIVES.                                            09/30/87
       014900*                                                                          09/30/87
       015000*************************************************************************  02/21/89
       015100* WHEN THERE IS A PERMANENT SESSION ERROR DETECTED,              *          02/21/89
       015200* THE MAJOR-MINOR CODE IS PLACED INTO A DATA BASE                *          02/21/89
       015300* FILE AND THE FILE IS PRINTED IN HEX USING COPYFILE.            *          02/21/89
       015400*************************************************************************  02/21/89
       015500*                                                                          09/30/87
       015600 GETFBA.                                                                    09/30/87
  210  015700     MOVE MAJ-MIN TO RC.                                                   01/14/88
  211  015800     MOVE "PROGRAM TERMINATED DUE TO ERROR IN CMNFIL FILE"                 09/30/87
       015900        TO ERRMSG.                                                         01/14/88
5728CB1 R01 M02 881028              COBOL SOURCE LISTING            ICFLIB/CSDMUL          03/21/89 08:20:30        Page     8
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
  212  016000     WRITE PRINTREC.                                                       09/30/87
  213  016100     CLOSE  CMNFIL DSPFIL QPRINT.                                          09/30/87
  214  016200     STOP RUN.                                                             09/30/87
       016300*                                                                          10/02/87
       016400 EXIT-DECLARATIVES.                                                        09/30/87
       016500     EXIT.                                                                 02/28/89
       016600*                                                                          09/30/87
  215  016700 END DECLARATIVES.                                                        09/30/87
5728CB1 R01 M02 881028              COBOL SOURCE LISTING            ICFLIB/CSDMUL          03/21/89 08:20:30        Page     9
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
       016800/                                                                          09/30/87
       016900 START-PROGRAM  SECTION.                                                   09/30/87
       017000*                                                                          09/30/87
       017100 START-PROGRAM-PARAGRAPH.                                                  09/30/87
       017200* 3                                                                       09/30/87
  216  017300     OPEN I-O   CMNFIL DSPFIL                                              09/30/87
       017400          OUTPUT QPRINT.                                                  09/30/87
  217  017500     MOVE ZEROS TO CMNF-INDIC-AREA.                                        09/30/87
       017600*                                                                          09/30/87
       017700*************************************************************************  09/30/87
       017800* THE FOLLOWING TEST IS TO ATTEMPT RECOVERY IF AN ERROR          *          09/30/87
       017900* OCCURS WHEN OPENING THE ICF FILE.                              *          01/15/88
       018000*************************************************************************  09/30/87
       018100* 4                                                                       09/30/87
  218  018200     IF ERR-SW = "1"                                                       09/30/87
  219  018300        THEN IF OPEN-COUNT IS = 9                                          09/30/87
  220  018400                THEN PERFORM DETACH-ROUTINE THRU DETACH-EXIT              09/30/87
  221  018500                     GO TO END-JOB                                        09/30/87
       018600             ELSE                                                         09/30/87
  222  018700                ADD 1 TO OPEN-COUNT                                       09/30/87
       018800                PERFORM ERROR-RECOVERY                                    09/30/87
  223  018900                GO TO START-PROGRAM-PARAGRAPH                             09/30/87
       019000     ELSE                                                                 09/30/87
  225  019100        MOVE 0 TO OPEN-COUNT.                                             09/30/87
       019200*                                                                          09/30/87
       019300*************************************************************************  09/30/87
       019400*                                                                *          09/30/87
       019500*   THE DISPLAY DEVICE IS IMPLICITLY ACQUIRED WHEN THE           *          10/15/87
       019600*   FILE IS OPENED.                                              *          09/30/87
       019700*                                                                *          09/30/87
       019800*   ALL OF THE ICF PROGRAM DEVICES ARE EXPLICITLY ACQUIRED.      *          02/21/89
       019900*                                                                *          09/30/87
       020000*   EACH OF THE FOUR TARGET PROGRAMS ARE EVOKED TO ESTABLISH     *          09/30/87
       020100*   TRANSACTIONS WITH THE REMOTE SYSTEMS.                        *          09/30/87
       020200*                                                                *          09/30/87
       020300*   THE MAIN INQUIRY MENU (CIMENU) IS WRITTEN TO THE USER'S      *          09/30/87
       020400*   DISPLAY.                                                     *          09/30/87
       020500*                                                                *          09/30/87
       020600*   EVOKE PROGRAM "CTDMUL" ON REMOTE SYSTEM IN LIBRARY ICFLIB.   *          10/13/87
```

Figure 10-22 (Part 6 of 14). Source Program Example — CSDMUL (User-Defined Formats)

```
      020700*****************************************************************          09/30/87
      020800* 5                                                                       09/30/87
  226 020900     ACQUIRE "ICF00   " FOR CMNFIL.                                       09/30/87
  227 021000     ACQUIRE "ICF01   " FOR CMNFIL.                                       09/30/87
  228 021100     ACQUIRE "ICF02   " FOR CMNFIL.                                       09/30/87
  229 021200     ACQUIRE "ICF03   " FOR CMNFIL.                                       09/30/87
  230 021300     PERFORM EVOKE-ROUTINE THRU EVOKE-EXIT.                               09/30/87
      021400*                                                                        09/30/87
  231 021500     WRITE DSPREC FORMAT IS "CIMENU"                                      09/30/87
      021600        INDICATORS ARE DSPF-INDIC-AREA.                                   09/30/87
      021700*                                                                        10/14/87
      021800*****************************************************************          09/30/87
      021900*                                                             *          09/30/87
      022000*                    DETERMINE USER'S REQUEST                 *          09/30/87
      022100*                                                             *          09/30/87
      022200*   A READ TO THE DISPLAY DEVICE IS ISSUED TO RECEIVE         *          10/15/87
5728CB1 R01 M02 881028              COBOL SOURCE LISTING          ICFLIB/CSDMUL          03/21/89 08:20:30       Page   10
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
      022300*   THE USER'S REQUEST. THE TYPE OF REQUEST MADE IS BASED ON THE   *          10/13/87
      022400*   DISPLAY FORMAT CURRENTLY ON THE SCREEN.  THE RECORD FORMAT     *          10/13/87
      022500*   NAME IS EXTRACTED FROM THE I/O FEEDBACK AREA FOR THE DISPLAY   *          10/13/87
      022600*   FILE AND USED TO DETERMINE WHAT ACTION SHOULD BE TAKEN NEXT.   *          10/13/87
      022700*                                                             *          09/30/87
      022800*****************************************************************          09/30/87
      022900* 6                                                                       09/30/87
      023000 READRQ.                                                                  09/30/87
  232 023100     READ DSPFIL INDICATORS ARE DSPF-INDIC-AREA.                          09/30/87
  233 023200     IF RCD-FMT = "CIMENU"                                                09/30/87
  234 023300         PERFORM MENU-ROUTINE THRU MENU-EXIT                              09/30/87
  235 023400         GO TO READRQ.                                                    09/30/87
  236 023500     IF RCD-FMT = "ITMMNU"                                                09/30/87
  237 023600         PERFORM ITMIN-ROUTINE THRU ITMIN-EXIT                            09/30/87
  238 023700         GO TO READRQ.                                                    09/30/87
  239 023800     IF RCD-FMT = "ITMSC2"                                                09/30/87
  240 023900         PERFORM ITMRTN-ROUTINE THRU ITMRTN-EXIT                          09/30/87
  241 024000         GO TO READRQ.                                                    09/30/87
  242 024100     IF RCD-FMT = "ITMSC3"                                                09/30/87
  243 024200         PERFORM ITMRTN-ROUTINE THRU ITMRTN-EXIT                          09/30/87
  244 024300         GO TO READRQ.                                                    09/30/87
  245 024400     IF RCD-FMT = "DTLMNU"                                                09/30/87
  246 024500         PERFORM DTLIN-ROUTINE THRU DTLIN-EXIT                            09/30/87
  247 024600         GO TO READRQ.                                                    09/30/87
  248 024700     IF RCD-FMT = "DTLSCR"                                                10/12/87
  249 024800         PERFORM DTLRTN-ROUTINE THRU DTLRTN-EXIT                          10/12/87
  250 024900         GO TO READRQ.                                                    10/12/87
  251 025000     WRITE DSPREC FORMAT IS "CIMENU".                                     09/30/87
      025100                                                                         10/12/87
  252 025200     GO TO READRQ.                                                        09/30/87
5728CB1 R01 M02 881028              COBOL SOURCE LISTING          ICFLIB/CSDMUL          03/21/89 08:20:30       Page   11
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
      025300/                                                                        09/30/87
      025400*****************************************************************          09/30/87
      025500*                                                             *          09/30/87
      025600*                          MAIN MENU                          *          09/30/87
      025700*                                                             *          09/30/87
      025800*    THE MAIN MENU IS READ TO DETERMINE THE REQUEST ENTERED    *          10/12/87
      025900*    BY THE USER. IF CMD 1 (*IN99) IS PRESSED, THE PROGRAM     *          10/12/87
      026000*    IS ENDED.  IF OPTION = 1, AN ITEM INQUIRY MENU IS WRITTEN TO *          10/12/87
      026100*    TO SCREEN.  IF OPTION = 2, A CUSTOMER INQUIRY MENU IS     *          10/12/87
      026200*    WRITTEN TO THE SCREEN.                                    *          10/12/87
      026300*                                                             *          09/30/87
      026400*****************************************************************          09/30/87
      026500* 7                                                                       09/30/87
      026600 MENU-ROUTINE.                                                            09/30/87
  253 026700     IF CMD-KEY = "01"                                                    09/30/87
  254 026800        PERFORM DETACH-ROUTINE THRU DETACH-EXIT                           09/30/87
  255 026900        GO TO END-JOB.                                                    09/30/87
  256 027000     IF OPTION = "1"                                                      09/30/87
```

Figure 10-22 (Part 7 of 14). Source Program Example — CSDMUL (User-Defined Formats)

```
  257  027100        WRITE DSPREC FORMAT IS "ITMMNU"                                  09/30/87
       027200     ELSE                                                                09/30/87
  258  027300        WRITE DSPREC FORMAT IS "DTLMNU".                                 09/30/87
       027400 MENU-EXIT.                                                             09/30/87
       027500     EXIT.                                                              09/30/87
5728CB1 R01 M02 881028              COBOL SOURCE LISTING          ICFLIB/CSDMUL      03/21/89 08:20:30      Page   12
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME  CHG/DATE
       027600/                                                                       09/30/87
       027700*************************************************************          09/30/87
       027800*                                                           *          09/30/87
       027900*                        ITEM INQUIRY                       *          09/30/87
       028000*                                                           *          09/30/87
       028100*    THE ITEM NUMBER REQUESTED BY THE USER ON THE ITEM INQUIRY *          09/30/87
       028200*    SCREEN IS CHECKED. THIS IS DETERMINED BY THE              *          09/30/87
       028300*    DISPLAY RECORD FORMAT BEING PROCESSED - IN THIS CASE ITMMNU. *       09/30/87
       028400*                                                           *          02/21/89
       028500*    IF CMD KEY 1 IS PRESSED, THE PROGRAM IS ENDED.  IF CMD KEY 2 *       10/13/87
       028600*    IS PRESSED, THE ITEM INQUIRY REQUEST IS CANCELED, AND THE *          09/30/87
       028700*    MAIN MENU (CIMENU) IS WRITTEN TO THE SCREEN.              *          09/30/87
       028800*                                                           *          09/30/87
       028900*    IF AN ITEM NUMBER IS ENTERED, A ITEM INQUIRY REQUEST IS  *          09/30/87
       029000*    SENT TO THE APPROPRIATE REMOTE SYSTEM.  THE REMOTE SYSTEM *          09/30/87
       029100*    IS SELECTED BASED ON THE ITEM NUMBER REQUESTED.           *          09/30/87
       029200*                                                           *          09/30/87
       029300*    A CHECK IS MADE FOR THREE CONDITIONS FOLLOWING THE READ.  *          10/14/87
       029400*    1) THE REMOTE SYSTEM TIMED OUT, 2) NO DATA RECEIVED, AND  *          10/14/87
       029500*    3) DATA RETURNED IN AN UNEXPECTED RECORD FORMAT.          *          10/14/87
       029600*                                                           *          10/14/87
       029700*    IF THE REMOTE SYSTEM TIMES OUT (MAJ-MIN = 0310) A MESSAGE *          10/14/87
       029800*    IS WRITTEN TO THE SCREEN, ASKING TO TRY AGAIN OR END THE  *          10/14/87
       029900*    PROGRAM.                                                  *          10/14/87
       030000*                                                           *          10/14/87
       030100*    IF NO DATA IS RECEIVED AFTER THE READ OPERATION TO THE    *          10/15/87
       030200*    PROGRAM DEVICE (MAJ-MIN = 03__ ) THE REQUEST IS SENT AGAIN *         10/14/87
       030300*    TO THE REMOTE SYSTEM AND THE READ OPERATION IS ISSUED TO  *          10/14/87
       030400*    THE THE PROGRAM DEVICE.                                   *          10/15/87
       030500*                                                           *          10/14/87
       030600*    IF THE RECORD RETURNS WITH THE WRONG RECORD FORMAT, THE   *          10/14/87
       030700*    PROGRAM WILL GO TO EXIT-FORMAT-ERR ROUTINE.               *          10/14/87
       030800*                                                           *          10/14/87
       030900*************************************************************          09/30/87
       031000* 8                                                                    09/30/87
  259  031100 ITMIN-ROUTINE.                                                        09/30/87
  260  031200     IF CMD-KEY = "01"                                                 09/30/87
  261  031300        PERFORM DETACH-ROUTINE THRU DETACH-EXIT                        10/12/87
  262  031400        GO TO END-JOB.                                                 10/12/87
  263  031500     IF CMD-KEY = "02"                                                 10/12/87
  264  031600        WRITE DSPREC FORMAT IS "CIMENU"                                10/12/87
  265  031700        GO TO ITMIN-EXIT.                                              10/12/87
  266  031800     MOVE CORR ITMMNU-I TO ITMREQ.                                     09/30/87
  267  031900     IF ITEMNO OF ITMMNU-I LESS THAN 399999 GO TO XICF01.              09/30/87
  269  032000     IF ITEMNO OF ITMMNU-I LESS THAN 699999 GO TO XICF02.              09/30/87
  271  032100     IF ITEMNO OF ITMMNU-I LESS THAN 899999 GO TO XICF03.              09/30/87
       032200 XICF01.                                                               09/30/87
  273  032300     MOVE "ICF01   " TO PGM-DEV-NME.                                   09/30/87
  274  032400     GO TO XITMIN.                                                     09/30/87
       032500 XICF02.                                                               09/30/87
  275  032600     MOVE "ICF02   " TO PGM-DEV-NME.                                   09/30/87
  276  032700     GO TO XITMIN.                                                     09/30/87
       032800 XICF03.                                                               09/30/87
  277  032900     MOVE "ICF03   " TO PGM-DEV-NME.                                   09/30/87
       033000 XITMIN.                                                               09/30/87
5728CB1 R01 M02 881028              COBOL SOURCE LISTING          ICFLIB/CSDMUL      03/21/89 08:20:30      Page   13
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME  CHG/DATE
  278  033100     MOVE ZEROS TO CMNF-INDIC-AREA.                                    09/30/87
  279  033200     WRITE CMNREC FORMAT IS "ITMREQ"                                   09/30/87
       033300        TERMINAL IS PGM-DEV-NME.                                       09/30/87
       033400 TRY-AGAIN.                                                            10/01/87
```

Figure 10-22 (Part 8 of 14). Source Program Example — CSDMUL (User-Defined Formats)

```
  280  033500     READ CMNFIL.                                                                  09/30/87
  281  033600     IF MAJ-MIN = "0310"                                                           10/01/87
  282  033700        WRITE DSPREC FORMAT IS "TIMOUT"                                            09/30/87
  283  033800        READ DSPFIL INDICATORS ARE DSPF-INDIC-AREA                                 09/30/87
  284  033900        IF TIMRSP = "1" GO TO TRY-AGAIN END-IF                                     01/21/88
  286  034000        IF TIMRSP = "2" GO TO END-JOB END-IF.                                      01/21/88
  288  034100     IF MAJ = "03"                                                                 09/30/87
  289  034200        GO TO XITMIN.                                                              09/30/87
  290  034300     IF RCD-FMT-NME IS NOT EQUAL "ITMRSP" GO TO EXIT-FORMAT-ERR.                   10/02/87
  292  034400     PERFORM ITMOUT-ROUTINE THRU ITMOUT-EXIT.                                      09/30/87
       034500 ITMIN-EXIT.                                                                      09/30/87
       034600     EXIT.                                                                       09/30/87
5728CB1 R01 M02 881028              COBOL SOURCE LISTING            ICFLIB/CSDMUL          03/21/89 08:20:30      Page   14
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
       034700/                                                                               10/14/87
       034800*************************************************************************      09/30/87
       034900*                                                                       *      09/30/87
       035000*                     PROCESS ITEM INFORMATION                          *      09/30/87
       035100*                                                                       *      09/30/87
       035200*    THE ITEM RECORD RECEIVED FROM THE TARGET PROGRAM AND THE           *      09/30/87
       035300*    INFORMATION ABOUT THE ITEM IS PROCESSED AND DISPLAYED.             *      09/30/87
       035400*    IF ITEMNO IS 0 OR LESS, IT IS AN INVALID REQUEST AND A FRESH       *      09/30/87
       035500*    ITEM MENU IS WRITTEN TO THE SCREEN.  IF THE REQUEST IS             *      09/30/87
       035600*    VALID, VALUES ARE CALCULATED BASED ON THE INFORMATION              *      09/30/87
       035700*    RECEIVED.                                                          *      09/30/87
       035800*                                                                       *      09/30/87
       035900*************************************************************************      09/30/87
       036000* 9                                                                           09/30/87
  293  036100 ITMOUT-ROUTINE.                                                                09/30/87
  294  036200     IF ITEMNO OF ITMRSP NOT GREATER THAN 0                                     09/30/87
  295  036300        WRITE DSPREC FORMAT IS "ITMMNU"                                         09/30/87
  296  036400        GO TO ITMOUT-EXIT.                                                      09/30/87
  297  036500     MOVE DESC TO DSC OF ITMSC2-O.                                              09/30/87
  298  036600     MOVE QTYLST TO QAVAIL OF ITMSC2-O.                                         09/30/87
  299  036700     MOVE QTYOO TO QTYO OF ITMSC2-O.                                            09/30/87
  300  036800     MOVE QTYOH TO QTYH OF ITMSC2-O.                                            09/30/87
  301  036900     MOVE QTYBO TO QTYB OF ITMSC2-O.                                            09/30/87
  302  037000     MOVE UNITQ TO UNT OF ITMSC2-O.                                             09/30/87
  303  037100     MOVE PR01 TO PR1 OF ITMSC2-O.                                              09/30/87
  304  037200     MOVE PR05 TO PR5 OF ITMSC2-O.                                              09/30/87
  305  037300     MOVE UFRT TO UFR OF ITMSC2-O.                                              09/30/87
  306  037400     WRITE DSPREC FORMAT IS "ITMSC2"                                            09/30/87
       037500        INDICATORS ARE DSPF-INDIC-AREA.                                         09/30/87
       037600 ITMOUT-EXIT.                                                                   09/30/87
       037700     EXIT.                                                                      09/30/87
       037800*                                                                             10/14/87
       037900*************************************************************************      02/21/89
       038000*                                                                       *      02/21/89
       038100*                   ADDITIONAL ITEM INFORMATION                         *      02/21/89
       038200*                                                                       *      02/21/89
       038300*    ADDITIONAL ITEM INFORMATION IS PROCESSED AND THE RESULT            *      02/21/89
       038400*    DISPLAYED ON THE SCREEN WHEN A RESPONSE IS READ FROM THE           *      02/21/89
       038500*    DISPLAY STATION WITH AN ITEM SCREEN RECORD FORMAT.                 *      02/21/89
       038600*                                                                       *      02/21/89
       038700*    IF CMD KEY 1 IS PRESSED, THE PROGRAM IS ENDED.  IF CMD KEY 2       *      02/21/89
       038800*    IS PRESSED, THE ITEM INQUIRY IS ENDED, AND THE MAIN MENU           *      02/21/89
       038900*    (CIMENU) IS WRITTEN TO THE SCREEN.  IF CMD KEY 3 IS PRESSED,       *      02/21/89
       039000*    THE ITEM INQUIRY MENU IS WRITTEN TO THE SCREEN.  BY PRESSING       *      02/21/89
       039100*    ENTER WHEN SCREEN 2 IS DISPLAYED, MORE INFORMATION (PROFIT-        *      02/21/89
       039200*    LOSS) IS WRITTEN TO THE SCREEN.  IF SCREEN 3 IS DISPLAYED,         *      02/21/89
       039300*    PRESSING ENTER WILL CAUSE THE ITEM INQUIRY MENU TO BE              *      02/21/89
       039400*    WRITTEN TO THE SCREEN.                                             *      02/21/89
       039500*                                                                       *      02/21/89
       039600*************************************************************************      02/21/89
       039700* 10                                                                          09/30/87
  307  039800 ITMRTN-ROUTINE.                                                                09/30/87
  308  039900     IF CMD-KEY = "01"                                                          09/30/87
  309  040000        PERFORM DETACH-ROUTINE THRU DETACH-EXIT                                 10/12/87
```

Figure 10-22 (Part 9 of 14). Source Program Example — CSDMUL (User-Defined Formats)

```
 310  040100        GO TO END-JOB.                                                    10/12/87
5728CB1 R01 M02 881028              COBOL SOURCE LISTING          ICFLIB/CSDMUL          03/21/89 08:20:30      Page   15
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
 311  040200     IF CMD-KEY = "02"                                                   09/30/87
 312  040300        WRITE DSPREC FORMAT IS "CIMENU"                                  10/12/87
 313  040400        GO TO ITMRTN-EXIT.                                               10/12/87
 314  040500     IF CMD-KEY = "03"                                                   09/30/87
 315  040600        WRITE DSPREC FORMAT IS "ITMMNU"                                  10/12/87
 316  040700        GO TO ITMRTN-EXIT.                                               10/12/87
 317  040800     IF RCD-FMT = "ITMSC2"                                               10/12/87
 318  040900        PERFORM PROFIT-LOSS THRU PROFIT-LOSS-EXIT                        10/12/87
 319  041000        WRITE DSPREC FORMAT IS "ITMSC3"                                  10/12/87
 320  041100        GO TO ITMRTN-EXIT.                                               10/12/87
 321  041200     WRITE DSPREC FORMAT IS "ITMMNU".                                    10/12/87
      041300 ITMRTN-EXIT.                                                           09/30/87
      041400     EXIT.                                                             09/30/87
      041500*                                                                      09/30/87
      041600*************************************************************          02/21/89
      041700*                                                           *          02/21/89
      041800*    PROFIT AND LOSS FIGURES ARE CALCULATED FOR THE ITEM NUMBER    *    02/21/89
      041900*    REQUESTED.  THESE ARE USED IN SCREEN TWO OF THE ITEM.         *    02/21/89
      042000*                                                           *          02/21/89
      042100*************************************************************          02/21/89
      042200*                                                                      09/30/87
 322  042300 PROFIT-LOSS.                                                          09/30/87
      042400* 11                                                                   09/30/87
 323  042500        SUBTRACT SLSTM FROM CSTTM GIVING PROFM.                        09/30/87
 324  042600        MULTIPLY PROFM BY 100 GIVING PROFM.                            09/30/87
 325  042700        IF SLSTM GREATER THAN 0                                        09/30/87
 326  042800           DIVIDE PROFM BY SLSTM GIVING PROFM.                         09/30/87
 327  042900        MULTIPLY QTYLST BY PR01 GIVING LOSTS.                          09/30/87
 328  043000        MOVE SLSTM  TO SLSM.                                           09/30/87
 329  043100        MOVE SLSTY  TO SLSY.                                           09/30/87
 330  043200        MOVE CSTTM  TO CSTM.                                           09/30/87
 331  043300        MOVE PROFM  TO PROFIT.                                         09/30/87
 332  043400        MOVE CSTTY  TO CSTY.                                           09/30/87
      043500 PROFIT-LOSS-EXIT.                                                     09/30/87
      043600     EXIT.                                                             09/30/87
5728CB1 R01 M02 881028              COBOL SOURCE LISTING          ICFLIB/CSDMUL          03/21/89 08:20:30      Page   16
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
      043700/                                                                      09/30/87
      043800*************************************************************          09/30/87
      043900*                                                           *          09/30/87
      044000*                       CUSTOMER INQUIRY                    *          09/30/87
      044100*                                                           *          09/30/87
      044200*    THE REQUEST FROM THE CUSTOMER INQUIRY MENU IS PROCESSED.       *    09/30/87
      044300*    IF CMD KEY 1 IS PRESSED, THE PROGRAM IS ENDED.  IF CMD KEY 2   *    10/13/87
      044400*    IS PRESSED, THE MAIN MENU (CIMENU) IS WRITTEN TO THE SCREEN.   *    10/14/87
      044500*                                                           *          09/30/87
      044600*    IF A CUSTOMER NUMBER IS ENTERED, THE CUSTOMER INQUIRY          *    09/30/87
      044700*    REQUEST IS SENT TO THE REMOTE SYSTEM.  THEN DTOUT-ROUTINE      *    10/13/87
      044800*    THRU DTOUT-EXIT ARE PERFORMED.                                 *    10/14/87
      044900*                                                           *          09/30/87
      045000*************************************************************          09/30/87
      045100* 12                                                                   09/30/87
 333  045200 DTLIN-ROUTINE.                                                        09/30/87
 334  045300     IF CMD-KEY = "01"                                                 09/30/87
 335  045400        PERFORM DETACH-ROUTINE THRU DETACH-EXIT                        10/12/87
 336  045500        GO TO END-JOB.                                                 10/12/87
 337  045600     IF CMD-KEY = "02"                                                 10/12/87
 338  045700        WRITE DSPREC FORMAT IS "CIMENU"                                10/12/87
 339  045800        GO TO DTLIN-EXIT.                                              10/12/87
      045900 EVDTL.                                                                09/30/87
 340  046000     MOVE "ICF00   " TO PGM-DEV-NME.                                   09/30/87
 341  046100     MOVE CORR DTLMNU-I TO DTLREQ.                                     09/30/87
 342  046200     MOVE ZEROS TO CMNF-INDIC-AREA.                                    09/30/87
 343  046300     WRITE CMNREC FORMAT IS "DTLREQ"                                   09/30/87
```

Figure 10-22 (Part 10 of 14). Source Program Example — CSDMUL (User-Defined Formats)

```
     046400        TERMINAL IS PGM-DEV-NME.                                          09/30/87
 344 046500     PERFORM DTOUT-ROUTINE THRU DTOUT-EXIT.                               09/30/87
     046600 DTLIN-EXIT.                                                               09/30/87
     046700     EXIT.                                                                 09/30/87
     046800*                                                                          10/14/87
     046900*************************************************************************  09/30/87
     047000*                                                                       *  09/30/87
     047100*                      PROCESS CUSTOMER INFORMATION                     *  09/30/87
     047200*                                                                       *  09/30/87
     047300*     THE CUSTOMER DATA RECEIVED FROM THE TARGET PROGRAM IS             *  10/13/87
     047400*     PROCESSED. IF CUSTOMER NUMBER IS ZERO OR LESS, IT IS AN           *  10/13/87
     047500*     INVALID REQUEST AND THE MAIN MENU IS WRITTEN TO THE SCREEN.       *  10/13/87
     047600*                                                                       *  09/30/87
     047700*************************************************************************  09/30/87
     047800* 13                                                                       09/30/87
 345 047900 DTOUT-ROUTINE.                                                            09/30/87
 346 048000     IF CUSTNO OF DTLRSP-I NOT GREATER THAN 0                             09/30/87
 347 048100        WRITE DSPREC FORMAT IS "CIMENU"                                   09/30/87
 348 048200        GO TO DTOUT-EXIT.                                                 09/30/87
 349 048300     PERFORM CUSTOMER-DETAIL THRU CUSTOMER-DETAIL-EXIT.                   10/12/87
     048400 DTOUT-EXIT.                                                               09/30/87
     048500     EXIT.                                                                 09/30/87
     048600*                                                                          10/14/87
     048700*************************************************************************  02/22/89
     048800*                                                                       *  02/22/89
     048900*    THIS ROUTINE HANDLES THE USER'S REQUEST FOLLOWING THE DISPLAY      *  02/22/89
     049000*    OF THE CUSTOMER INFORMATION.  CMD KEY 1 WILL EXIT THE JOB,         *  02/22/89
     049100*    CMD KEY 2 WILL DISPLAY THE MAIN MENU, AND "ENTER" WILL BRING       *  02/22/89
5728CB1 R01 M02 881028              COBOL SOURCE LISTING            ICFLIB/CSDMUL          03/21/89 08:20:30       Page   17
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
     049200*    UP THE CUSTOMER INQUIRY MENU.                                      *  02/22/89
     049300*                                                                       *  02/22/89
     049400*************************************************************************  02/22/89
     049500* 14                                                                       10/14/87
 350 049600 DTLRTN-ROUTINE.                                                           10/12/87
 351 049700     IF CMD-KEY = "01"                                                    10/12/87
 352 049800        PERFORM DETACH-ROUTINE THRU DETACH-EXIT                           10/12/87
 353 049900        GO TO END-JOB.                                                    10/12/87
 354 050000     IF CMD-KEY = "02"                                                    10/12/87
 355 050100        WRITE DSPREC FORMAT IS "CIMENU"                                   10/12/87
 356 050200        GO TO DTLRTN-EXIT.                                                10/12/87
 357 050300     WRITE DSPREC FORMAT IS "DTLMNU".                                     10/12/87
     050400 DTLRTN-EXIT.                                                              10/12/87
     050500     EXIT.                                                                 10/12/87
     050600*                                                                          10/12/87
     050700*************************************************************************  02/21/89
     050800*                                                                       *  02/21/89
     050900*    THE READ OPERATION TO THE PROGRAM DEVICE IS ISSUED.                *  02/21/89
     051000*    A CHECK IS MADE FOR THREE CONDITIONS FOLLOWING THE READ.           *  02/21/89
     051100*    1) THE REMOTE SYSTEM TIMED OUT, 2) NO DATA RECEIVED, AND           *  02/21/89
     051200*    3) DATA RETURNED IN AN UNEXPECTED RECORD FORMAT.                   *  02/21/89
     051300*                                                                       *  02/21/89
     051400*    IF THE REMOTE SYSTEM TIMES OUT (MAJ-MIN = 0310) A MESSAGE          *  02/21/89
     051500*    IS WRITTEN TO THE SCREEN, ASKING TO TRY AGAIN OR END THE           *  02/21/89
     051600*    PROGRAM.                                                           *  02/21/89
     051700*                                                                       *  02/21/89
     051800*    IF NO DATA IS RECEIVED AFTER THE READ OPERATION TO THE             *  02/21/89
     051900*    PROGRAM DEVICE (MAJ-MIN = 03__ ) THE REQUEST IS SENT AGAIN         *  02/21/89
     052000*    TO THE REMOTE SYSTEM AND THE READ OPERATION IS ISSUED TO           *  02/21/89
     052100*    THE ICF PROGRAM DEVICE.                                            *  02/21/89
     052200*                                                                       *  02/21/89
     052300*    IF THE RECORD RETURNS WITH THE WRONG RECORD FORMAT, THE            *  02/21/89
     052400*    PROGRAM WILL GO TO EXIT-FORMAT-ERR ROUTINE.                        *  02/21/89
     052500*                                                                       *  02/22/89
     052600*************************************************************************  02/21/89
     052700* 15                                                                       10/14/87
     052800*                                                                          09/30/87
 358 052900 CUSTOMER-DETAIL.                                                          09/30/87
```

Figure 10-22 (Part 11 of 14). Source Program Example — CSDMUL (User-Defined Formats)

```
  359  053000     MOVE ZEROS TO CMNF-INDIC-AREA.                                    09/30/87
  360  053100     READ CMNFIL.                                                      09/30/87
  361  053200     IF MAJ-MIN = "0310"                                               10/01/87
  362  053300        WRITE DSPREC FORMAT IS "TIMOUT"                                09/30/87
  363  053400        READ DSPFIL INDICATORS ARE DSPF-INDIC-AREA                     09/30/87
  364  053500        IF TIMRSP = "1" GO TO CUSTOMER-DETAIL END-IF                   01/21/88
  366  053600        IF TIMRSP = "2" GO TO END-JOB END-IF.                          01/21/88
  368  053700     IF MAJ = "03"                                                     09/30/87
  369  053800        MOVE ZEROS TO CMNF-INDIC-AREA                                  09/30/87
  370  053900        WRITE CMNREC FORMAT IS "DTLREQ"                                09/30/87
       054000           TERMINAL IS PGM-DEV-NME                                     09/30/87
  371  054100        GO TO CUSTOMER-DETAIL.                                         09/30/87
  372  054200     IF RCD-FMT-NME IS NOT EQUAL "DTLRSP" GO TO EXIT-FORMAT-ERR.       10/02/87
  374  054300     MOVE CUSTNO OF DTLRSP-I TO CUSTN OF DTLSCR-O.                     10/12/87
  375  054400     MOVE DNAME OF DTLRSP-I TO CNAME OF DTLSCR-O.                      03/21/89
  376  054500     MOVE DLSTOR OF DTLRSP-I TO DLSTR OF DTLSCR-O.                     10/12/87
  377  054600     MOVE DSLSTM OF DTLRSP-I TO DSLSM OF DTLSCR-O.                     10/12/87
5728CB1 R01 M02 881028              COBOL SOURCE LISTING          ICFLIB/CSDMUL          03/21/89 08:20:30       Page   18
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
  378  054700     MOVE DSPM01 OF DTLRSP-I TO DSPM1 OF DTLSCR-O.                     10/12/87
  379  054800     MOVE DSPM02 OF DTLRSP-I TO DSPM2 OF DTLSCR-O.                     10/12/87
  380  054900     MOVE DSTTYD OF DTLRSP-I TO DSTYD OF DTLSCR-O.                     10/12/87
  381  055000     MOVE IDEPT OF DTLRSP-I TO DEPT OF DTLSCR-O.                       10/12/87
  382  055100     WRITE DSPREC FORMAT IS "DTLSCR".                                  10/12/87
       055200 CUSTOMER-DETAIL-EXIT.                                                09/30/87
       055300     EXIT.                                                           09/30/87
5728CB1 R01 M02 881028              COBOL SOURCE LISTING          ICFLIB/CSDMUL          03/21/89 08:20:30       Page   19
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
       055400/                                                                   09/30/87
       055500*************************************************************       02/21/89
       055600*                                                           *       02/21/89
       055700*    THE EVOKE-ROUTINE IS CALLED TO EVOKE THE TARGET PROGRAM.*       02/21/89
       055800*    THE SAME TARGET PROGRAM (ICFLIB/CTDMULCL) IS EVOKED AT  *       02/21/89
       055900*    FOUR DIFFERENT REMOTE SYSTEMS.  THE PROGRAM DEVICE      *       02/21/89
       056000*    IDENTIFIES WHICH SESSION SHOULD BE EVOKED.  THE PROGRAM *       02/21/89
       056100*    DEVICE WAS SPECIFIED IN CMID PRIOR TO CALLING THIS ROUTINE.*    02/21/89
       056200*                                                           *       02/21/89
       056300*************************************************************       02/21/89
       056400* 16                                                                10/14/87
       056500*                                                                   09/30/87
  383  056600 EVOKE-ROUTINE.                                                     09/30/87
  384  056700        MOVE "CTDMULCL" TO PGMID OF EVKREQ-O.                       09/30/87
  385  056800        MOVE "ICFLIB" TO LIB  OF EVKREQ-O.                          09/30/87
  386  056900        MOVE "ICF00   " TO PGM-DEV-NME                              09/30/87
  387  057000        WRITE CMNREC FORMAT IS "EVKREQ"                             09/30/87
       057100           TERMINAL IS PGM-DEV-NME.                                 09/30/87
  388  057200        MOVE "ICF01   " TO PGM-DEV-NME                              09/30/87
  389  057300        WRITE CMNREC FORMAT IS "EVKREQ"                             09/30/87
       057400           TERMINAL IS PGM-DEV-NME.                                 09/30/87
  390  057500        MOVE "ICF02   " TO PGM-DEV-NME                              09/30/87
  391  057600        WRITE CMNREC FORMAT IS "EVKREQ"                             09/30/87
       057700           TERMINAL IS PGM-DEV-NME.                                 09/30/87
  392  057800        MOVE "ICF03   " TO PGM-DEV-NME                              09/30/87
  393  057900        WRITE CMNREC FORMAT IS "EVKREQ"                             09/30/87
       058000           TERMINAL IS PGM-DEV-NME.                                 09/30/87
       058100 EVOKE-EXIT.                                                       09/30/87
       058200     EXIT.                                                          09/30/87
       058300*                                                                  09/30/87
       058400*************************************************************       02/21/89
       058500*                                                           *       02/21/89
       058600*    THE TRANSACTION AND SESSION ARE ENDED FOR EACH OF THE  *       02/21/89
       058700*    REMOTE SYSTEMS.                                        *       02/21/89
       058800*                                                           *       02/21/89
       058900*************************************************************       02/21/89
       059000* 17                                                                10/14/87
  394  059100 ERROR-RECOVERY.                                                    09/30/87
  395  059200     PERFORM DETACH-ROUTINE THRU DETACH-EXIT.                       09/30/87
  396  059300     CLOSE CMNFIL DSPFIL                                            09/30/87
```

Figure 10-22 (Part 12 of 14). Source Program Example — CSDMUL (User-Defined Formats)

```
     059400           QPRINT.                                                                    09/30/87
 397 059500      MOVE "0" TO ERR-SW.                                                             09/30/87
     059600 ERROR-RECOVERY-EXIT.                                                                 09/30/87
     059700      EXIT.                                                                           09/30/87
     059800*************************************************************************            02/21/89
     059900*                                                                       *            02/21/89
     060000* EXIT-FORMAT-ERR IS PERFORMED WHEN A READ TO CMNFIL RETURNS WITH  *                  02/21/89
     060100* AN UNEXPECTED RCD-FMT-NME IN THE I-O-FEEDBACK AREA FOR CMNFIL.   *                  02/21/89
     060200* AN ERROR MESSAGE IS PRINTED AND THE PROGRAM ENDS.                *                  02/21/89
     060300*                                                                  *                  02/21/89
     060400*************************************************************************            02/21/89
     060500* 18                                                                                 10/14/87
 398 060600 EXIT-FORMAT-ERR.                                                                    10/01/87
 399 060700      MOVE MAJ-MIN TO RC.                                                            01/14/88
 400 060800      MOVE "RECORD FORMAT IS INCORRECT ON READ          "                            10/01/87
5728CB1 R01 M02 881028              COBOL SOURCE LISTING            ICFLIB/CSDMUL          03/21/89 08:20:30       Page   20
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
     060900         TO ERRMSG.                                                                  01/14/88
 401 061000      WRITE PRINTREC.                                                                10/01/87
 402 061100      CLOSE  CMNFIL DSPFIL QPRINT.                                                   10/01/87
 403 061200      STOP RUN.                                                                      10/01/87
     061300*                                                                                   09/30/87
     061400*************************************************************************            02/21/89
     061500*                                                                       *            02/21/89
     061600*    THIS ROUTINE IS CALLED TO END THE TRANSACTION WITH THE         *                02/21/89
     061700*    REMOTE SYSTEM.                                                 *                02/21/89
     061800*                                                                   *                02/21/89
     061900*************************************************************************            02/21/89
     062000* 19                                                                                 10/14/87
     062100 DETACH-ROUTINE.                                                                      09/30/87
 404 062200      MOVE "ICF00   " TO PGM-DEV-NME                                                 09/30/87
 405 062300      WRITE CMNREC FORMAT IS "DETACH"                                                09/30/87
     062400         TERMINAL IS PGM-DEV-NME.                                                    09/30/87
 406 062500      MOVE "ICF01   " TO PGM-DEV-NME                                                 09/30/87
 407 062600      WRITE CMNREC FORMAT IS "DETACH"                                                09/30/87
     062700         TERMINAL IS PGM-DEV-NME.                                                    09/30/87
 408 062800      MOVE "ICF02   " TO PGM-DEV-NME                                                 09/30/87
 409 062900      WRITE CMNREC FORMAT IS "DETACH"                                                09/30/87
     063000         TERMINAL IS PGM-DEV-NME.                                                    09/30/87
 410 063100      MOVE "ICF03   " TO PGM-DEV-NME                                                 09/30/87
 411 063200      WRITE CMNREC FORMAT IS "DETACH"                                                09/30/87
     063300         TERMINAL IS PGM-DEV-NME.                                                    09/30/87
     063400 DETACH-EXIT.                                                                        09/30/87
     063500      EXIT.                                                                          09/30/87
     063600*                                                                                   09/30/87
     063700*************************************************************************            02/21/89
     063800*                                                                       *            02/21/89
     063900*    THIS ROUTINE IS CALLED TO RELEASE THE PROGRAM DEVICES, END     *                02/21/89
     064000*    THE SESSION AND END THE PROGRAM.                               *                02/21/89
     064100*                                                                   *                02/21/89
     064200*************************************************************************            02/21/89
     064300* 20                                                                                 10/14/87
     064400*                                                                                   09/30/87
 412 064500 END-JOB.                                                                            09/30/87
 413 064600      DROP "ICF00   " FROM CMNFIL.                                                   09/30/87
 414 064700      DROP "ICF01   " FROM CMNFIL.                                                   09/30/87
 415 064800      DROP "ICF02   " FROM CMNFIL.                                                   09/30/87
 416 064900      DROP "ICF03   " FROM CMNFIL.                                                   09/30/87
 417 065000      CLOSE  CMNFIL DSPFIL QPRINT.                                                   09/30/87
 418 065100      STOP RUN.                                                                      09/30/87
     065200*                                                                                   09/30/87
                        * * * * *  E N D  O F  S O U R C E  * * * * *
5728CB1 R01 M02 881028                COBOL MESSAGES                ICFLIB/CSDMUL          03/21/89 08:20:30       Page   21
 STMT
*   25  MSGID: LBL0600  SEVERITY: 10  SEQNBR:  005200
        Message . . . . :   No INPUT fields found for format DETACH.
*   25  MSGID: LBL0600  SEVERITY: 10  SEQNBR:  005200
        Message . . . . :   No INPUT fields found for format EOS.
```

Figure 10-22 (Part 13 of 14). Source Program Example — CSDMUL (User-Defined Formats)

```
*   25  MSGID: LBL0600  SEVERITY: 10  SEQNBR:  005200
        Message . . . . :   No INPUT fields found for format EVKREQ.
*   80  MSGID: LBL0600  SEVERITY: 10  SEQNBR:  005600
        Message . . . . :   No OUTPUT fields found for format CIMENU.
*   80  MSGID: LBL0600  SEVERITY: 10  SEQNBR:  005600
        Message . . . . :   No OUTPUT fields found for format DTLMNU.
*   80  MSGID: LBL0600  SEVERITY: 10  SEQNBR:  005600
        Message . . . . :   No OUTPUT fields found for format ITMMNU.
*   80  MSGID: LBL0600  SEVERITY: 10  SEQNBR:  005600
        Message . . . . :   No OUTPUT fields found for format TIMOUT.
                                      MESSAGE SUMMARY
  TOTAL    INFO(0-4)    WARNING(5-19)    ERROR(20-29)    SEVERE(30-39)    TERMINAL(40-99)
     7          0              7               0               0                0
                    * * * * *  E N D  O F  C O B O L  M E S S A G E S  * * * * *
 652 source records read
 171 copy records read
 2 copy members processed
 0 sequence errors
 10 was the highest severity message issued
  LBL0901 00  Program CSDMUL created in library ICFLIB.
                     * * * * *  E N D  O F  C O M P I L A T I O N  * * * * *
```

Figure 10-22 (Part 14 of 14). Source Program Example — CSDMUL (User-Defined Formats)

```
5728CB1 R01 M02 881028              IBM AS/400  COBOL/400          ICFLIB/CSFMUL          03/21/89 08:22:28       Page     1
Program name . . . . . . . . . . . : CSFMUL in ICFLIB
Source file  . . . . . . . . . . . : QICFPUB in ICFLIB     Member - CSFMUL     03/21/89 08:17:15
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
User profile . . . . . . . . . . . : *USER
Authority  . . . . . . . . . . . . : *CHANGE
Text . . . . . . . . . . . . . . . : CBL Multiple Session Inquiry - Source $$
Compiler . . . . . . . . . . . . . : IBM AS/400  COBOL/400
5728CB1 R01 M02 881028              COBOL SOURCE LISTING          ICFLIB/CSFMUL          03/21/89 08:22:28       Page     2
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
    1  000100 IDENTIFICATION DIVISION.                                                              10/01/87
    2  000200 PROGRAM-ID.              CSFMUL.                                                      10/01/87
       000300*************************************************************************           10/01/87
       000400*  THIS PROGRAM ASSIGNS FOUR SESSIONS AS FOLLOWS:                      *              10/01/87
       000500*    'ICF00' TO INQUIRE ABOUT A CUSTOMER ACCOUNT BEFORE AN             *              10/01/87
       000600*            ORDER IS PROCESSED.                                       *              10/01/87
       000700*    'ICF01' TO INQUIRE ABOUT THE INVENTORY STATUS OF AN ITEM          *              10/01/87
       000800*            BEING ORDERED (ITEM 000001 THRU 399999).                  *              10/01/87
       000900*    'ICF02' TO INQUIRE ABOUT THE INVENTORY STATUS OF AN ITEM          *              10/01/87
       001000*            BEING ORDERED (ITEM 400000 THRU 699999).                  *              10/01/87
       001100*    'ICF03' TO INQUIRE ABOUT THE INVENTORY STATUS OF AN ITEM          *              10/01/87
       001200*            BEING ORDERED (ITEM 700000 THRU 999999).                  *              10/01/87
       001300*  A DISPLAY DEVICE IS USED TO ENTER THE REQUEST ( USING A             *              10/01/87
       001400*  CUSTOMER AND AN ITEM MENU) THAT IS SENT TO THE REMOTE               *              10/01/87
       001500*  SYSTEM.                                                             *              10/15/87
       001600*************************************************************************           10/01/87
    3  001700 ENVIRONMENT DIVISION.                                                              10/01/87
    4  001800 CONFIGURATION SECTION.                                                             10/01/87
    5  001900 SOURCE-COMPUTER.          IBM-AS400.                                               01/15/88
    6  002000 OBJECT-COMPUTER.          IBM-AS400.                                               01/15/88
    7  002100 SPECIAL-NAMES.            I-O-FEEDBACK IS IO-FEEDBACK                              10/01/87
    8  002200                           OPEN-FEEDBACK IS OPEN-FBA.                               10/01/87
    9  002300 INPUT-OUTPUT SECTION.                                                              10/01/87
   10  002400 FILE-CONTROL.                                                                      10/01/87
       002500* 1                                                                                 10/14/87
       002600*************************************************************************           10/01/87
       002700*                                                                     *             10/01/87
       002800*                F I L E   S P E C I F I C A T I O N S                *             10/01/87
       002900*                                                                     *             10/01/87
       003000*    CMNFIL :   ICF FILE USED TO SEND A REQUEST TO ONE                *             02/21/89
       003100*               OF FOUR DIFFERENT TARGET PROGRAMS.  MULTIPLE          *             10/01/87
       003200*               SESSIONS ARE ACTIVE CONCURRENTLY.                     *             10/01/87
       003300*                                                                     *             10/01/87
       003400*    DSPFIL :   DISPLAY FILE USED TO ENTER A REQUEST TO BE            *             10/14/87
       003500*               SENT TO A REMOTE SYSTEM.                              *             10/01/87
       003600*                                                                     *             10/01/87
       003700*************************************************************************           10/01/87
   11  003800     SELECT CMNFIL ASSIGN TO WORKSTATION-CMNFIL-SI                                  10/08/87
   12  003900         ORGANIZATION IS TRANSACTION                                                10/01/87
   13  004000         CONTROL-AREA IS TR-CTL-AREA                                                10/01/87
   14  004100         FILE STATUS IS STATUS-IND MAJ-MIN.                                         10/01/87
   15  004200     SELECT DSPFIL ASSIGN TO WORKSTATION-DSPFIL                                     10/08/87
   16  004300         ORGANIZATION IS TRANSACTION                                                10/01/87
   17  004400         CONTROL-AREA IS DISPLAY-FEEDBACK                                           10/01/87
   18  004500         FILE STATUS IS STATUS-DSP.                                                 10/01/87
   19  004600     SELECT QPRINT ASSIGN TO PRINTER-QSYSPRT.                                       10/01/87
   20  004700 DATA DIVISION.                                                                     10/01/87
   21  004800 FILE SECTION.                                                                      10/01/87
   22  004900 FD  CMNFIL                                                                         10/08/87
   23  005000     LABEL RECORDS ARE STANDARD.                                                    10/01/87
   24  005100 01  CMNREC.                                                                       10/01/87
```

Figure 10-23 (Part 1 of 13). Source Program Example — CSFMUL (System-Supplied Formats)

```
   25  005200     05  ITMRSP.                                                                  10/01/87
   26  005300         07 RECITM                  PIC X.                                       10/01/87
   27  005400         07 ITEMNO                  PIC 9(6).                                    10/01/87
   28  005500         07 DESC                    PIC X(30).                                   10/01/87
5728CB1 R01 M02 881028              COBOL SOURCE LISTING            ICFLIB/CSFMUL          03/21/89 08:22:28        Page     3
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
   29  005600         07 QTYLST                  PIC 9(7).                                    10/01/87
   30  005700         07 QTYOH                   PIC 9(7).                                    10/01/87
   31  005800         07 QTYOO                   PIC 9(7).                                    10/01/87
   32  005900         07 QTYBO                   PIC 9(7).                                    10/01/87
   33  006000         07 UNITQ                   PIC 99.                                      10/01/87
   34  006100         07 PR01                    PIC 9(5)V99.                                 10/01/87
   35  006200         07 PR05                    PIC 9(7).                                    10/01/87
   36  006300         07 UFRT                    PIC 999V99.                                  10/01/87
   37  006400         07 SLSTM                   PIC 9(7)V99.                                 10/01/87
   38  006500         07 SLSTY                   PIC 9(9)V99.                                 10/01/87
   39  006600         07 CSTTM                   PIC 9(7)V99.                                 10/01/87
   40  006700         07 CSTTY                   PIC 9(9)V99.                                 10/01/87
   41  006800         07 PRO                     PIC 999V99.                                  10/01/87
   42  006900         07 LOS                     PIC 9(7)V99.                                 10/01/87
   43  007000         07 FILL1                   PIC X(56).                                   10/01/87
   44  007100     05  ITMREQ REDEFINES ITMRSP.                                                10/01/87
   45  007200         07  LNGTH                  PIC 9(4).                                    02/22/89
   46  007300         07  ITEMNO                 PIC 9(6).                                    10/01/87
   47  007400     05  DTLREQ REDEFINES ITMRSP.                                                10/01/87
   48  007500         07  LNGTH                 PIC 9(4).                                    02/22/89
   49  007600         07  CUSTNO                 PIC 9(6).                                    10/01/87
   50  007700     05  DTLRSP REDEFINES ITMRSP.                                                10/01/87
   51  007800         07 RECCUS                  PIC X.                                       10/01/87
   52  007900         07 CUSTNO                  PIC 9(6).                                    10/01/87
   53  008000         07 DNAME                   PIC X(30).                                   10/01/87
   54  008100         07 DLSTOR                  PIC 9(6).                                    10/01/87
   55  008200         07 DSLSTM                  PIC 9(9).                                    10/01/87
   56  008300         07 DSPM01                  PIC 9(9).                                    10/01/87
   57  008400         07 DSPM02                  PIC 9(9).                                    10/01/87
   58  008500         07 DSPM03                  PIC 9(9).                                    10/01/87
   59  008600         07 DSTTYD                  PIC 9(11).                                   10/01/87
   60  008700         07 IDEPT                   PIC 999.                                     10/01/87
   61  008800         07 FILL2                   PIC X(57).                                   10/01/87
   62  008900     05  EVKREQ REDEFINES ITMRSP.                                                10/01/87
   63  009000         07  PGMID                  PIC X(8).                                    10/01/87
   64  009100         07  FILLER                 PIC X(16).                                   10/01/87
   65  009200         07  LIB                    PIC X(8).                                    10/01/87
   66  009300         07  FILLER                 PIC X(20).                                   10/01/87
   67  009400         07  LNGTH                  PIC 9(4).                                    02/22/89
   68  009500 FD  DSPFIL                                                                       10/08/87
   69  009600     LABEL RECORDS ARE STANDARD.                                                  10/01/87
   70  009700 01  DSPREC.                                                                     10/01/87
   71  009800     COPY DDS-ALL-FORMATS-I-O OF DSPFIL.                                          10/08/87
   72 +000001       05  DSPFIL-RECORD PIC X(79).                                 <-ALL-FMTS
      +000002*  INPUT FORMAT:CIMENU     FROM FILE DSPFIL     OF LIBRARY ICFLIB   <-ALL-FMTS
      +000003*                         MENU FOR INQUIRY                         <-ALL-FMTS
   73 +000004       05  CIMENU-I      REDEFINES DSPFIL-RECORD.                  <-ALL-FMTS
   74 +000005           06 CIMENU-I-INDIC.                                       <-ALL-FMTS
   75 +000006                07 IN99         PIC 1  INDIC 99.                   <-ALL-FMTS
   76 +000007                07 IN98         PIC 1  INDIC 98.                   <-ALL-FMTS
   77 +000008                07 IN97         PIC 1  INDIC 97.                   <-ALL-FMTS
   78 +000009           06 OPTION            PIC X(1).                          <-ALL-FMTS
      +000010* OUTPUT FORMAT:CIMENU     FROM FILE DSPFIL     OF LIBRARY ICFLIB   <-ALL-FMTS
      +000011*                         MENU FOR INQUIRY                         <-ALL-FMTS
      +000012*      05  CIMENU-O      REDEFINES DSPFIL-RECORD.                  <-ALL-FMTS
5728CB1 R01 M02 881028              COBOL SOURCE LISTING            ICFLIB/CSFMUL          03/21/89 08:22:28        Page     4
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
      +000013*  INPUT FORMAT:DTLMNU     FROM FILE DSPFIL     OF LIBRARY ICFLIB   <-ALL-FMTS
      +000014*                         CUSTOMER INQUIRY SCREEN 1                <-ALL-FMTS
   79 +000015       05  DTLMNU-I      REDEFINES DSPFIL-RECORD.                  <-ALL-FMTS
   80 +000016           06 DTLMNU-I-INDIC.                                       <-ALL-FMTS
   81 +000017                07 IN99         PIC 1  INDIC 99.                   <-ALL-FMTS
```

Figure 10-23 (Part 2 of 13). Source Program Example — CSFMUL (System-Supplied Formats)

```
 82 +000018                07 IN98         PIC 1  INDIC 98.                                  <-ALL-FMTS
 83 +000019                07 IN97         PIC 1  INDIC 97.                                  <-ALL-FMTS
 84 +000020             06 CUSTNO          PIC S9(6).                                        <-ALL-FMTS
    +000021* OUTPUT FORMAT:DTLMNU     FROM FILE DSPFIL     OF LIBRARY ICFLIB                   <-ALL-FMTS
    +000022*                          CUSTOMER INQUIRY SCREEN 1                               <-ALL-FMTS
    +000023*      05  DTLMNU-O      REDEFINES DSPFIL-RECORD.                                  <-ALL-FMTS
    +000024*  INPUT FORMAT:DTLSCR     FROM FILE DSPFIL     OF LIBRARY ICFLIB                   <-ALL-FMTS
    +000025*                          CUSTOMER INQUIRY SCR. #2                                <-ALL-FMTS
 85 +000026       05  DTLSCR-I      REDEFINES DSPFIL-RECORD.                                  <-ALL-FMTS
 86 +000027           06 DTLSCR-I-INDIC.                                                      <-ALL-FMTS
 87 +000028                07 IN99         PIC 1  INDIC 99.                                  <-ALL-FMTS
 88 +000029                07 IN98         PIC 1  INDIC 98.                                  <-ALL-FMTS
 89 +000030                07 IN97         PIC 1  INDIC 97.                                  <-ALL-FMTS
    +000031* OUTPUT FORMAT:DTLSCR     FROM FILE DSPFIL     OF LIBRARY ICFLIB                   <-ALL-FMTS
    +000032*                          CUSTOMER INQUIRY SCR. #2                                <-ALL-FMTS
 90 +000033       05  DTLSCR-O      REDEFINES DSPFIL-RECORD.                                  <-ALL-FMTS
 91 +000034           06 CUSTN           PIC X(6).                                           <-ALL-FMTS
 92 +000035           06 DEPT            PIC S9(3).                                          <-ALL-FMTS
 93 +000036           06 DLSTR           PIC S9(6).                                          <-ALL-FMTS
 94 +000037           06 DSLSM           PIC S9(9).                                          <-ALL-FMTS
 95 +000038           06 DSPM1           PIC S9(9).                                          <-ALL-FMTS
 96 +000039           06 DSPM2           PIC S9(9).                                          <-ALL-FMTS
 97 +000040           06 DSPM3           PIC S9(9).                                          <-ALL-FMTS
 98 +000041           06 DSTYD           PIC S9(11).                                         <-ALL-FMTS
 99 +000042           06 CNAME           PIC X(5).                                           <-ALL-FMTS
    +000043*  INPUT FORMAT:ITMMNU     FROM FILE DSPFIL     OF LIBRARY ICFLIB                   <-ALL-FMTS
    +000044*                          ITEM INQUIRY SCREEN ONE                                 <-ALL-FMTS
100 +000045       05  ITMMNU-I      REDEFINES DSPFIL-RECORD.                                  <-ALL-FMTS
101 +000046           06 ITMMNU-I-INDIC.                                                      <-ALL-FMTS
102 +000047                07 IN99         PIC 1  INDIC 99.                                  <-ALL-FMTS
103 +000048                07 IN98         PIC 1  INDIC 98.                                  <-ALL-FMTS
104 +000049                07 IN97         PIC 1  INDIC 97.                                  <-ALL-FMTS
105 +000050           06 ITEMNO          PIC S9(6).                                          <-ALL-FMTS
    +000051* OUTPUT FORMAT:ITMMNU     FROM FILE DSPFIL     OF LIBRARY ICFLIB                   <-ALL-FMTS
    +000052*                          ITEM INQUIRY SCREEN ONE                                 <-ALL-FMTS
    +000053*      05  ITMMNU-O      REDEFINES DSPFIL-RECORD.                                  <-ALL-FMTS
    +000054*  INPUT FORMAT:ITMSC2     FROM FILE DSPFIL     OF LIBRARY ICFLIB                   <-ALL-FMTS
    +000055*                          ITEM INQUIRY SCREEN TWO                                 <-ALL-FMTS
106 +000056       05  ITMSC2-I      REDEFINES DSPFIL-RECORD.                                  <-ALL-FMTS
107 +000057           06 ITMSC2-I-INDIC.                                                      <-ALL-FMTS
108 +000058                07 IN99         PIC 1  INDIC 99.                                  <-ALL-FMTS
109 +000059                07 IN98         PIC 1  INDIC 98.                                  <-ALL-FMTS
110 +000060                07 IN97         PIC 1  INDIC 97.                                  <-ALL-FMTS
    +000061* OUTPUT FORMAT:ITMSC2     FROM FILE DSPFIL     OF LIBRARY ICFLIB                   <-ALL-FMTS
    +000062*                          ITEM INQUIRY SCREEN TWO                                 <-ALL-FMTS
111 +000063       05  ITMSC2-O      REDEFINES DSPFIL-RECORD.                                  <-ALL-FMTS
112 +000064           06 DSC             PIC X(30).                                          <-ALL-FMTS
113 +000065           06 QAVAIL           PIC S9(7).                                          <-ALL-FMTS
114 +000066           06 QTYH            PIC S9(7).                                          <-ALL-FMTS
115 +000067           06 QTYO            PIC S9(7).                                          <-ALL-FMTS
5728CB1 R01 M02 881028               COBOL SOURCE LISTING            ICFLIB/CSFMUL          03/21/89 08:22:28       Page     5
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
116 +000068           06 QTYB            PIC S9(7).                                          <-ALL-FMTS
117 +000069           06 UNT             PIC X(2).                                           <-ALL-FMTS
118 +000070           06 PR1             PIC S9(5)V9(2).                                     <-ALL-FMTS
119 +000071           06 PR5             PIC S9(7).                                          <-ALL-FMTS
120 +000072           06 UFR             PIC S9(3)V9(2).                                     <-ALL-FMTS
    +000073*  INPUT FORMAT:ITMSC3     FROM FILE DSPFIL     OF LIBRARY ICFLIB                   <-ALL-FMTS
    +000074*                          ITEM INQUIRY SCREEN 3                                   <-ALL-FMTS
121 +000075       05  ITMSC3-I      REDEFINES DSPFIL-RECORD.                                  <-ALL-FMTS
122 +000076           06 ITMSC3-I-INDIC.                                                      <-ALL-FMTS
123 +000077                07 IN99         PIC 1  INDIC 99.                                  <-ALL-FMTS
124 +000078                07 IN98         PIC 1  INDIC 98.                                  <-ALL-FMTS
125 +000079                07 IN97         PIC 1  INDIC 97.                                  <-ALL-FMTS
    +000080* OUTPUT FORMAT:ITMSC3     FROM FILE DSPFIL     OF LIBRARY ICFLIB                   <-ALL-FMTS
    +000081*                          ITEM INQUIRY SCREEN 3                                   <-ALL-FMTS
126 +000082       05  ITMSC3-O      REDEFINES DSPFIL-RECORD.                                  <-ALL-FMTS
127 +000083           06 SLSM            PIC S9(7)V9(2).                                     <-ALL-FMTS
```

Figure 10-23 (Part 3 of 13). Source Program Example — CSFMUL (System-Supplied Formats)

```
 128 +000084           06 SLSY             PIC S9(9)V9(2).                          <-ALL-FMTS
 129 +000085           06 CSTM             PIC S9(7)V9(2).                          <-ALL-FMTS
 130 +000086           06 CSTY             PIC S9(9)V9(2).                          <-ALL-FMTS
 131 +000087           06 PROFIT           PIC S9(3)V9(2).                          <-ALL-FMTS
 132 +000088           06 LOSTS            PIC S9(7)V9(2).                          <-ALL-FMTS
     +000089*  INPUT FORMAT:TIMOUT     FROM FILE DSPFIL     OF LIBRARY ICFLIB       <-ALL-FMTS
     +000090*                          TIME OUT SCREEN                              <-ALL-FMTS
 133 +000091       05  TIMOUT-I      REDEFINES DSPFIL-RECORD.                       <-ALL-FMTS
 134 +000092           06 TIMOUT-I-INDIC.                                           <-ALL-FMTS
 135 +000093                07 IN99        PIC 1  INDIC 99.                        <-ALL-FMTS
 136 +000094                07 IN98        PIC 1  INDIC 98.                        <-ALL-FMTS
 137 +000095                07 IN97        PIC 1  INDIC 97.                        <-ALL-FMTS
 138 +000096           06 TIMRSP           PIC X(1).                                <-ALL-FMTS
     +000097* OUTPUT FORMAT:TIMOUT     FROM FILE DSPFIL     OF LIBRARYICFLIB      <-ALL-FMTS
     +000098*                          TIME OUT SCREEN                              <-ALL-FMTS
     +000099*      05  TIMOUT-O      REDEFINES DSPFIL-RECORD.                       <-ALL-FMTS
 139  009900 FD  QPRINT                                                                    10/01/87
 140  010000     LABEL RECORDS ARE OMITTED.                                               10/01/87
 141  010100 01  PRINTREC.                                                               01/14/88
 142  010200     05  RC                        PIC 9999.                                 01/15/88
 143  010300     05  ERRMSG                    PIC X(128).                               01/14/88
 144  010400 WORKING-STORAGE SECTION.                                                     10/01/87
 145  010500 77  STATUS-IND                    PIC X(2).                                 10/01/87
 146  010600 77  STATUS-DSP                    PIC X(2).                                 10/01/87
 147  010700 77  MAJ-MIN-SAV                   PIC X(4).                                 10/01/87
 148  010800 77  EOF-PFILE-SW                  PIC X VALUE "0".                          10/01/87
 149  010900 77  ERR-SW                        PIC X VALUE "0".                          10/01/87
 150  011000 77  INDON                         PIC 1 VALUE B"1".                         10/01/87
 151  011100 77  INDOFF                        PIC 1 VALUE B"0".                         10/01/87
 152  011200 77  OPEN-COUNT                    PIC 9(1) VALUE 0.                         10/01/87
 153  011300 77  LEN                           PIC 9(10)V9(5) COMP.                      10/01/87
 154  011400 77  PROFM                         PIC 9(7)V9(2) COMP-4.                     10/01/87
 155  011500 77  CMD2                          PIC X(31)                                 10/01/87
 156  011600       VALUE "CPYF HEXDUMP *LIST PRTFMT(*HEX)".                              10/01/87
 157  011700 01  TR-CTL-AREA.                                                            10/01/87
 158  011800     05  FILLER                    PIC X(2).                                 10/01/87
 159  011900     05  PGM-DEV-NME               PIC X(10).                                10/01/87
 160  012000     05  RCD-FMT-NME               PIC X(10).                                10/01/87
 161  012100 01  DSPF-INDIC-AREA.                                                        10/01/87
5728CB1 R01 M02 881028              COBOL SOURCE LISTING            ICFLIB/CSFMUL          03/21/89 08:22:28       Page     6
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
 162  012200     05  IN23                      PIC 1 INDIC 23.                           10/01/87
 163  012300         88 IN23-ON                     VALUE B"1".                         10/01/87
 164  012400         88 IN23-OFF                    VALUE B"0".                         10/01/87
 165  012500     05  IN97                      PIC 1 INDIC 97.                           10/01/87
 166  012600         88 IN97-ON                     VALUE B"1".                         10/01/87
 167  012700         88 IN97-OFF                    VALUE B"0".                         10/01/87
 168  012800     05  IN98                      PIC 1 INDIC 98.                           10/01/87
 169  012900         88 IN98-ON                     VALUE B"1".                         10/01/87
 170  013000         88 IN98-OFF                    VALUE B"0".                         10/01/87
 171  013100     05  IN99                      PIC 1 INDIC 99.                           10/01/87
 172  013200         88 IN99-ON                     VALUE B"1".                         10/01/87
 173  013300         88 IN99-OFF                    VALUE B"0".                         10/01/87
 174  013400 01  MAJ-MIN.                                                                10/01/87
 175  013500     05  MAJ                        PIC X(2).                               10/01/87
 176  013600     05  MIN                        PIC X(2).                               10/01/87
 177  013700 01  DISPLAY-FEEDBACK.                                                       10/01/87
 178  013800     05  CMD-KEY                   PIC X(2).                                 10/01/87
 179  013900     05  FILLER                    PIC X(10).                                10/01/87
 180  014000     05  RCD-FMT                   PIC X(10).                                10/01/87
5728CB1 R01 M02 881028              COBOL SOURCE LISTING            ICFLIB/CSFMUL          03/21/89 08:22:28       Page     7
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
      014100/                                                                             10/01/87
 181  014200 PROCEDURE DIVISION.                                                          10/01/87
      014300 DECLARATIVES.                                                                10/01/87
      014400* 2                                                                          10/14/87
```

Figure 10-23 (Part 4 of 13). Source Program Example — CSFMUL (System-Supplied Formats)

```
     014500*************************************************************************                02/22/89
     014600*                                                                       *                02/22/89
     014700* AN ERROR ON THE DISPLAY FILE - DSPFIL - MAKES IT INACTIVE             *                02/22/89
     014800* AN ERROR MESSAGE IS PRINTED, THE FILES ARE CLOSED AND THE             *                02/22/89
     014900* PROGRAM IS ENDED.                                                     *                02/22/89
     015000*                                                                       *                02/22/89
     015100*************************************************************************                02/22/89
     015200*                                                                                        10/08/87
     015300 DSP-ERROR SECTION.                                                                       10/05/87
     015400     USE AFTER STANDARD ERROR PROCEDURE ON DSPFIL.                                       10/08/87
     015500*                                                                                        10/05/87
     015600 DSPFIL-EXCEPTION.                                                                        10/08/87
 182 015700     MOVE "DISPLAY ERROR. JOB TERMINATED" TO ERRMSG.                                     02/21/89
 183 015800     WRITE PRINTREC.                                                                     02/21/89
 184 015900     CLOSE CMNFIL DSPFIL QPRINT.                                                         02/21/89
 185 016000     STOP RUN.                                                                           02/21/89
     016100*                                                                                        10/05/87
     016200 CMN-ERROR SECTION.                                                                       10/14/87
     016300     USE AFTER STANDARD ERROR PROCEDURE ON CMNFIL.                                       10/08/87
     016400 CMNFIL-EXCEPTION.                                                                        10/08/87
     016500*************************************************************************                02/22/89
     016600* CHECK THE MAJOR/MINOR CODES AND TAKE APPROPRIATE ACTION               *                02/22/89
     016700* MAJOR CODE 34 IS AN INPUT EXCEPTION                                   *                02/22/89
     016800*************************************************************************                02/22/89
 186 016900     IF MAJ-MIN = "3431"                                                                 10/01/87
     017000* DATA TRUNCATED IN INPUT AREA. SAVE RETURN CODE .                                       10/01/87
 187 017100           MOVE MAJ-MIN TO MAJ-MIN-SAV                                                   10/01/87
 188 017200           GO TO EXIT-DECLARATIVES.                                                      10/13/87
     017300*                                                                                        10/08/87
     017400* RECOVERABLE SESSION ERROR.  CLOSE ICF FILE.                                            01/15/88
 189 017500     IF MAJ = "83"                                                                       10/01/87
 190 017600        MOVE MAJ-MIN TO RC                                                               01/14/88
 191 017700        MOVE "PROGRAM STARTED AGAIN DUE TO SESSION ERROR"                                10/01/87
     017800        TO ERRMSG                                                                        01/14/88
 192 017900        WRITE PRINTREC                                                                   10/01/87
 193 018000        MOVE "1" TO ERR-SW                                                               10/01/87
 194 018100        GO TO EXIT-DECLARATIVES.                                                         10/01/87
     018200*                                                                                        10/01/87
     018300*************************************************************************                02/21/89
     018400* THIS ROUTINE IS CALLED WHEN THERE IS A PERMANENT SESSION ERROR.  *                     02/21/89
     018500* GET INFORMATION FROM THE MAJOR-MINOR CODE AND PLACE IT INTO      *                     02/21/89
     018600* A DATA BASE FILE.  THEN PRINT THE FILE IN HEX USING COPYFILE.    *                     02/21/89
     018700*************************************************************************                02/21/89
     018800*                                                                                        10/01/87
     018900 GETFBA.                                                                                  10/01/87
 195 019000     MOVE MAJ-MIN TO RC.                                                                 01/14/88
 196 019100     MOVE "PROGRAM TERMINATED DUE TO ERROR IN CMNFIL FILE"                               10/08/87
     019200        TO ERRMSG.                                                                       01/14/88
 197 019300     WRITE PRINTREC.                                                                     10/01/87
 198 019400     CLOSE CMNFIL  DSPFIL                                                                10/08/87
     019500           QPRINT.                                                                       10/01/87
5728CB1 R01 M02 881028              COBOL SOURCE LISTING            ICFLIB/CSFMUL          03/21/89 08:22:28       Page     8
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
 199 019600     STOP RUN.                                                                           10/01/87
     019700*                                                                                        10/01/87
     019800 EXIT-DECLARATIVES.                                                                       10/01/87
     019900     EXIT.                                                                               02/28/89
     020000*                                                                                        10/01/87
 200 020100 END DECLARATIVES.                                                                       10/01/87
5728CB1 R01 M02 881028              COBOL SOURCE LISTING            ICFLIB/CSFMUL          03/21/89 08:22:28       Page     9
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
     020200/                                                                                        10/01/87
     020300 START-PROGRAM  SECTION.                                                                 10/01/87
     020400*                                                                                        10/01/87
     020500 START-PROGRAM-PARAGRAPH.                                                                10/01/87
     020600* 3                                                                                     10/14/87
 201 020700     OPEN I-O   CMNFIL DSPFIL                                                            10/08/87
     020800          OUTPUT QPRINT.                                                                10/01/87
```

Figure 10-23 (Part 5 of 13). Source Program Example — CSFMUL (System-Supplied Formats)

```
      020900*                                                                   10/01/87
      021000*************************************************************      02/21/89
      021100* THE FOLLOWING TEST IS TO ATTEMPT RECOVERY IF AN ERROR OCCURS     *  02/21/89
      021200* WHEN OPENING THE ICF FILE.                                       *  02/21/89
      021300*************************************************************      02/21/89
      021400* 4                                                                10/14/87
  202 021500     IF ERR-SW = "1"                                               10/01/87
  203 021600        THEN IF OPEN-COUNT IS = 9                                  10/01/87
  204 021700                THEN PERFORM DETACH-ROUTINE THRU DETACH-EXIT       10/01/87
  205 021800                     GO TO END-JOB                                 10/01/87
      021900             ELSE                                                  10/01/87
  206 022000                ADD 1 TO OPEN-COUNT                                10/01/87
      022100                PERFORM ERROR-RECOVERY                             10/01/87
  207 022200                GO TO START-PROGRAM-PARAGRAPH                      10/01/87
      022300     ELSE                                                          10/01/87
  209 022400        MOVE 0 TO OPEN-COUNT.                                      10/01/87
      022500*                                                                  10/01/87
      022600*************************************************************      10/14/87
      022700*                                                           *      10/01/87
      022800*   THE DISPLAY DEVICE IS IMPLICITLY ACQUIRED WHEN THE      *      10/15/87
      022900*   FILE IS OPENED.                                         *      10/01/87
      023000*                                                           *      10/01/87
      023100*   ALL OF THE ICF PROGRAM DEVICES ARE EXPLICITLY ACQUIRED. *      02/21/89
      023200*                                                           *      10/01/87
      023300*   EACH OF THE FOUR TARGET PROGRAMS ARE EVOKED TO ESTABLISH*      10/01/87
      023400*   TRANSACTIONS WITH THE REMOTE SYSTEMS.                   *      10/01/87
      023500*                                                           *      10/01/87
      023600*   THE MAIN INQUIRY MENU (CIMENU) IS WRITTEN TO THE USER'S *      10/01/87
      023700*   DISPLAY.                                                *      10/01/87
      023800*                                                           *      10/01/87
      023900*   EVOKE PROGRAM "CTDMUL" ON REMOTE SYSTEM IN LIBRARY ICFLIB.*    10/14/87
      024000*                                                           *      02/22/89
      024100*************************************************************      10/14/87
      024200* 5                                                                10/14/87
  210 024300     ACQUIRE "ICF00   " FOR CMNFIL.                                10/08/87
  211 024400     ACQUIRE "ICF01   " FOR CMNFIL.                                10/08/87
  212 024500     ACQUIRE "ICF02   " FOR CMNFIL.                                10/08/87
  213 024600     ACQUIRE "ICF03   " FOR CMNFIL.                                10/08/87
  214 024700     PERFORM EVOKE-ROUTINE THRU EVOKE-EXIT.                        10/01/87
      024800*                                                                  10/01/87
  215 024900     WRITE DSPREC FORMAT IS "CIMENU"                               10/01/87
      025000        INDICATORS ARE DSPF-INDIC-AREA.                            10/01/87
      025100*************************************************************      10/01/87
      025200*                                                           *      10/01/87
      025300*                    DETERMINE USER'S REQUEST               *      10/01/87
      025400*                                                           *      10/01/87
      025500*   THIS PORTION OF THE PROGRAM ISSUES A READ TO THE DISPLAY *      10/01/87
      025600*   DEVICE TO RECEIVE THE USER'S REQUEST.  THE TYPE         *      10/15/87
5728CB1 R01 M02 881028               COBOL SOURCE LISTING          ICFLIB/CSFMUL          03/21/89 08:22:28       Page   10
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
      025700*   OF REQUEST MADE IS BASED ON THE DISPLAY FORMAT CURRENTLY *      10/01/87
      025800*   ON THE SCREEN.  THE RECORD FORMAT NAME IS EXTRACTED FROM *      10/01/87
      025900*   THE I/O FEEDBACK AREA OF THE DISPLAY FILE AND USED TO   *      10/13/87
      026000*   DETERMINE WHICH ACTION SHOULD BE TAKEN NEXT.            *      10/13/87
      026100*                                                           *      10/01/87
      026200*************************************************************      10/14/87
      026300* 6                                                                10/14/87
      026400 READRQ.                                                            10/01/87
  216 026500     READ DSPFIL INDICATORS ARE DSPF-INDIC-AREA.                   10/08/87
  217 026600     IF RCD-FMT = "CIMENU"                                         10/01/87
  218 026700         PERFORM MENU-ROUTINE THRU MENU-EXIT                       10/01/87
  219 026800         GO TO READRQ.                                             10/01/87
  220 026900     IF RCD-FMT = "ITMMNU"                                         10/01/87
  221 027000         PERFORM ITMIN-ROUTINE THRU ITMIN-EXIT                     10/01/87
  222 027100         GO TO READRQ.                                             10/01/87
  223 027200     IF RCD-FMT = "ITMSC2"                                         10/01/87
  224 027300         PERFORM ITMRTN-ROUTINE THRU ITMRTN-EXIT                   10/01/87
  225 027400         GO TO READRQ.                                             10/01/87
```

Figure 10-23 (Part 6 of 13). Source Program Example — CSFMUL (System-Supplied Formats)

```
 226  027500     IF RCD-FMT = "ITMSC3"                                                    10/01/87
 227  027600        PERFORM ITMRTN-ROUTINE THRU ITMRTN-EXIT                               10/01/87
 228  027700        GO TO READRQ.                                                         10/01/87
 229  027800     IF RCD-FMT = "DTLMNU"                                                    10/01/87
 230  027900        PERFORM DTLIN-ROUTINE THRU DTLIN-EXIT                                 10/01/87
 231  028000        GO TO READRQ.                                                         10/08/87
 232  028100     IF RCD-FMT = "DTLSCR"                                                    10/08/87
 233  028200        PERFORM DTLRTN-ROUTINE THRU DTLRTN-EXIT                               10/08/87
 234  028300        GO TO READRQ.                                                         10/01/87
 235  028400     WRITE DSPREC FORMAT IS "CIMENU".                                         10/01/87
 236  028500     GO TO READRQ.
5728CB1 R01 M02 881028              COBOL SOURCE LISTING          ICFLIB/CSFMUL          03/21/89 08:22:28       Page   11
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
      028600/                                                                           10/01/87
      028700*************************************************************************  10/13/87
      028800*                                                                       *  10/13/87
      028900*                              MAIN MENU                                *  10/13/87
      029000*                                                                       *  10/13/87
      029100*    THE MAIN MENU IS READ TO DETERMINE THE REQUEST ENTERED             *  10/13/87
      029200*    BY THE USER. IF CMD 1 IS PRESSED, THE PROGRAM IS ENDED.            *  10/13/87
      029300*    IF OPTION = 1, AN ITEM INQUIRY MENU IS WRITTEN TO                  *  10/13/87
      029400*    TO SCREEN.  IF OPTION = 2, A CUSTOMER INQUIRY MENU IS              *  10/13/87
      029500*    WRITTEN TO THE SCREEN.                                             *  10/13/87
      029600*                                                                       *  10/13/87
      029700*************************************************************************  10/14/87
      029800* 7                                                                        10/01/87
      029900 MENU-ROUTINE.                                                             10/01/87
 237  030000     IF CMD-KEY = "01"                                                    10/01/87
 238  030100        PERFORM DETACH-ROUTINE THRU DETACH-EXIT                           10/01/87
 239  030200        GO TO END-JOB.                                                    10/01/87
 240  030300     IF OPTION = "1"                                                      10/01/87
 241  030400        WRITE DSPREC FORMAT IS "ITMMNU"                                   10/01/87
      030500     ELSE                                                                 10/01/87
 242  030600        WRITE DSPREC FORMAT IS "DTLMNU".                                  10/01/87
      030700 MENU-EXIT.                                                               10/01/87
      030800     EXIT.                                                                10/01/87
5728CB1 R01 M02 881028              COBOL SOURCE LISTING          ICFLIB/CSFMUL          03/21/89 08:22:28       Page   12
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
      030900/                                                                           10/01/87
      031000*************************************************************************  10/13/87
      031100*                                                                       *  10/13/87
      031200*                          ITEM INQUIRY                                 *  10/13/87
      031300*                                                                       *  10/13/87
      031400*    THE ITEM NUMBER REQUESTED BY THE USER ON THE ITEM INQUIRY          *  10/13/87
      031500*    SCREEN IS CHECKED. THIS IS DETERMINED BY THE                       *  10/13/87
      031600*    DISPLAY RECORD FORMAT BEING PROCESSED - IN THIS CASE ITMMNU.       *  10/13/87
      031700*                                                                       *  02/21/89
      031800*    IF CMD KEY 1 IS PRESSED, THE PROGRAM IS ENDED.  IF CMD KEY 2       *  10/13/87
      031900*    IS PRESSED, THE ITEM INQUIRY REQUEST IS CANCELED, AND THE          *  10/13/87
      032000*    MAIN MENU (CIMENU) IS WRITTEN TO THE SCREEN.                       *  10/13/87
      032100*                                                                       *  10/13/87
      032200*    IF AN ITEM NUMBER IS ENTERED, A ITEM INQUIRY REQUEST IS            *  10/13/87
      032300*    SENT TO THE APPROPRIATE REMOTE SYSTEM.  THE REMOTE SYSTEM          *  10/13/87
      032400*    IS SELECTED BASED ON THE ITEM NUMBER REQUESTED.                    *  10/13/87
      032500*                                                                       *  10/13/87
      032600*    A CHECK IS MADE FOR THREE CONDITIONS FOLLOWING THE READ.           *  10/14/87
      032700*    1) THE REMOTE SYSTEM TIMED OUT, 2) NO DATA RECEIVED, AND           *  10/14/87
      032800*    3) DATA RETURNED IN AN UNEXPECTED RECORD FORMAT.                   *  10/14/87
      032900*                                                                       *  10/14/87
      033000*    IF THE REMOTE SYSTEM TIMES OUT (MAJ-MIN = 0310) A MESSAGE          *  10/14/87
      033100*    IS WRITTEN TO THE SCREEN, ASKING TO TRY AGAIN OR END THE           *  10/14/87
      033200*    PROGRAM.                                                           *  10/14/87
      033300*                                                                       *  10/14/87
      033400*    IF NO DATA IS RECEIVED AFTER THE READ OPERATION TO THE             *  10/15/87
      033500*    PROGRAM DEVICE (MAJ-MIN = 03__ ) THE REQUEST IS SENT AGAIN         *  10/14/87
      033600*    TO THE REMOTE SYSTEM AND THE READ OPERATION IS ISSUED TO           *  10/14/87
      033700*    THE PROGRAM DEVICE.                                                *  10/15/87
      033800*                                                                       *  10/14/87
```

Figure 10-23 (Part 7 of 13). Source Program Example — CSFMUL (System-Supplied Formats)

```
      033900*    IF THE RECORD RETURNS WITH THE WRONG RECORD FORMAT, THE       *                10/14/87
      034000*    PROGRAM WILL GO TO EXIT-FORMAT-ERR ROUTINE.                   *                10/14/87
      034100*                                                                  *                10/14/87
      034200*****************************************************************                10/13/87
      034300* 8                                                                               10/14/87
  243 034400 ITMIN-ROUTINE.                                                                   10/01/87
  244 034500     IF CMD-KEY = "01"                                                            10/08/87
  245 034600        PERFORM DETACH-ROUTINE THRU DETACH-EXIT                                   10/08/87
  246 034700        GO TO END-JOB.                                                            10/08/87
  247 034800     IF CMD-KEY = "02"                                                            10/08/87
  248 034900        WRITE DSPREC FORMAT IS "CIMENU"                                           10/01/87
  249 035000        GO TO ITMIN-EXIT.                                                         10/01/87
  250 035100     MOVE CORR ITMMNU-I TO ITMREQ.                                                10/01/87
  251 035200     IF ITEMNO OF ITMMNU-I LESS THAN 399999 GO TO XICF01.                         10/01/87
  253 035300     IF ITEMNO OF ITMMNU-I LESS THAN 699999 GO TO XICF02.                         10/01/87
  255 035400     IF ITEMNO OF ITMMNU-I LESS THAN 899999 GO TO XICF03.                         10/01/87
      035500 XICF01.                                                                          10/01/87
  257 035600     MOVE "ICF01   " TO PGM-DEV-NME.                                              10/01/87
  258 035700     GO TO XITMIN.                                                                10/01/87
      035800 XICF02.                                                                          10/01/87
  259 035900     MOVE "ICF02   " TO PGM-DEV-NME.                                              10/01/87
  260 036000     GO TO XITMIN.                                                                10/01/87
      036100 XICF03.                                                                          10/01/87
  261 036200     MOVE "ICF03   " TO PGM-DEV-NME.                                              10/01/87
      036300 XITMIN.                                                                          10/01/87
5728CB1 R01 M02 881028              COBOL SOURCE LISTING          ICFLIB/CSFMUL  03/21/89 08:22:28  Page  13
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME  CHG/DATE
  262 036400     MOVE 150 TO LNGTH OF ITMREQ.                                                 02/22/89
  263 036500     WRITE CMNREC FORMAT IS "$$SEND"                                              10/01/87
      036600        TERMINAL IS PGM-DEV-NME.                                                  10/01/87
      036700 RETRY-ITEM.                                                                      10/01/87
  264 036800     READ CMNFIL.                                                                 10/01/87
  265 036900     IF MAJ-MIN = "0310"                                                          10/08/87
  266 037000        WRITE DSPREC FORMAT IS "TIMOUT"                                           10/01/87
  267 037100        READ DSPFIL INDICATORS ARE DSPF-INDIC-AREA                                10/01/87
  268 037200        IF TIMRSP = "1" GO TO RETRY-ITEM END-IF                                   10/08/87
  270 037300        IF TIMRSP = "2" GO TO END-JOB END-IF.                                     01/21/88
  272 037400     IF MAJ = "03"                                                                01/21/88
  273 037500        GO TO XITMIN.                                                             10/01/87
  274 037600     IF RCD-FMT-NME IS NOT EQUAL "ITMRSP" GO TO EXIT-FORMAT-ERR.                  10/01/87
  276 037700     PERFORM ITMOUT-ROUTINE THRU ITMOUT-EXIT.                                     10/02/87
      037800 ITMIN-EXIT.                                                                      10/01/87
      037900     EXIT.                                                                        10/01/87
5728CB1 R01 M02 881028              COBOL SOURCE LISTING          ICFLIB/CSFMUL  03/21/89 08:22:28  Page  14
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME  CHG/DATE
      038000/                                                                                 10/14/87
      038100*****************************************************************                10/01/87
      038200*                                                                 *               10/01/87
      038300*                     PROCESS ITEM INFORMATION                    *               10/01/87
      038400*                                                                 *               10/01/87
      038500*    THIS SECTION PROCESSES THE ITEM RECORD RECEIVED FROM THE     *               10/01/87
      038600*    TARGET PROGRAM AND THE INFORMATION ABOUT THE ITEM IS         *               02/21/89
      038700*    DISPLAYED.  IF ITEMNO IS 0 OR LESS, IT IS AN INVALID REQUEST *               10/15/87
      038800*    AND A FRESH ITEM MENU I WRITTEN TO THE SCREEN.  IF THE       *               10/15/87
      038900*    REQUEST IS VALID, VALUES ARE CALCULATED BASED ON THE         *               10/15/87
      039000*    INFORMATION RECEIVED.                                        *               10/15/87
      039100*                                                                 *               10/01/87
      039200*****************************************************************                10/01/87
      039300* 9                                                                               10/14/87
  277 039400 ITMOUT-ROUTINE.                                                                  10/01/87
  278 039500     IF ITEMNO OF ITMRSP NOT GREATER THAN 0                                       10/01/87
  279 039600        WRITE DSPREC FORMAT IS "ITMMNU"                                           10/01/87
  280 039700        GO TO ITMOUT-EXIT.                                                        10/01/87
  281 039800     MOVE QTYLST TO QAVAIL OF ITMSC2-O.                                           10/01/87
  282 039900     MOVE QTYOO TO QTYO OF ITMSC2-O.                                              10/01/87
  283 040000     MOVE QTYOH TO QTYH OF ITMSC2-O.                                              10/01/87
  284 040100     MOVE QTYBO TO QTYB OF ITMSC2-O.                                              10/01/87
  285 040200     MOVE UNITQ TO UNT OF ITMSC2-O.                                               10/01/87
```

Figure 10-23 (Part 8 of 13). Source Program Example — CSFMUL (System-Supplied Formats)

```
286  040300     MOVE PR01 TO PR1 OF ITMSC2-O.                                  10/01/87
287  040400     MOVE PR05 TO PR5 OF ITMSC2-O.                                  10/01/87
288  040500     MOVE UFRT TO UFR OF ITMSC2-O.                                  10/01/87
289  040600     MOVE DESC TO DSC OF ITMSC2-O.                                  10/07/87
290  040700     WRITE DSPREC FORMAT IS "ITMSC2".                               10/07/87
     040800 ITMOUT-EXIT.                                                       10/01/87
     040900     EXIT.                                                         10/01/87
     041000*                                                                  10/14/87
     041100*************************************************************      02/21/89
     041200*                                                           *      02/21/89
     041300*                   ADDITIONAL ITEM INFORMATION             *      02/21/89
     041400*                                                           *      02/21/89
     041500*    ADDITIONAL ITEM INFORMATION IS PROCESSED AND THE RESULT *      02/21/89
     041600*    DISPLAYED ON THE SCREEN WHEN A RESPONSE IS READ FROM THE *      02/21/89
     041700*    DISPLAY STATION WITH AN ITEM SCREEN RECORD FORMAT.      *      02/21/89
     041800*                                                           *      02/21/89
     041900*    IF CMD KEY 1 IS PRESSED, THE PROGRAM IS ENDED.  IF CMD KEY 2 *  02/21/89
     042000*    IS PRESSED, THE ITEM INQUIRY IS ENDED, AND THE MAIN MENU *     02/21/89
     042100*    (CIMENU) IS WRITTEN TO THE SCREEN.  IF CMD KEY 3 IS PRESSED, *  02/21/89
     042200*    THE ITEM INQUIRY MENU IS WRITTEN TO THE SCREEN.  BY PRESSING *  02/21/89
     042300*    ENTER WHEN SCREEN 2 IS DISPLAYED, MORE INFORMATION (PROFIT- *   02/21/89
     042400*    LOSS) WILL BE DISPLAYED TO THE SCREEN.  WHEN SCREEN 3 IS *     02/21/89
     042500*    DISPLAYED, AN ENTER WILL WILL CAUSE THE ITEM INQUIRY MENU *    02/21/89
     042600*    TO BE WRITTEN TO THE SCREEN.                            *      02/21/89
     042700*                                                           *      02/21/89
     042800*************************************************************      02/21/89
     042900* 10                                                               10/14/87
 291 043000 ITMRTN-ROUTINE.                                                   10/01/87
 292 043100     IF CMD-KEY = "01"                                            10/08/87
 293 043200        PERFORM DETACH-ROUTINE THRU DETACH-EXIT                   10/08/87
 294 043300        GO TO END-JOB.                                            10/08/87
 295 043400     IF CMD-KEY = "02"                                            10/08/87
5728CB1 R01 M02 881028              COBOL SOURCE LISTING          ICFLIB/CSFMUL          03/21/89 08:22:28       Page    15
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
 296 043500        WRITE DSPREC FORMAT IS "CIMENU"                           10/01/87
 297 043600        GO TO ITMRTN-EXIT.                                        10/01/87
 298 043700     IF CMD-KEY = "03"                                            10/08/87
 299 043800        WRITE DSPREC FORMAT IS "ITMMNU"                           10/08/87
 300 043900        GO TO ITMRTN-EXIT.                                        10/08/87
 301 044000     IF RCD-FMT = "ITMSC2"                                        10/08/87
 302 044100        PERFORM PROFIT-LOSS THRU PROFIT-LOSS-EXIT                 10/01/87
 303 044200        WRITE DSPREC FORMAT IS "ITMSC3"                           10/01/87
 304 044300        GO TO ITMRTN-EXIT.                                        10/01/87
 305 044400     WRITE DSPREC FORMAT IS "ITMMNU".                             10/01/87
     044500 ITMRTN-EXIT.                                                      10/01/87
     044600     EXIT.                                                        10/01/87
     044700*                                                                 10/01/87
     044800*************************************************************      02/21/89
     044900*                                                           *      02/21/89
     045000*    THIS SECTION OF THE PROGRAM IS CALLED TO PROCESS       *      02/21/89
     045100*    THE INFORMATION FOR THE ITEM NUMBER REQUESTED BEFORE   *      02/21/89
     045200*    IT IS SENT BACK TO THE REQUESTING REMOTE SYSTEM.       *      02/21/89
     045300*                                                           *      02/21/89
     045400*************************************************************      02/21/89
     045500* 11                                                               10/14/87
     045600*                                                                  10/01/87
 306 045700 PROFIT-LOSS.                                                      10/01/87
 307 045800        SUBTRACT SLSTM FROM CSTTM GIVING PROFM.                   10/01/87
 308 045900        MULTIPLY PROFM BY 100 GIVING PROFM.                       10/01/87
 309 046000        IF SLSTM GREATER THAN 0                                   10/01/87
 310 046100           DIVIDE PROFM BY SLSTM GIVING PROFM.                    10/01/87
 311 046200        MULTIPLY QTYLST BY PR01 GIVING LOSTS.                     10/01/87
 312 046300        MOVE SLSTM  TO SLSM.                                      10/01/87
 313 046400        MOVE SLSTY  TO SLSY.                                      10/01/87
 314 046500        MOVE CSTTM  TO CSTM.                                      10/01/87
 315 046600        MOVE PROFM  TO PROFIT.                                    10/01/87
 316 046700        MOVE CSTTY  TO CSTY.                                      10/01/87
     046800 PROFIT-LOSS-EXIT.                                                10/01/87
```

Figure 10-23 (Part 9 of 13). Source Program Example — CSFMUL (System-Supplied Formats)

```
      046900     EXIT.                                                                    10/01/87
5728CB1 R01 M02 881028              COBOL SOURCE LISTING          ICFLIB/CSFMUL          03/21/89 08:22:28      Page   16
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
      047000/                                                                           10/01/87
      047100*************************************************************************  10/13/87
      047200*                                                                       *   10/13/87
      047300*                        CUSTOMER INQUIRY                               *   10/13/87
      047400*                                                                       *   10/13/87
      047500*    THE REQUEST FROM THE CUSTOMER INQUIRY MENU IS PROCESSED.           *   10/13/87
      047600*    IF CMD KEY 1 IS PRESSED, THE PROGRAM IS ENDED.  IF CMD KEY 2       *   10/13/87
      047700*    ID PRESSED, THE MAIN MENU (CIMENU) IS WRITTEN TO THE SCREEN.       *   10/13/87
      047800*                                                                       *   10/13/87
      047900*    IF A CUSTOMER NUMBER IS ENTERED, THE CUSTOMER INQUIRY              *   10/13/87
      048000*    REQUEST IS SENT TO THE REMOTE SYSTEM.  THEN DTOUT-ROUTINE          *   10/13/87
      048100*    THRU DTOUT-EXIT ARE PERFORMED.                                     *   10/14/87
      048200*                                                                       *   10/13/87
      048300*************************************************************************  10/13/87
      048400* 12                                                                        10/14/87
  317 048500 DTLIN-ROUTINE.                                                             10/01/87
  318 048600     IF CMD-KEY = "01"                                                      10/08/87
  319 048700        PERFORM DETACH-ROUTINE THRU DETACH-EXIT                             10/08/87
  320 048800        GO TO END-JOB.                                                      10/08/87
  321 048900     IF CMD-KEY = "02"                                                      10/08/87
  322 049000        WRITE DSPREC FORMAT IS "CIMENU"                                     10/08/87
  323 049100        GO TO DTLIN-EXIT.                                                   10/01/87
      049200 EVDTL.                                                                     10/01/87
  324 049300     MOVE 150 TO LNGTH OF DTLREQ.                                           02/22/89
  325 049400     MOVE "ICF00   " TO PGM-DEV-NME.                                        10/01/87
  326 049500     MOVE CORR DTLMNU-I TO DTLREQ.                                          10/01/87
  327 049600     WRITE CMNREC FORMAT IS "$$SEND"                                        10/01/87
      049700        TERMINAL IS PGM-DEV-NME.                                            10/01/87
  328 049800     PERFORM DTOUT-ROUTINE THRU DTOUT-EXIT.                                 10/01/87
      049900 DTLIN-EXIT.                                                                10/01/87
      050000     EXIT.                                                                  10/01/87
      050100*                                                                          10/14/87
      050200*************************************************************************  10/13/87
      050300*                                                                       *   10/13/87
      050400*                    PROCESS CUSTOMER INFORMATION                       *   10/13/87
      050500*                                                                       *   10/13/87
      050600*    THE CUSTOMER DATA RECEIVED FROM THE TARGET PROGRAM IS              *   10/13/87
      050700*    PROCESSED. IF CUSTOMER NUMBER IS ZERO OR LESS, IT IS AN            *   10/13/87
      050800*    INVALID REQUEST AND THE MAIN MENU IS WRITTEN TO THE SCREEN.        *   10/13/87
      050900*                                                                       *   10/13/87
      051000*************************************************************************  10/13/87
      051100* 13                                                                        10/14/87
  329 051200 DTOUT-ROUTINE.                                                             10/01/87
  330 051300     IF CUSTNO OF DTLRSP NOT GREATER THAN 0                                 10/01/87
  331 051400        WRITE DSPREC FORMAT IS "CIMENU"                                     10/01/87
  332 051500        GO TO DTOUT-EXIT.                                                   10/01/87
  333 051600     PERFORM CUSTOMER-DETAIL THRU CUSTOMER-DETAIL-EXIT.                     10/07/87
      051700 DTOUT-EXIT.                                                                10/01/87
      051800     EXIT.                                                                  10/01/87
      051900*                                                                          10/01/87
      052000*************************************************************************  02/21/89
      052100*                                                                       *   02/21/89
      052200*    THIS SECTION OF THE PROGRAM HANDLES THE REQUEST FOLLOWING          *   02/21/89
      052300*    THE DISPLAY OF THE CUSTOMER INFORMATION.  CMD KEY 1 WILL           *   02/21/89
      052400*    EXIT THE JOB, CMD KEY 2 WILL DISPLAY THE MAIN MENU, AND            *   02/21/89
5728CB1 R01 M02 881028              COBOL SOURCE LISTING          ICFLIB/CSFMUL          03/21/89 08:22:28      Page   17
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
      052500*    AN "ENTER" WILL BRING UP THE CUSTOMER INQUIRY MENU.                *   02/21/89
      052600*                                                                       *   02/21/89
      052700*************************************************************************  02/21/89
      052800* 14                                                                        10/14/87
  334 052900 DTLRTN-ROUTINE.                                                            10/08/87
  335 053000     IF CMD-KEY = "01"                                                      10/08/87
  336 053100        PERFORM DETACH-ROUTINE THRU DETACH-EXIT                             10/08/87
  337 053200        GO TO END-JOB.                                                      10/08/87
```

Figure 10-23 (Part 10 of 13). Source Program Example — CSFMUL (System-Supplied Formats)

```
 338  053300     IF CMD-KEY = "02"                                                    10/08/87
 339  053400        WRITE DSPREC FORMAT IS "CIMENU"                                   10/08/87
 340  053500        GO TO DTLRTN-EXIT.                                                10/08/87
 341  053600     WRITE DSPREC FORMAT IS "DTLMNU".                                     10/08/87
      053700 DTLRTN-EXIT.                                                            10/08/87
      053800     EXIT.                                                               10/08/87
5728CB1 R01 M02 881028              COBOL SOURCE LISTING          ICFLIB/CSFMUL      03/21/89 08:22:28  Page   18
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
      053900/                                                                       10/14/87
      054000*************************************************************************02/21/89
      054100*                                                                      *02/21/89
      054200*    THE READ OPERATION TO THE PROGRAM DEVICE IS ISSUED.               *02/21/89
      054300*    A CHECK IS MADE FOR THREE CONDITIONS FOLLOWING THE READ.          *02/21/89
      054400*    1) THE REMOTE SYSTEM TIMED OUT, 2) NO DATA RECEIVED, AND          *02/21/89
      054500*    3) DATA RETURNED IN AN UNEXPECTED RECORD FORMAT.                  *02/21/89
      054600*                                                                      *02/21/89
      054700*    IF THE REMOTE SYSTEM TIMES OUT (MAJ-MIN = 0310) A MESSAGE         *02/21/89
      054800*    IS WRITTEN TO THE SCREEN, ASKING TO TRY AGAIN OR END THE          *02/21/89
      054900*    PROGRAM.                                                          *02/21/89
      055000*                                                                      *02/21/89
      055100*    IF NO DATA IS RECEIVED AFTER THE READ OPERATION TO THE            *02/21/89
      055200*    PROGRAM DEVICE (MAJ-MIN = 03__ ) THE REQUEST IS SENT AGAIN        *02/21/89
      055300*    TO THE REMOTE SYSTEM AND THE READ OPERATION IS ISSUED TO          *02/21/89
      055400*    THE ICF PROGRAM DEVICE.                                           *02/21/89
      055500*                                                                      *02/21/89
      055600*    IF THE RECORD RETURNS WITH THE WRONG RECORD FORMAT, THE           *02/21/89
      055700*    PROGRAM WILL GO TO EXIT-FORMAT-ERR ROUTINE.                       *02/21/89
      055800*************************************************************************02/21/89
      055900* 15                                                                   10/14/87
      056000*                                                                      10/01/87
 342  056100 CUSTOMER-DETAIL.                                                      10/01/87
 343  056200     READ CMNFIL.                                                      10/08/87
 344  056300     IF MAJ-MIN = "0310"                                               10/01/87
 345  056400        WRITE DSPREC FORMAT IS "TIMOUT"                                10/01/87
 346  056500        READ DSPFIL INDICATORS ARE DSPF-INDIC-AREA                     10/08/87
 347  056600        IF TIMRSP = "1" GO TO CUSTOMER-DETAIL END-IF                   01/21/88
 349  056700        IF TIMRSP = "2" GO TO END-JOB END-IF.                          01/21/88
 351  056800     IF MAJ = "03"                                                     10/01/87
 352  056900        WRITE CMNREC FORMAT IS "$$SEND"                                10/01/87
      057000           TERMINAL IS PGM-DEV-NME                                     10/01/87
 353  057100        GO TO CUSTOMER-DETAIL.                                         10/01/87
 354  057200     IF RCD-FMT-NME IS NOT EQUAL "DTLRSP" GO TO EXIT-FORMAT-ERR.       10/02/87
 356  057300     MOVE CUSTNO OF DTLRSP TO CUSTN OF DTLSCR-O.                       10/07/87
 357  057400     MOVE DNAME OF DTLRSP TO CNAME OF DTLSCR-O.                        03/21/89
 358  057500     MOVE DLSTOR OF DTLRSP TO DLSTR OF DTLSCR-O.                       10/07/87
 359  057600     MOVE DSLSTM OF DTLRSP TO DSLSM OF DTLSCR-O.                       10/07/87
 360  057700     MOVE DSPM01 OF DTLRSP TO DSPM1 OF DTLSCR-O.                       10/07/87
 361  057800     MOVE DSPM02 OF DTLRSP TO DSPM2 OF DTLSCR-O.                       10/07/87
 362  057900     MOVE DSPM03 OF DTLRSP TO DSPM3 OF DTLSCR-O.                       10/07/87
 363  058000     MOVE DSTTYD OF DTLRSP TO DSTYD OF DTLSCR-O.                       10/07/87
 364  058100     MOVE IDEPT OF DTLRSP TO DEPT OF DTLSCR-O.                         10/07/87
 365  058200     WRITE DSPREC FORMAT IS "DTLSCR".                                  10/07/87
      058300 CUSTOMER-DETAIL-EXIT.                                                  10/01/87
      058400     EXIT.                                                             10/01/87
5728CB1 R01 M02 881028              COBOL SOURCE LISTING          ICFLIB/CSFMUL      03/21/89 08:22:28  Page   19
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
      058500/                                                                       10/01/87
      058600*************************************************************************02/21/89
      058700*                                                                      *02/21/89
      058800*    THIS SUBROUTINE IS CALLED TO EVOKE THE TARGET PROGRAM.            *02/21/89
      058900*    THE SAME TARGET PROGRAM (ICFLIB/CTFMULCL) IS EVOKED AT            *02/21/89
      059000*    FOUR DIFFERENT REMOTE SYSTEMS.  THE PROGRAM DEVICE                *02/21/89
      059100*    IDENTIFIES WHICH SESSION SHOULD BE EVOKED.  THE PROGRAM           *02/21/89
      059200*    DEVICE WAS SPECIFIED IN CMID PRIOR TO CALLING THIS ROUTINE.       *02/21/89
      059300*                                                                      *02/21/89
      059400*************************************************************************02/21/89
      059500* 16                                                                   10/14/87
      059600*                                                                      10/01/87
```

Figure 10-23 (Part 11 of 13). Source Program Example — CSFMUL (System-Supplied Formats)

```
 366  059700 EVOKE-ROUTINE.                                                  10/01/87
 367  059800        MOVE 0 TO LNGTH OF EVKREQ.                               02/22/89
 368  059900        MOVE "CTFMULCL" TO PGMID OF EVKREQ.                      10/01/87
 369  060000        MOVE "ICFLIB" TO LIB  OF EVKREQ.                         10/08/87
 370  060100        MOVE "ICF00   " TO PGM-DEV-NME                           10/01/87
 371  060200        WRITE CMNREC FORMAT IS "$$EVOKNI"                        10/01/87
      060300           TERMINAL IS PGM-DEV-NME.                              10/01/87
 372  060400        MOVE "ICF01   " TO PGM-DEV-NME                           10/01/87
 373  060500        WRITE CMNREC FORMAT IS "$$EVOKNI"                        10/01/87
      060600           TERMINAL IS PGM-DEV-NME.                              10/01/87
 374  060700        MOVE "ICF02   " TO PGM-DEV-NME                           10/01/87
 375  060800        WRITE CMNREC FORMAT IS "$$EVOKNI"                        10/01/87
      060900           TERMINAL IS PGM-DEV-NME.                              10/01/87
 376  061000        MOVE "ICF03   " TO PGM-DEV-NME                           10/01/87
 377  061100        WRITE CMNREC FORMAT IS "$$EVOKNI"                        10/01/87
      061200           TERMINAL IS PGM-DEV-NME.                              10/01/87
      061300 EVOKE-EXIT.                                                     10/01/87
      061400     EXIT.                                                        10/01/87
      061500*                                                                10/01/87
      061600*************************************************************** 10/01/87
      061700*                                                             * 02/21/89
      061800*    THE TRANSACTION AND SESSION ARE ENDED FOR EACH OF THE    * 02/21/89
      061900*    REMOTE SYSTEMS.                                          * 02/21/89
      062000*                                                             * 02/21/89
      062100*************************************************************** 02/21/89
      062200* 17                                                           10/14/87
 378  062300 ERROR-RECOVERY.                                                 10/01/87
 379  062400     PERFORM DETACH-ROUTINE THRU DETACH-EXIT.                    10/01/87
 380  062500     CLOSE CMNFIL DSPFIL                                         10/08/87
      062600           QPRINT.                                               10/01/87
 381  062700     MOVE "0" TO ERR-SW.                                         10/01/87
      062800 ERROR-RECOVERY-EXIT.                                            10/01/87
      062900     EXIT.                                                       10/01/87
      063000*************************************************************** 02/21/89
      063100*                                                             * 02/21/89
      063200* EXIT-FORMAT-ERR IS BRANCHED TO AFTER A READ TO CMNFIL.  THE  * 02/21/89
      063300* RCD-FMT-NME RETURNED IN THE I-O-FEEDBACK AREA DOES NOT MATCH * 02/21/89
      063400* THE FORMAT EXPECTED BY THE PROGRAM.  AN ERROR MESSAGE IS     * 02/21/89
      063500* PRINTED AND THE PROGRAM ENDS.                               * 02/21/89
      063600*                                                             * 02/21/89
      063700*************************************************************** 02/21/89
      063800* 18                                                           10/14/87
 382  063900 EXIT-FORMAT-ERR.                                                10/01/87
5728CB1 R01 M02 881028              COBOL SOURCE LISTING          ICFLIB/CSFMUL          03/21/89 08:22:28       Page   20
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
 383  064000     MOVE MAJ-MIN TO RC.                                         01/14/88
 384  064100     MOVE "RECORD FORMAT IS INCORRECT ON READ           "        10/01/87
      064200        TO ERRMSG.                                               01/14/88
 385  064300     WRITE PRINTREC.                                             10/01/87
 386  064400     CLOSE  CMNFIL DSPFIL QPRINT.                                10/08/87
 387  064500     STOP RUN.                                                   10/01/87
      064600*                                                                10/01/87
      064700*************************************************************** 02/21/89
      064800*                                                             * 02/21/89
      064900*    THIS SECTION OF THE PROGRAM IS CALLED TO END             * 02/21/89
      065000*    THE TRANSACTION WITH THE REMOTE SYSTEM.                  * 02/21/89
      065100*                                                             * 02/21/89
      065200*************************************************************** 02/21/89
      065300* 19                                                           10/14/87
      065400 DETACH-ROUTINE.                                                 10/01/87
 388  065500     MOVE 0 TO LNGTH OF ITMREQ.                                  02/22/89
 389  065600     MOVE "ICF00   " TO PGM-DEV-NME                              10/01/87
 390  065700     WRITE CMNREC FORMAT IS "$$SENDET"                           10/01/87
      065800           TERMINAL IS PGM-DEV-NME.                              10/01/87
 391  065900     MOVE "ICF01   " TO PGM-DEV-NME                              10/01/87
 392  066000     WRITE CMNREC FORMAT IS "$$SENDET"                           10/01/87
      066100           TERMINAL IS PGM-DEV-NME.                              10/01/87
 393  066200     MOVE "ICF02   " TO PGM-DEV-NME                              10/01/87
```

Figure 10-23 (Part 12 of 13). Source Program Example — CSFMUL (System-Supplied Formats)

```
 394  066300     WRITE CMNREC FORMAT IS "$$SENDET"                              10/01/87
      066400           TERMINAL IS PGM-DEV-NME.                                 10/01/87
 395  066500     MOVE "ICF03   " TO PGM-DEV-NME                                 10/01/87
 396  066600     WRITE CMNREC FORMAT IS "$$SENDET"                              10/01/87
      066700           TERMINAL IS PGM-DEV-NME.                                 10/01/87
      066800 DETACH-EXIT.                                                       10/01/87
      066900     EXIT.                                                          10/01/87
      067000*                                                                   10/01/87
      067100*************************************************************       02/21/89
      067200*                                                           *       02/21/89
      067300*    THIS SECTION OF THE PROGRAM IS CALLED TO RELEASE THE PROGRAM  *  02/21/89
      067400*    DEVICES, END THE SESSION AND END THE PROGRAM.          *       02/21/89
      067500*                                                           *       02/21/89
      067600*************************************************************       02/21/89
      067700* 20                                                                10/14/87
      067800*                                                                   10/01/87
 397  067900 END-JOB.                                                           10/01/87
 398  068000     DROP "ICF00   " FROM CMNFIL.                                   10/08/87
 399  068100     DROP "ICF01   " FROM CMNFIL.                                   10/08/87
 400  068200     DROP "ICF02   " FROM CMNFIL.                                   10/08/87
 401  068300     DROP "ICF03   " FROM CMNFIL.                                   10/08/87
 402  068400     CLOSE  CMNFIL DSPFIL QPRINT.                                   10/08/87
 403  068500     STOP RUN.                                                      10/01/87
      068600*                                                                   10/01/87
                        * * * * *  E N D  O F  S O U R C E  * * * * *
5728CB1 R01 M02 881028                  COBOL MESSAGES              ICFLIB/CSFMUL          03/21/89 08:22:28       Page   21
 STMT
*   71  MSGID: LBL0600  SEVERITY: 10  SEQNBR:  009800
        Message . . . . :   No OUTPUT fields found for format CIMENU.
*   71  MSGID: LBL0600  SEVERITY: 10  SEQNBR:  009800
        Message . . . . :   No OUTPUT fields found for format DTLMNU.
*   71  MSGID: LBL0600  SEVERITY: 10  SEQNBR:  009800
        Message . . . . :   No OUTPUT fields found for format ITMMNU.
*   71  MSGID: LBL0600  SEVERITY: 10  SEQNBR:  009800
        Message . . . . :   No OUTPUT fields found for format TIMOUT.
                                    MESSAGE SUMMARY
  TOTAL    INFO(0-4)    WARNING(5-19)    ERROR(20-29)    SEVERE(30-39)    TERMINAL(40-99)
     4           0              4               0               0                0
                  * * * * *  E N D  O F  C O B O L  M E S S A G E S  * * * * *
 686 source records read
 99 copy records read
 1 copy members processed
 0 sequence errors
 10 was the highest severity message issued
  LBL0901 00  Program CSFMUL created in library ICFLIB.
                        * * * * *  E N D  O F  C O M P I L A T I O N  * * * * *
```

Figure 10-23 (Part 13 of 13). Source Program Example — CSFMUL (System-Supplied Formats)

Target Program Multiple-Session Inquiry (Example II)

The following describes the COBOL target program for multiple-session inquiry program example.

Program Files: The COBOL multiple-session target program uses the following files:

CFILE	A ICF file used to send records to and receive records from the source program. It is done with the file-level INDARA DDS keyword, indicating a separate indicator area.
PFILE	A database file used to retrieve the record for the item requested from the remote system.
QPRINT	A printer file used to print error messages resulting from communications errors.

DDS Source: The DDS for the ICF file (CFILE) is illustrated in Figure 10-24.

```
5714PW1 R01M00  880301                SEU SOURCE LISTING                    10/14/87 17:20:35          PAGE 1
SOURCE FILE . . . . . .  QICFPUB/ICFLIB
MEMBER  . . . . . . . .  CFILE
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
  100      A*****************************************************************                      10/14/87
  200      A*                                                               *                      10/14/87
  300      A*                          ICF FILE                             *                      10/14/87
  400      A*            USED IN TARGET MULTIPLE SESSION PROGRAM            *                      10/14/87
  500      A*                                                               *                      10/14/87
  600      A*****************************************************************                      10/14/87
  700      A                                    INDARA                                             08/04/87
  800      A  05                                RQSWRT                                             08/04/87
  900      A  10                                ALWWRT                                             08/04/87
 1000      A                                    INDTXT(10 '10 END TRANS.')                         08/04/87
 1100      A  15                                EOS                                                08/04/87
 1200      A  20                                FAIL                                               08/06/87
 1300      A                                    INDTXT(20 '20 F ABORT ST')                         08/06/87
 1400      A                                    RCVFAIL(25 'RECEIVED FAIL')                        08/04/87
 1500      A  30                                DETACH                                             08/06/87
 1600      A                                    INDTXT(30 '30>DETACH TGT')                         08/06/87
 1700      A                                    RCVDETACH(44 'RECV DETACH')                        08/04/87
 1800      A          R SNDPART                                                                    08/04/87
 1900      A                                    INVITE                                             08/14/87
 2000      A            RECTYP         1                                                         10/01/87
 2100      A            ITEMNO         6                                                         10/08/87
 2200      A            EDATA        130                                                         08/04/87
 2300      A            FILL1         13                                                         10/08/87
 2400      A          R RCVPART                                                                    08/04/87
 2500      A            RECID2         6                                                         10/08/87
 2600      A            PARTDS        80                                                         10/08/87
 2700      A            FILL4         64                                                         08/04/87
 2800      A          R RCVTRND                                                                    08/07/87
 2900      A                                    RCVTRNRND(40 'END OF TRN')                         08/07/87
                              * * * *  E N D  O F  S O U R C E  * * * *
```

Figure 10-24. DDS Source for ICF File Used in Target Program Multiple Session Inquiry

The DDS source for the database file (PFILE) is illustrated in Figure 10-25.

```
5714PW1 R01M00  880301                SEU SOURCE LISTING                    10/16/87 07:43:14          PAGE 1
SOURCE FILE . . . . . .  QICFPUB/ICFLIB
MEMBER  . . . . . . . .  PFILE
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
  100      A                                    LIFO                                               07/02/87
  200      A          R DBREC                                                                      05/06/87
  300      A            RECCUS         1                                                         10/01/87
  400      A            DBSEQ          6                                                         08/18/87
  500      A            DBDATA       130                                                         07/02/87
  600      A            DBFILL        13                                                         10/01/87
  700      A          K DBSEQ                                                                      07/04/87
                              * * * *  E N D  O F  S O U R C E  * * * *
```

Figure 10-25. DDS Source for ICF File Used in Target Program Multiple Session Inquiry

The command needed to create the ICF file is:

```
CRTICFF  FILE(ICFLIB/CFILE)  SRCFILE(ICFLIB/QICFPUB)  SRCMBR(CFILE)
   ACQPGMDEV(RQSDEV)  TEXT("TARGET ICF FILE FOR MULTIPLE SESSION PROGRAM")
```

The command needed to define the program device entry is:

```
OVRICFDEVE PGMDEV(RQSDEV) RMTLOCNAME(*REQUESTER)
```

Program Explanation: The following explains the structure of the program examples illustrated in Figure 10-26 on page 10-80 and Figure 10-27 on page 10-85. The ICF file used in the first example is defined by the user, and uses externally described data formats (DDS). The second example uses the same file, but uses program-described data and system-supplied formats. The reference letters in the example below correspond to those in the following program examples.

Although the basic structure of the two examples provided is the same, there are differences because of the way the user-defined formats and the system-supplied formats are used. All output operations to the ICF file in the first example are done using the the WRITE statement with the record format name coded as operand. The output operations to the ICF file in the second example using system-supplied formats are issued with the name of the system-supplied format coded as a literal operand.

Differences between the first and second example are described as notes in each of the following descriptions where necessary.

1 This section defines the ICF file (CFILE) and the database file (PFILE) used in the program.

CFILE is the ICF file used to send records to and receive records from the remote system.

MAJ-MIN is the variable name used to check for the ICF file return codes.

CMNF-INDIC-AREA is the indicator area used with the ICF file to choose options on DDS keywords and operations, and receive response indicators on input operations.

Note: In the program using system-supplied formats, the input records for CFILE are explicitly coded in the program since CFILE is now treated as a program-described file. The system-supplied file, QICDMF, could have been used instead of CFILE. Using system-supplied files can be done by specifying QICDMF in the file specification, or by using an OVRICFF command to change the file name from CFILE to QICDMF.

2 This section defines the error handling for the program. The routine first checks the major/minor return code to determine if the error is recoverable. If return code 3431 is received, it is saved, and control is passed back to the main calling program. Return code 3431 is not considered a serious error in this example. The program then exits the declaratives.

If any other error has occurred, the program prints a message saying that the program ended abnormally, and then ends.

3 This routine opens all the files.

Because the ICF file was created using the ACQPGMDEV parameter, the target session is automatically acquired when the file is opened.

4 This routine reads data from the program device (CFILE) through a perform statement until a change direction is received. The program then goes to 5 to read the database file. When change direction is received, indicator 40 is set on, as defined by the RCVTRNRND DDS keyword in the DDS source file for ICF file.

Note: In the program using system-supplied formats, a minor return code of '00' is checked to determine if change direction has been received.

5 The program uses the requested number received from the source program to access the record from the database. The information retrieved from the database file (PFILE) is moved to the work area for the ICF file. A write operation is issued to the program device using record format SNDPART. The write operation sends the requested information back to the source program.

If the requested number is not found, zero is propagated into the field.

If an error occurs on the write operation, control passes to 2.

If no error occurs on the write, control goes back to 4.

Note: In the program using the system-supplied format, the WRITE statement uses the $$SEND format to send the data.

6 A read operation is issued to the program device.

If a detach indication is received, the program goes to 8 to end the program. When a detach is received, indicator 44 is set on, as defined by the RCVDETACH keyword in the DDS for the ICF file.

Note: In the system-supplied format example, the read operation is sent without using a record format name. Also, a minor return code of '08' is checked for the detach received condition.

7 This routine is called to issue the read operation to the the program device until the RCVTRNRND indication is received.

Note: In the system-supplied format example, there is no 7. Instead, a minor return code of '00' is checked for the turnaround indication in 4.

8 This routine is called to end the program.

The following message is written to the print file:

```
CTDMUL HAS COMPLETED NORMALLY
```

The files are closed. The program device is automatically released as a result of the close operation, and the program ends.

Note: In the program using system-supplied formats, the following message is printed:

```
CTFMUL HAS COMPLETED NORMALLY
```

```
5728CB1 R01 M02 881028              IBM AS/400  COBOL/400          ICFLIB/CTDMUL          03/01/89 13:41:44       Page     1
Program name . . . . . . . . . . . . : CTDMUL in ICFLIB
Source file  . . . . . . . . . . . . : QICFPUB in ICFLIB     Member - CTDMUL     02/28/89 13:48:10
Compiler option  . . . . . . . . . . : *NONE
Code generation option . . . . . . . : *NONE
Code generation severity level . . . : 29
Print file . . . . . . . . . . . . . : QSYSPRT in *LIBL
FIPS flagging option . . . . . . . . : *NOFIPS *NOSEG *NODEB *NOOBSOLETE
Text not available for message LBX6039 file QLBLMSG.
Flagging level . . . . . . . . . . . : 0
Replace existing program . . . . . . : *YES
User profile . . . . . . . . . . . . : *USER
Authority  . . . . . . . . . . . . . : *CHANGE
Text . . . . . . . . . . . . . . . . : CBL Multiple Session Inquiry - Target DDS
Compiler . . . . . . . . . . . . . . : IBM AS/400  COBOL/400
5728CB1 R01 M02 881028              COBOL SOURCE LISTING          ICFLIB/CTDMUL          03/01/89 13:41:44       Page     2
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
    1  000100 IDENTIFICATION DIVISION.                                                              10/01/87
    2  000200 PROGRAM-ID.                 CTDMUL.                                                   10/01/87
       000300*************************************************************************              10/01/87
       000400*  THIS PROGRAM WILL HANDLE THE REQUEST FOR EITHER A CUSTOMER      *                   10/01/87
       000500*  NUMBER OR AN ITEM NUMBER.  THIS IS ACCOMPLISHED BY MAKING       *                   10/01/87
       000600*  THE DATA BASE FILE STRUCTURE (KEY LENGTH, KEY POSITION, RECORD  *                   10/01/87
       000700*  LENGTH, RECORD SIZE, ETC.) THE SAME FOR BOTH FILES WITH ONLY    *                   10/01/87
       000800*  THE RECORD CONTENTS DIFFERENT.                                  *                   10/01/87
       000900*                                                                  *                   10/01/87
       001000*  THIS PROGRAM ENDS WHEN A DETACH REQUEST IS RECEIVED FROM        *                   10/01/87
       001100*  THE SOURCE PROGRAM.                                             *                   10/01/87
       001200*                                                                  *                   10/01/87
       001300*  INDICATORS ASSOCIATED WITH THE ICF FILE I/O OPERATION           *                   02/21/89
       001400*  ARE DECLARED IN THE WORKING-STORAGE SECTION AND ARE REFERENCED  *                   10/15/87
       001500*  FOR EVERY I/O OPERATION ISSUED.                                 *                   10/15/87
       001600*************************************************************************              10/01/87
    3  001700 ENVIRONMENT DIVISION.                                                                 10/01/87
    4  001800 CONFIGURATION SECTION.                                                                10/01/87
    5  001900 SOURCE-COMPUTER.           IBM-AS400.                                                 01/15/88
    6  002000 OBJECT-COMPUTER.           IBM-AS400.                                                 01/15/88
    7  002100 SPECIAL-NAMES.             I-O-FEEDBACK IS IO-FBA                                     10/01/87
    8  002200                            OPEN-FEEDBACK IS OPEN-FBA.                                 10/01/87
    9  002300 INPUT-OUTPUT SECTION.                                                                 10/01/87
       002400* 1                                                                                    10/01/87
   10  002500 FILE-CONTROL.                                                                         10/01/87
   11  002600     SELECT PFILE ASSIGN TO DATABASE-PFILE                                             10/01/87
   12  002700         ORGANIZATION IS INDEXED                                                       10/01/87
   13  002800         ACCESS IS RANDOM                                                              10/01/87
   14  002900         RECORD KEY IS EXTERNALLY-DESCRIBED-KEY                                        10/01/87
   15  003000            WITH DUPLICATES.                                                           10/01/87
   16  003100     SELECT CFILE ASSIGN TO WORKSTATION-CFILE-SI                                       10/01/87
   17  003200         ORGANIZATION IS TRANSACTION                                                   10/01/87
   18  003300         FILE STATUS IS STATUS-IND MAJ-MIN.                                            10/01/87
   19  003400     SELECT QPRINT ASSIGN TO PRINTER-QSYSPRT.                                          10/01/87
   20  003500 DATA DIVISION.                                                                        10/01/87
   21  003600 FILE SECTION.                                                                         10/01/87
   22  003700 FD  PFILE                                                                             10/01/87
   23  003800     LABEL RECORDS ARE STANDARD.                                                       10/01/87
   24  003900 01  PREC.                                                                            10/01/87
   25  004000     COPY DDS-ALL-FORMATS OF PFILE.                                                    10/01/87
   26 +000001       05  PFILE-RECORD PIC X(150).                          <-ALL-FMTS
      +000002*     I-O FORMAT:DBREC      FROM FILE PFILE      OF LIBRARY ICFLIB      <-ALL-FMTS
      +000003*                                                               <-ALL-FMTS
```

Figure 10-26 (Part 1 of 5). Target Program Example — CTDMUL (User-Defined Formats)

```
    +000004*THE KEY DEFINITIONS FOR RECORD FORMAT  DBREC                                  <-ALL-FMTS
    +000005*  NUMBER              NAME              RETRIEVAL     TYPE    ALTSEQ        <-ALL-FMTS
    +000006*   0001   DBSEQ                          ASCENDING      AN      NO          <-ALL-FMTS
 27 +000007       05  DBREC         REDEFINES PFILE-RECORD.                            <-ALL-FMTS
 28 +000008           06 RECCUS            PIC X(1).                                    <-ALL-FMTS
 29 +000009           06 DBSEQ             PIC X(6).                                    <-ALL-FMTS
 30 +000010           06 DBDATA            PIC X(130).                                  <-ALL-FMTS
 31 +000011           06 DBFILL            PIC X(13).                                   <-ALL-FMTS
 32  004100 FD  CFILE                                                                    10/01/87
 33  004200     LABEL RECORDS ARE STANDARD.                                              10/01/87
 34  004300 01  ICFREC.                                                                 10/01/87
 35  004400     COPY DDS-ALL-FORMATS-I-O OF CFILE.                                       10/01/87
5728CB1 R01 M02 881028               COBOL SOURCE LISTING            ICFLIB/CTDMUL      03/01/89 13:41:44      Page     3
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S COPYNAME   CHG/DATE
 36 +000001       05  CFILE-RECORD PIC X(150).                                         <-ALL-FMTS
    +000002*  INPUT FORMAT:SNDPART    FROM FILE CFILE      OF LIBRARY ICFLIB          <-ALL-FMTS
    +000003*                                                                          <-ALL-FMTS
 37 +000004       05  SNDPART-I     REDEFINES CFILE-RECORD.                           <-ALL-FMTS
 38 +000005           06 RECTYP            PIC X(1).                                   <-ALL-FMTS
 39 +000006           06 ITEMNO            PIC X(6).                                   <-ALL-FMTS
 40 +000007           06 EDATA             PIC X(130).                                 <-ALL-FMTS
 41 +000008           06 FILL1             PIC X(13).                                  <-ALL-FMTS
    +000009* OUTPUT FORMAT:SNDPART    FROM FILE CFILE      OF LIBRARY ICFLIB          <-ALL-FMTS
    +000010*                                                                          <-ALL-FMTS
 42 +000011       05  SNDPART-O     REDEFINES CFILE-RECORD.                           <-ALL-FMTS
 43 +000012           06 RECTYP            PIC X(1).                                   <-ALL-FMTS
 44 +000013           06 ITEMNO            PIC X(6).                                   <-ALL-FMTS
 45 +000014           06 EDATA             PIC X(130).                                 <-ALL-FMTS
 46 +000015           06 FILL1             PIC X(13).                                  <-ALL-FMTS
    +000016*  INPUT FORMAT:RCVPART    FROM FILE CFILE      OF LIBRARY ICFLIB          <-ALL-FMTS
    +000017*                                                                          <-ALL-FMTS
 47 +000018       05  RCVPART-I     REDEFINES CFILE-RECORD.                           <-ALL-FMTS
 48 +000019           06 RECID2            PIC X(6).                                   <-ALL-FMTS
 49 +000020           06 PARTDS            PIC X(80).                                  <-ALL-FMTS
 50 +000021           06 FILL4             PIC X(64).                                  <-ALL-FMTS
    +000022* OUTPUT FORMAT:RCVPART    FROM FILE CFILE      OF LIBRARY ICFLIB          <-ALL-FMTS
    +000023*                                                                          <-ALL-FMTS
 51 +000024       05  RCVPART-O     REDEFINES CFILE-RECORD.                           <-ALL-FMTS
 52 +000025           06 RECID2            PIC X(6).                                   <-ALL-FMTS
 53 +000026           06 PARTDS            PIC X(80).                                  <-ALL-FMTS
 54 +000027           06 FILL4             PIC X(64).                                  <-ALL-FMTS
    +000028*  INPUT FORMAT:RCVTRND    FROM FILE CFILE      OF LIBRARY ICFLIB          <-ALL-FMTS
    +000029*                                                                          <-ALL-FMTS
    +000030*      05  RCVTRND-I     REDEFINES CFILE-RECORD.                           <-ALL-FMTS
    +000031* OUTPUT FORMAT:RCVTRND    FROM FILE CFILE      OF LIBRARY ICFLIB          <-ALL-FMTS
    +000032*                                                                          <-ALL-FMTS
    +000033*      05  RCVTRND-O     REDEFINES CFILE-RECORD.                           <-ALL-FMTS
 55  004500 FD  QPRINT                                                               10/01/87
 56  004600     LABEL RECORDS ARE OMITTED.                                           10/01/87
 57  004700 01  PRINTREC.                                                           01/14/88
 58  004800     05  RC              PIC 9999.                                       01/15/88
 59  004900     05  ERRMSG          PIC X(128).                                     01/14/88
 60  005000 WORKING-STORAGE SECTION.                                                 10/01/87
 61  005100 77  MAJ-MIN-SAV                    PIC X(4).                            10/01/87
 62  005200 77  STATUS-IND                     PIC X(2).                            10/01/87
 63  005300 77  INDON                          PIC 1 VALUE B"1".                    10/01/87
 64  005400 77  INDOFF                         PIC 1 VALUE B"0".                    10/01/87
 65  005500 77  LEN                            PIC 9(10)V9(5) COMP                  10/01/87
 66  005600                                     VALUE 0.                             10/01/87
 67  005700 77  CMD2                           PIC X(31)                            10/01/87
```

Figure 10-26 (Part 2 of 5). Target Program Example — CTDMUL (User-Defined Formats)

```
   68  005800       VALUE "CPYF HEXDUMP *LIST PRTFMT(*HEX)".                                  10/01/87
   69  005900 01  CMNF-INDIC-AREA.                                                           10/01/87
       006000* ALLOW WRITE (ALWWRT) INDICATOR                                                10/01/87
   70  006100      05  IN10                        PIC 1 INDIC 10.                         10/01/87
   71  006200          88 IN10-ON                        VALUE B"1".                       10/01/87
   72  006300          88 IN10-OFF                       VALUE B"0".                       10/01/87
       006400* RECEIVE TURNAROUND (RCVTRNRND) INDICATOR                                     10/01/87
   73  006500      05  IN40                        PIC 1 INDIC 40.                         10/01/87
   74  006600          88 IN40-ON                        VALUE B"1".                       10/01/87
5728CB1 R01 M02 881028              COBOL SOURCE LISTING            ICFLIB/CTDMUL          03/01/89 13:41:44       Page     4
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
   75  006700          88 IN40-OFF                       VALUE B"0".                       10/01/87
       006800* RECEIVE DETACH (RCVDETACH) INDICATOR                                        10/01/87
   76  006900      05  IN44                        PIC 1 INDIC 44.                         10/01/87
   77  007000          88 IN44-ON                        VALUE B"1".                       10/01/87
   78  007100          88 IN44-OFF                       VALUE B"0".                       10/01/87
   79  007200 01  MAJ-MIN.                                                                 10/01/87
   80  007300      05  MAJ                         PIC X(2).                               10/01/87
   81  007400      05  MIN                         PIC X(2).                               10/01/87
5728CB1 R01 M02 881028              COBOL SOURCE LISTING            ICFLIB/CTDMUL          03/01/89 13:41:44       Page     5
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
       007500/                                                                             10/01/87
   82  007600 PROCEDURE DIVISION.                                                          10/01/87
       007700 DECLARATIVES.                                                                10/01/87
       007800 ERR-SECTION SECTION.                                                         10/01/87
       007900*************************************************************************     10/01/87
       008000* 2                                                                          10/01/87
       008100*                                                                            10/01/87
       008200     USE AFTER STANDARD ERROR PROCEDURE ON CFILE.                            10/01/87
       008300 CFILE-EXCEPTION.                                                             10/01/87
       008400*                                                                            10/01/87
       008500* CHECK THE MAJOR/MINOR CODES AND TAKE APPROPRIATE ACTION                    10/01/87
       008600*                                                                            10/01/87
       008700* MAJOR CODE 34 - INPUT EXCEPTION.                                           10/01/87
   83  008800     IF MAJ = "34"                                                           10/01/87
       008900* DATA TRUNCATED IN INPUT AREA.                                              10/01/87
   84  009000         IF MIN = "31"                                                       10/01/87
   85  009100            MOVE MAJ-MIN TO MAJ-MIN-SAV                                     10/12/87
   86  009200            GO TO EXIT-DECLARATIVES                                         10/12/87
       009300         ELSE                                                               10/12/87
   87  009400            GO TO EXIT-DECLARATIVES.                                        10/12/87
       009500*************************************************************************     10/01/87
       009600*                                                                       *    02/21/89
       009700* PRINT A MESSAGE SAYING CTDMUL PROGRAM ENDED ABNORMALLY.               *    02/21/89
       009800* CLOSE ALL THE FILES AND END THE PROGRAM. THIS ROUTINE IS CALLED       *    02/21/89
       009900* WHEN A NON-RECOVERABLE ERROR OCCURS IN ICF FILE.                      *    02/21/89
       010000*                                                                       *    02/21/89
       010100*************************************************************************     10/01/87
       010200 GETFBA.                                                                     10/01/87
   88  010300     MOVE MAJ-MIN TO RC.                                                     01/14/88
   89  010400     MOVE "CTDMUL HAS COMPLETED ABNORMALLY" TO ERRMSG.                       01/14/88
   90  010500     WRITE PRINTREC.                                                         10/01/87
   91  010600     CLOSE PFILE                                                             10/01/87
       010700           CFILE                                                            10/01/87
       010800           QPRINT.                                                          10/01/87
   92  010900     STOP RUN.                                                               10/01/87
       011000*                                                                            10/01/87
       011100 EXIT-DECLARATIVES.                                                           10/01/87
       011200     EXIT.                                                                   10/01/87
       011300*                                                                            10/01/87
```

Figure 10-26 (Part 3 of 5). Target Program Example — CTDMUL (User-Defined Formats)

```
  93  011400 END DECLARATIVES.                                                                  10/01/87
      011500*****************************************************************                   10/01/87
5728CB1 R01 M02 881028              COBOL SOURCE LISTING            ICFLIB/CTDMUL          03/01/89 13:41:44      Page     6
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
      011600/                                                                                   10/01/87
      011700 START-PROGRAM  SECTION.                                                            10/01/87
      011800*                                                                                  02/27/89
      011900 START-PROGRAM-PARAGRAPH.                                                           10/01/87
      012000* 3                                                                                10/01/87
  94  012100     OPEN OUTPUT QPRINT                                                            10/01/87
      012200          I-O   CFILE                                                              10/01/87
      012300          INPUT PFILE.                                                             10/01/87
      012400*****************************************************************                  10/01/87
      012500*                                                               *                  10/01/87
      012600*  READ THE REQUEST FROM THE SOURCE PROGRAM.  INDICATOR 40      *                  10/01/87
      012700*  INDICATES RCVTRNRND OCCURED.  INDICATOR 44 INDICATES THAT    *                  10/01/87
      012800*  DETACH INDICATOR HAS BEEN RECEIVED FROM THE REMOTE SYSTEM.   *                  10/01/87
      012900*                                                               *                  10/01/87
      013000*  THIS PROGRAM CHECKS FOR ERRORS ON EVERY ICF FILE             *                  01/25/88
      013100*  FILE OPERATION.  A MAJOR CODE GREATER THAN 03 INDICATES      *                  10/01/87
      013200*  AN ERROR.                                                    *                  10/01/87
      013300*                                                               *                  10/01/87
      013400*****************************************************************                  10/01/87
      013500* 4                                                                                10/01/87
      013600 RECEIVE-DATA.                                                                      10/01/87
  95  013700     PERFORM READ-CFILE THRU READ-CFILE-EXIT.                                      10/01/87
  96  013800     IF IN40-ON                                                                    10/01/87
  97  013900        GO TO SEND-DATA.                                                           10/01/87
  98  014000     PERFORM RCVTRNRND THRU RCVTRNRND-EXIT                                         10/01/87
      014100        UNTIL IN40-ON.                                                             10/01/87
      014200*****************************************************************                  10/01/87
      014300*                                                               *                  10/01/87
      014400*  A REQUEST FROM THE SOURCE PROGRAM RESULTS IN READING A SINGLE*                   10/01/87
      014500*  RECORD CONTAINING THE REQUESTED CUSTOMER OR ORDER NUMBER.  THE*                  10/01/87
      014600*  RESPONSE WILL BE RETURNED IN A SINGLE RECORD CONTAINING EITHER*                  10/01/87
      014700*  THE ITEM OR CUSTOMER INFORMATION, DEPENDING ON THE DATA BASE *                   10/01/87
      014800*  CONTENT.                                                     *                  10/01/87
      014900*                                                               *                  10/01/87
      015000*  THE RESPONSE IS SENT TO THE SOURCE PROGRAM BY WRITING TO THE *                   10/01/87
      015100*  PROGRAM DEVICE FILE USING FORMAT SNDPART.                    *                  10/15/87
      015200*                                                               *                  10/01/87
      015300*  WHEN THE REQUESTED CUSTOMER OR ITEM NUMBER IS NOT FOUND,     *                  10/01/87
      015400*  000000 IS PROPAGATED TO THE KEY FIELD BEFORE THE RESPONSE    *                  10/01/87
      015500*  IS SENT BACK TO THE SOURCE PROGRAM.                          *                  10/01/87
      015600*                                                               *                  02/22/89
      015700*****************************************************************                  10/01/87
      015800*                                                                                  10/01/87
      015900* 5                                                                                10/01/87
      016000 SEND-DATA.                                                                         10/01/87
  99  016100     MOVE RECID2 OF RCVPART-I TO DBSEQ.                                            10/01/87
 100  016200     READ PFILE INVALID KEY MOVE 0 TO DBSEQ.                                       10/01/87
 102  016300     MOVE RECCUS TO RECTYP OF SNDPART-O.                                           10/01/87
 103  016400     MOVE ZEROS TO CMNF-INDIC-AREA.                                                10/01/87
 104  016500     MOVE DBSEQ TO ITEMNO OF SNDPART-O.                                            10/01/87
 105  016600     MOVE DBDATA TO EDATA OF SNDPART-O                                             10/01/87
 106  016700     WRITE ICFREC FROM PREC FORMAT IS "SNDPART"                                    10/01/87
      016800        INDICATORS ARE CMNF-INDIC-AREA.                                           10/01/87
 107  016900     GO TO RECEIVE-DATA.                                                           10/01/87
      017000*****************************************************************                  10/01/87
```

Figure 10-26 (Part 4 of 5). Target Program Example — CTDMUL (User-Defined Formats)

```
5728CB1 R01 M02 881028              COBOL SOURCE LISTING            ICFLIB/CTDMUL          03/01/89 13:41:44       Page     7
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
      017100*                                                                 *                    10/01/87
      017200* THIS ROUTINE ISSUES READ OPERATION TO THE PROGRAM DEVICE.       *                    10/15/87
      017300* DETACH INDICATION IS CHECKED FOR AND IF IT OCCURED, THE         *                    10/01/87
      017400* PROGRAM IS ENDED (IN44-ON).                                     *                    10/01/87
      017500*                                                                 *                    10/01/87
      017600*******************************************************************                    10/01/87
      017700* 6                                                                                    10/01/87
      017800 READ-CFILE.                                                                           10/01/87
  108 017900     MOVE ZEROS TO CMNF-INDIC-AREA.                                                    10/01/87
  109 018000     READ CFILE FORMAT IS "RCVPART"                                                    10/01/87
      018100        INDICATORS ARE CMNF-INDIC-AREA.                                                10/01/87
  110 018200     SET IN40-ON TO TRUE.                                                              01/25/88
  111 018300     IF IN44-ON                                                                        10/01/87
  112 018400        GO TO END-PROGRAM.                                                             10/01/87
      018500 READ-CFILE-EXIT.                                                                      10/01/87
      018600     EXIT.                                                                             02/28/89
      018700*                                                                                     10/01/87
      018800*******************************************************************                    10/01/87
      018900*                                                                 *                    02/21/89
      019000* THIS ROUTINE READS THE ICF FILE UNTIL RCVTRNRND OCCURS          *                    02/21/89
      019100* DETACH INDICATION IS CHECKED FOR AND IF IT OCCURED, THE         *                    10/01/87
      019200* PROGRAM IS ENDED (IN44-ON).                                     *                    10/01/87
      019300*                                                                 *                    02/21/89
      019400*******************************************************************                    10/01/87
      019500* 7                                                                                    10/01/87
  113 019600 RCVTRNRND.                                                                            10/01/87
  114 019700     MOVE ZEROS TO CMNF-INDIC-AREA.                                                    10/01/87
  115 019800     READ CFILE FORMAT IS "RCVTRND"                                                    10/01/87
      019900        INDICATORS ARE CMNF-INDIC-AREA.                                                10/01/87
  116 020000     IF IN44-ON                                                                        10/01/87
  117 020100        GO TO END-PROGRAM.                                                             10/01/87
      020200 RCVTRNRND-EXIT.                                                                       10/01/87
      020300     EXIT.                                                                             02/28/89
      020400*                                                                                     10/01/87
      020500*******************************************************************                    10/01/87
      020600*                                                                 *                    02/21/89
      020700* ROUTINE TO END THE JOB AND CLOSE THE FILES.                     *                    02/21/89
      020800*                                                                 *                    02/21/89
      020900*******************************************************************                    10/01/87
      021000*                                                                                      10/01/87
      021100* 8                                                                                    10/14/87
  118 021200 END-PROGRAM.                                                                          10/01/87
  119 021300     MOVE MAJ-MIN TO RC.                                                               01/14/88
  120 021400     MOVE "CTDMUL HAS COMPLETED NORMALLY" TO ERRMSG.                                   01/14/88
  121 021500     WRITE PRINTREC.                                                                   10/01/87
  122 021600     CLOSE PFILE                                                                       10/01/87
      021700           CFILE                                                                       10/01/87
      021800           QPRINT.                                                                     10/01/87
  123 021900     STOP RUN.                                                                         10/01/87
                        * * * * *  E N D  O F  S O U R C E  * * * * *
5728CB1 R01 M02 881028                COBOL MESSAGES              ICFLIB/CTDMUL          03/01/89 13:41:44       Page     8
 STMT
*   35  MSGID: LBL0600  SEVERITY: 10  SEQNBR:  004400
      Message . . . . :   No INPUT fields found for format RCVTRND.
*   35  MSGID: LBL0600  SEVERITY: 10  SEQNBR:  004400
      Message . . . . :   No OUTPUT fields found for format RCVTRND.
                                    MESSAGE SUMMARY
 TOTAL    INFO(0-4)    WARNING(5-19)    ERROR(20-29)    SEVERE(30-39)    TERMINAL(40-99)
    2         0              2               0               0                0
                  * * * * *  E N D  O F  C O B O L  M E S S A G E S  * * * * *
219 source records read
44 copy records read
2 copy members processed
0 sequence errors
10 was the highest severity message issued
 LBL0901 00  Program CTDMUL created in library ICFLIB.
                        * * * * *  E N D  O F  C O M P I L A T I O N  * * * * *
```

Figure 10-26 (Part 5 of 5). Target Program Example — CTDMUL (User-Defined Formats)

```
5728CB1 R01 M02 881028              IBM AS/400  COBOL/400          ICFLIB/CTFMUL          03/01/89 13:43:42       Page    1
Program name . . . . . . . . . . . . : CTFMUL in ICFLIB
Source file  . . . . . . . . . . . . : QICFPUB in ICFLIB     Member - CTFMUL     02/28/89 13:50:33
Compiler option  . . . . . . . . . . : *NONE
Code generation option . . . . . . . : *NONE
Code generation severity level . . . : 29
Print file . . . . . . . . . . . . . : QSYSPRT in *LIBL
FIPS flagging option . . . . . . . . : *NOFIPS *NOSEG *NODEB *NOOBSOLETE
Text not available for message LBX6039 file QLBLMSG.
Flagging level . . . . . . . . . . . : 0
Replace existing program . . . . . . : *YES
User profile . . . . . . . . . . . . : *USER
Authority  . . . . . . . . . . . . . : *CHANGE
Text . . . . . . . . . . . . . . . . : CBL Multiple Session Inquiry - Target $$
Compiler . . . . . . . . . . . . . . : IBM AS/400  COBOL/400
5728CB1 R01 M02 881028              COBOL SOURCE LISTING          ICFLIB/CTFMUL          03/01/89 13:43:42       Page    2
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
    1  000100 IDENTIFICATION DIVISION.                                                              10/01/87
    2  000200 PROGRAM-ID.                CTFMUL.                                                    10/01/87
       000300*****************************************************************                    10/01/87
       000400*  THIS PROGRAM WILL HANDLE THE REQUEST FOR EITHER A CUSTOMER      *                   10/01/87
       000500*  NUMBER OR AN ITEM NUMBER.  THIS IS ACCOMPLISHED BY MAKING       *                   10/01/87
       000600*  THE DATA BASE FILE STRUCTURE (KEY LENGTH, KEY POSITION, RECORD  *                   10/01/87
       000700*  LENGTH, RECORD SIZE, ETC.) THE SAME FOR BOTH FILES WITH ONLY    *                   10/01/87
       000800*  THE RECORD CONTENTS DIFFERENT.                                  *                   10/01/87
       000900*                                                                  *                   10/01/87
       001000*  THIS PROGRAM ENDS WHEN A DETACH REQUEST IS RECEIVED FROM        *                   10/01/87
       001100*  THE SOURCE PROGRAM.                                             *                   10/01/87
       001200*                                                                  *                   10/01/87
       001300*  THIS PROGRAM USES THE SYSTEM-SUPPLIED FORMAT TO ISSUE THE I/O   *                   10/01/87
       001400*  OPERATION AND THEREFORE, DOES NOT USE THE OPTION INDICATORS     *                   10/01/87
       001500*  ASSOCIATED WITH THE KEYWORDS. NOTICE THAT THE ICF FILE          *                   02/21/89
       001600*  FILE DECLARATION SELECT STATEMENT REFLECTS THE USE OF A         *                   10/15/87
       001700*  SEPARATE INDICATOR AREA FOR INDICATORS.                         *                   10/15/87
       001800*****************************************************************                    10/01/87
    3  001900 ENVIRONMENT DIVISION.                                                                 10/01/87
    4  002000 CONFIGURATION SECTION.                                                                10/01/87
    5  002100 SOURCE-COMPUTER.           IBM-AS400.                                                 01/15/88
    6  002200 OBJECT-COMPUTER.           IBM-AS400.                                                 01/15/88
    7  002300 SPECIAL-NAMES.             I-O-FEEDBACK IS IO-FBA                                     10/01/87
    8  002400                            OPEN-FEEDBACK IS OPEN-FBA.                                 10/01/87
    9  002500 INPUT-OUTPUT SECTION.                                                                 10/01/87
       002600* 1                                                                                    10/01/87
   10  002700 FILE-CONTROL.                                                                         10/01/87
   11  002800     SELECT PFILE ASSIGN TO DATABASE-PFILE                                             10/01/87
   12  002900         ORGANIZATION IS INDEXED                                                       10/01/87
   13  003000         ACCESS IS RANDOM                                                              10/01/87
   14  003100         RECORD KEY IS EXTERNALLY-DESCRIBED-KEY                                        10/01/87
   15  003200            WITH DUPLICATES.                                                           10/01/87
   16  003300     SELECT CFILE ASSIGN TO WORKSTATION-CFILE-SI                                       10/01/87
   17  003400         ORGANIZATION IS TRANSACTION                                                   10/01/87
   18  003500         FILE STATUS IS STATUS-IND MAJ-MIN.                                            10/01/87
   19  003600     SELECT QPRINT ASSIGN TO PRINTER-QSYSPRT.                                          10/01/87
   20  003700 DATA DIVISION.                                                                        10/01/87
   21  003800 FILE SECTION.                                                                         10/01/87
   22  003900 FD  PFILE                                                                             10/01/87
   23  004000     LABEL RECORDS ARE STANDARD.                                                       10/01/87
   24  004100 01  PREC.                                                                            10/01/87
   25  004200     COPY DDS-ALL-FORMATS OF PFILE.                                                    10/01/87
   26 +000001       05  PFILE-RECORD PIC X(150).                          <-ALL-FMTS
```

Figure 10-27 (Part 1 of 5). Target Program Example — CTFMUL (System-Supplied Formats)

```
      +000002*    I-O FORMAT:DBREC      FROM FILE PFILE      OF LIBRARY ICFLIB           <-ALL-FMTS
      +000003*                                                                           <-ALL-FMTS
      +000004*THE KEY DEFINITIONS FOR RECORD FORMAT  DBREC                               <-ALL-FMTS
      +000005*  NUMBER              NAME              RETRIEVAL     TYPE    ALTSEQ       <-ALL-FMTS
      +000006*   0001   DBSEQ                         ASCENDING      AN      NO         <-ALL-FMTS
   27 +000007       05  DBREC         REDEFINES PFILE-RECORD.                          <-ALL-FMTS
   28 +000008           06 RECCUS            PIC X(1).                                  <-ALL-FMTS
   29 +000009           06 DBSEQ             PIC X(6).                                  <-ALL-FMTS
   30 +000010           06 DBDATA            PIC X(130).                                <-ALL-FMTS
   31 +000011           06 DBFILL            PIC X(13).                                 <-ALL-FMTS
   32  004300 FD  CFILE                                                                          10/01/87
   33  004400     LABEL RECORDS ARE STANDARD.                                                    10/01/87
5728CB1 R01 M02 881028              COBOL SOURCE LISTING            ICFLIB/CTFMUL        03/01/89 13:43:42      Page     3
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME  CHG/DATE
   34  004500 01  ICFREC.                                                                    10/01/87
   35  004600     05  SNDPART.                                                               10/01/87
   36  004700         10  LNGTH                 PIC 9(4).                                  02/22/89
   37  004800         10  RECTYP                 PIC X.                                    10/01/87
   38  004900         10  ITEMNO                 PIC X(6).                                 10/01/87
   39  005000         10  EDATA                  PIC X(130).                               10/01/87
   40  005100         10  FILL1                  PIC X(13).                                10/01/87
   41  005200     05  RCVPART REDEFINES SNDPART.                                            10/01/87
   42  005300         10  RECID2                 PIC 9(6).                                 10/01/87
   43  005400         10  PARTDS                 PIC X(80).                                10/01/87
   44  005500         10  FILL4                  PIC X(64).                                10/01/87
   45  005600 FD  QPRINT                                                                     10/01/87
   46  005700     LABEL RECORDS ARE OMITTED.                                                 10/01/87
   47  005800 01  PRINTREC.                                                                 01/14/88
   48  005900     05  RC                         PIC 9999.                                 01/14/88
   49  006000     05  ERRMSG                     PIC X(128).                               01/14/88
   50  006100 WORKING-STORAGE SECTION.                                                    10/01/87
   51  006200 77  MAJ-MIN-SAV                   PIC X(4).                                 10/01/87
   52  006300 77  STATUS-IND                    PIC X(2).                                 10/01/87
   53  006400 77  INDON                         PIC 1 VALUE B"1".                         10/01/87
   54  006500 77  INDOFF                        PIC 1 VALUE B"0".                         10/01/87
   55  006600 77  LEN                           PIC 9(10)V9(5) COMP                       10/01/87
   56  006700                                    VALUE 0.                                  10/01/87
   57  006800 77  CMD2                          PIC X(31)                                 10/01/87
   58  006900       VALUE "CPYF HEXDUMP *LIST PRTFMT(*HEX)".                               10/01/87
       007000*                                                                            10/08/87
   59  007100 01  CMNF-INDIC-AREA.                                                        10/01/87
       007200* RECEIVE TURNAROUND (RCVTRNRND) INDICATOR                                   10/01/87
   60  007300     05  IN40                       PIC 1 INDIC 40.                           10/01/87
   61  007400         88 IN40-ON                      VALUE B"1".                         10/01/87
   62  007500         88 IN40-OFF                     VALUE B"0".                         10/01/87
       007600* RECEIVE DETACH (RCVDETACH) INDICATOR                                       10/01/87
   63  007700     05  IN44                       PIC 1 INDIC 44.                           10/01/87
   64  007800         88 IN44-ON                      VALUE B"1".                         10/01/87
   65  007900         88 IN44-OFF                     VALUE B"0".                         10/01/87
       008000*                                                                            10/08/87
   66  008100 01  MAJ-MIN.                                                               10/01/87
   67  008200     05  MAJ                        PIC X(2).                                 10/01/87
   68  008300     05  MIN                        PIC X(2).                                 10/01/87
5728CB1 R01 M02 881028              COBOL SOURCE LISTING            ICFLIB/CTFMUL        03/01/89 13:43:42      Page     4
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME  CHG/DATE
       008400/                                                                          10/01/87
   69  008500 PROCEDURE DIVISION.                                                     10/01/87
       008600 DECLARATIVES.                                                           10/01/87
       008700 ERR-SECTION SECTION.                                                    10/01/87
       008800*************************************************************************10/01/87
```

Figure 10-27 (Part 2 of 5). Target Program Example — CTFMUL (System-Supplied Formats)

```
    008900* 2                                                                  10/01/87
    009000*                                                                    10/01/87
    009100     USE AFTER STANDARD ERROR PROCEDURE ON CFILE.                    10/01/87
    009200 CFILE-EXCEPTION.                                                    10/01/87
    009300*                                                                    10/01/87
    009400* CHECK THE MAJOR/MINOR CODES AND TAKE APPROPRIATE ACTION            10/01/87
    009500* MAJOR CODE 34 - INPUT EXCEPTION.                                   10/08/87
    009600*                                                                    10/01/87
 70 009700     IF MAJ = "34"                                                   10/01/87
    009800* DATA TRUNCATED IN INPUT AREA.                                      10/01/87
 71 009900         IF MIN = "31"                                               10/01/87
 72 010000            MOVE MAJ-MIN TO MAJ-MIN-SAV                              10/10/87
 73 010100            GO TO EXIT-DECLARATIVES                                  10/10/87
    010200         ELSE                                                        10/10/87
 74 010300            GO TO EXIT-DECLARATIVES.                                 10/10/87
    010400*                                                                    10/08/87
    010500**************************************************************       10/01/87
    010600*                                                           *        02/21/89
    010700* PRINT A MESSAGE SAYING CTDMUL PROGRAM ENDED ABNORMALLY.   *        02/21/89
    010800* CLOSE ALL THE FILES AND END THE PROGRAM. THIS ROUTINE IS CALLED *   02/21/89
    010900* WHEN A NON-RECOVERABLE ERROR OCCURS IN THE ICF FILE.      *        02/21/89
    011000*                                                           *        02/21/89
    011100**************************************************************       10/01/87
    011200*                                                                    10/08/87
    011300 GETFBA.                                                             10/01/87
 75 011400     MOVE MAJ-MIN TO RC.                                             01/14/88
 76 011500     MOVE "CTFMUL HAS COMPLETED ABNORMALLY" TO ERRMSG.               01/14/88
 77 011600     WRITE PRINTREC.                                                 10/01/87
 78 011700     CLOSE PFILE                                                     10/01/87
    011800           CFILE                                                    10/01/87
    011900           QPRINT.                                                  10/01/87
 79 012000     STOP RUN.                                                       10/01/87
    012100*                                                                    10/01/87
    012200 EXIT-DECLARATIVES.                                                  10/01/87
    012300     EXIT.                                                           10/01/87
    012400*                                                                    10/01/87
 80 012500 END DECLARATIVES.                                                   10/01/87
    012600**************************************************************       10/01/87
5728CB1 R01 M02 881028              COBOL SOURCE LISTING          ICFLIB/CTFMUL      03/01/89 13:43:42      Page     5
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
    012700/                                                                    10/01/87
    012800 START-PROGRAM  SECTION.                                             10/01/87
    012900 START-PROGRAM-PARAGRAPH.                                            10/01/87
    013000* 3                                                                 10/01/87
 81 013100     OPEN OUTPUT QPRINT                                              10/01/87
    013200          I-O   CFILE                                               10/01/87
    013300          INPUT PFILE.                                              10/01/87
    013400**************************************************************       10/01/87
    013500*                                                           *        10/01/87
    013600*  READ THE REQUEST FROM THE SOURCE PROGRAM. MINOR RETURN CODE '00'*  10/01/87
    013700*  INDICATES RCVTRNRND OCCURED. MINOR RETURN CODE OF '08'   *        02/21/89
    013800*  INDICATES DETACH HAS BEEN RECEIVED.                      *        10/15/87
    013900*                                                           *        10/01/87
    014000*  THIS PROGRAM CHECKS FOR ERRORS ON EVERY ICF FILE I/O     *        02/22/89
    014100*  OPERATION.  A MAJOR CODE GREATER THAN 03 INDICATES AN ERROR.*      02/22/89
    014200*                                                           *        10/01/87
    014300**************************************************************       10/01/87
    014400* 4                                                                 10/01/87
    014500 RECEIVE-DATA.                                                       10/01/87
 82 014600     MOVE SPACES TO MAJ-MIN.                                         10/10/87
```

Figure 10-27 (Part 3 of 5). Target Program Example — CTFMUL (System-Supplied Formats)

```
 83  014700     PERFORM READ-CFILE THRU READ-CFILE-EXIT UNTIL                    10/01/87
     014800        MIN IS EQUAL TO "00".                                         10/01/87
     014900*************************************************************        10/01/87
     015000*                                                           *        10/01/87
     015100*  A REQUEST FROM THE SOURCE PROGRAM RESULTS IN READING A SINGLE   *   10/01/87
     015200*  RECORD CONTAINING THE REQUESTED CUSTOMER OR ORDER NUMBER.  THE  *   10/01/87
     015300*  RESPONSE WILL BE RETURNED IN A SINGLE RECORD CONTAINING EITHER  *   10/01/87
     015400*  THE ITEM OR CUSTOMER INFORMATION, DEPENDING ON THE DATA BASE    *   10/01/87
     015500*  CONTENT.                                                        *   10/01/87
     015600*                                                                  *   10/01/87
     015700*  THE RESPONSE IS SENT TO THE SOURCE PROGRAM BY WRITING TO THE    *   10/01/87
     015800*  PROGRAM DEVICE FILE USING FORMAT SNDPART.                       *   10/15/87
     015900*                                                                  *   10/01/87
     016000*  WHEN THE REQUESTED CUSTOMER OR ITEM NUMBER IS NOT FOUND,        *   10/01/87
     016100*  000000 IS PROPAGATED TO THE KEY FIELD BEFORE THE RESPONSE       *   10/01/87
     016200*  IS SENT BACK TO THE SOURCE PROGRAM.                             *   10/01/87
     016300*                                                                  *   02/22/89
     016400*************************************************************        10/01/87
     016500*                                                                    10/01/87
     016600* 5                                                                  10/01/87
     016700 SEND-DATA.                                                          10/01/87
 84  016800     MOVE RECID2 OF RCVPART TO DBSEQ.                                10/01/87
 85  016900     READ PFILE INVALID KEY MOVE 000000 TO DBSEQ.                    10/01/87
 87  017000     MOVE RECCUS TO RECTYP.                                          10/01/87
 88  017100     MOVE 150 TO LNGTH OF SNDPART.                                   02/22/89
 89  017200     MOVE DBSEQ TO ITEMNO OF SNDPART.                                10/01/87
 90  017300     MOVE DBDATA TO EDATA OF SNDPART.                                10/01/87
 91  017400     WRITE ICFREC FORMAT IS "$$SEND".                                10/01/87
 92  017500     GO TO RECEIVE-DATA.                                             10/01/87
     017600*************************************************************        10/01/87
     017700*                                                           *        02/21/89
     017800* THIS ROUTINE ISSUES THE READ OPERATION THE PROGRAM DEVICE UNTIL  *   02/21/89
     017900* RCVTRNRND OCCURS.                                               *   02/21/89
     018000* DETACH INDICATION IS CHECKED FOR AND IF IT OCCURED, THE         *   10/01/87
     018100* PROGRAM IS ENDED (RC=__08).                                     *   10/01/87
5728CB1 R01 M02 881028              COBOL SOURCE LISTING          ICFLIB/CTFMUL          03/01/89 13:43:42       Page    6
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
     018200*                                                           *        02/21/89
     018300*************************************************************        10/15/87
     018400* 6                                                                  10/01/87
     018500 READ-CFILE.                                                         10/01/87
 93  018600     READ CFILE                                                      10/01/87
     018700        INDICATORS ARE CMNF-INDIC-AREA.                              10/01/87
 94  018800     IF MIN = "08"                                                   10/01/87
 95  018900        GO TO END-PROGRAM.                                           10/01/87
     019000 READ-CFILE-EXIT.                                                    10/01/87
     019100     EXIT.                                                           02/28/89
     019200*                                                                   10/01/87
     019300*************************************************************        10/01/87
     019400*                                                           *        02/21/89
     019500* ROUTINE TO END THE JOB AND CLOSE THE FILES.               *        02/21/89
     019600*                                                           *        02/21/89
     019700*************************************************************        10/01/87
     019800*                                                                    10/01/87
     019900*(H)                                                                 10/14/87
 96  020000 END-PROGRAM.                                                        10/01/87
 97  020100     MOVE MAJ-MIN TO RC.                                             01/14/88
 98  020200     MOVE "CTFMUL HAS COMPLETED NORMALLY" TO ERRMSG.                 01/14/88
 99  020300     WRITE PRINTREC.                                                 10/01/87
100  020400     CLOSE PFILE                                                     10/01/87
```

Figure 10-27 (Part 4 of 5). Target Program Example — CTFMUL (System-Supplied Formats)

```
      020500           CFILE                                                                  10/01/87
      020600           QPRINT.                                                                10/01/87
  101 020700       STOP RUN.                                                                  10/01/87
                        * * * * *  E N D  O F  S O U R C E  * * * * *
5728CB1 R01 M02 881028                  COBOL MESSAGES              ICFLIB/CTFMUL          03/01/89 13:43:42       Page    7
 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  * * * * *
207 source records read
11 copy records read
1 copy members processed
0 sequence errors
0 was the highest severity message issued
 LBL0901 00  Program CTFMUL created in library ICFLIB.
                         * * * * *  E N D  O F  C O M P I L A T I O N  * * * * *
```

Figure 10-27 (Part 5 of 5). Target Program Example — CTFMUL (System-Supplied Formats)

Chapter 11. Communications Applications with RPG/400

Previous chapters in this manual describe the functions provided by ICF. This chapter introduces you to the RPG/400 interface for ICF and provides program examples.

Two program examples are presented in this chapter. For each example, both the source and target programs are provided. Each program is written first with user-defined formats (data description specifications or DDS) and then with system-supplied formats.

The first example is a batch data transfer application using a single session. The second example is a multiple-session inquiry application using one display file and four ICF sessions.

Not all programming considerations or techniques are illustrated in each example in this chapter. Review these examples and the examples provided in the appropriate programming manual before beginning application design and coding.

Note: The examples in this chapter were written to the APPC communications type. Minor changes might be required if another communications type is used.

Introduction to the RPG/400 Interface

Before you write an RPG/400 communications application, you must understand the high-level language interface provided by RPG/400. ICF files are defined as WORKSTN files in RPG/400.

The operations you use in the communications portion of your program are similar to work station operations. In the noncommunications portion of your program, you can use all noncommunications operations you normally use to process data that is sent or received between your program and the remote program.

Table 11-1 briefly introduces the RPG/400 statements you use in the communications portion of your program.

Table 11-1. RPG/400 Statements

ICF Operation	RPG/400 Operation Code	Function
Open	OPEN	Opens the ICF file
Acquire	ACQ	Acquires a program device to establish a session
Get-attributes	POST[1]	Gets the status information of a program device
Read	READ[2]	Receives data from a specific program device
Read-from-invited-program-devices	READ[2]	Receives data from any invited program device[3]
Write	WRITE	Performs many of the ICF communications functions in a session
Write/Read	EXFMT	Performs the specified function and then receives data from the remote system
Release	REL	Releases the program device to end the session
Close	CLOSE	Closes the ICF file

1 The POST operation can retrieve either input/output (I/O) feedback information or the get-attributes. The information you get depends on the factors specified with the POST.

2 An RPG/400 read operation can be directed either to a specific program device or to all invited program devices. The support provided by the RPG/400 compiler determines whether to issue an ICF read or read-from-invited-program-devices operation based on the format of the read operation. For example, if a read is sent with a specific format or terminal specified, the read operation is interpreted as an ICF READ operation. Refer to the RPG/400 language manual for more information.

3 The read-from-invited-program-devices operation could complete without data if the timer interval established with either the timer function or wait record (WAITRCD) ends, or your job is ended (controlled).

Refer to the appropriate RPG/400 manual for details on the syntax and function of each operation.

RPG/400 Status Values

You must also understand the relationship between RPG *STATUS values and ICF major/minor return codes.

Table 11-2 shows the *STATUS values as returned in the RPG/400 INFDS for each major and minor return code. Use this list to determine the ICF return code or group of codes that corresponds to the *STATUS value.

*Table 11-2. *STATUS Values for Major and Minor Return Codes*

Major Code	Minor Code	*STATUS Value
00	All	00000
02	All	00000
03	All (except 09 and 10)	00000
03	09	01282
03	10	01331
04	All	01299
08	00	01285
11	00	00011
34	All	01201
80, 81	All	01251
82, 83	All	01255

Note: The mapping in Table 11-2 applies to major/minor codes set as a result of acquire, release, and general I/O operations. Certain major/minor codes are set for open and close errors as well as for other I/O errors. In cases where a major/minor code is set as a result of an open or close error, the return code will map to either the 01205, 01216, or 01217 *STATUS value, depending on which is applicable.

The return code field will not be updated for a *STATUS value of 01285, 01261, or 01281 when an I/O operation was attempted to an unacquired program device, because RPG/400 detects these conditions before calling ICF data management. This mapping is shown in order to note the appropriate RPG/400 *STATUS value to check for the given error condition.

RPG/400 Offset Values

In order for the RPG/400 support to access information from the I/O feedback area, you need to add the following RPG/400 offset values to the offset values listed in Appendix C.

Table 11-3. Offset Values for RPG/400

Open Feedback Area	Common I/O Feedback Area	File Dependent Feedback Area
81	241	367

Example Programs

The programs presented in this section are:

- Example I (Batch Data Transfer)

  Figure 11-1 shows a batch data transfer program that reads a database file and sends the data to a remote system. When the source program finishes sending its records, it sends an indication that it is done sending records to the target program. The target program then starts sending its records until it reaches the end-of-file. At end-of-file, the target program sends a detach indication to the source program. The two programs end their sessions.

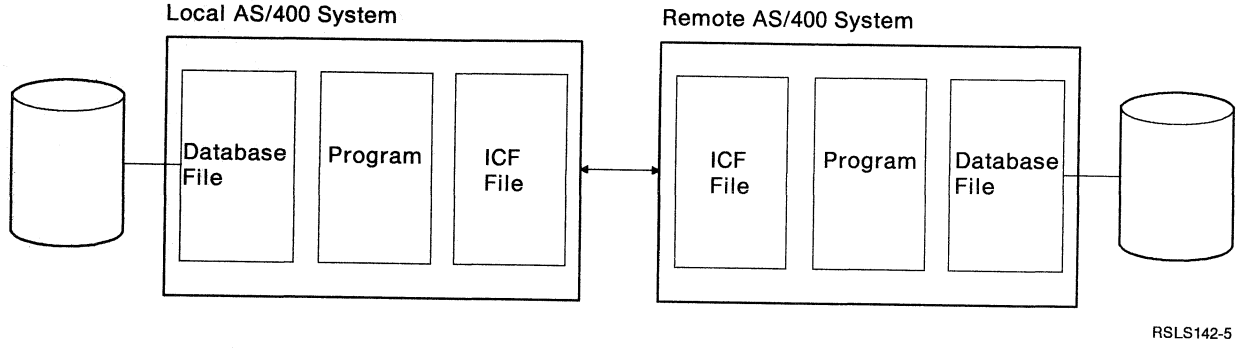


Figure 11-1. Batch Data Transfer

- Example II (Multiple-Session Inquiry)

  Figure 11-2 on page 11-5 shows an inquiry program that accepts inquiries from a display device, sends the request to one of four remote AS/400 systems, and waits for a response to the inquiry. Based on the input received from the display device, the program determines the target program to which it sends the inquiry request. The same program resides in each of the remote systems.

Figure 11-2 contains a display device and four ICF communications program devices.

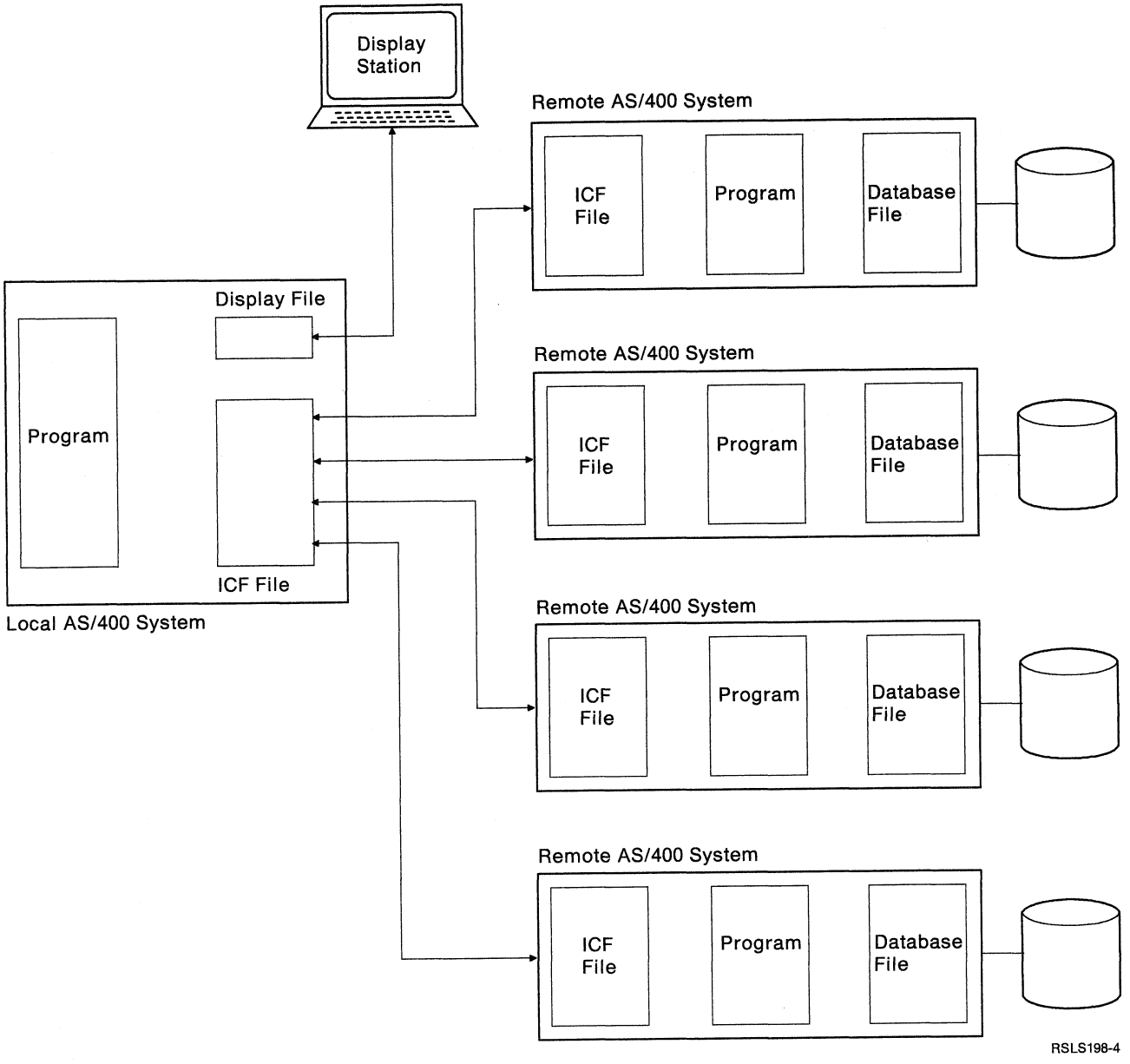


Figure 11-2. Multiple-Session Inquiry

The remainder of this chapter discusses the details of the two application examples. The DDS source for the ICF file, program listings, and an explanation of the programs are included.

Batch Data Transfer (Example I)

The following figures show a batch data transfer program. A source AS/400 system program communicates with a target program on another AS/400 program using the ICF support. The source program starts a target program on a remote AS/400 system, and transfers a file to that target program.

The target program responds, after receiving an indication that the source is done sending, by reading its own file and then sends the records to the source program until it reaches end-of-file. At end-of-file, the target program sends a detach request to the source program and ends its session.

Both the source program and the target program are described.

Transaction Flow of the Batch Data Transfer (Example I)

In Figure 11-3, the source program issues an evoke to start a program at the remote AS/400 system.

Note: An acquire operation is not necessary since the device was acquired during the open operation. The device was acquired during the open operation because the ACQPGMDEV parameter was used when the ICF file was created.

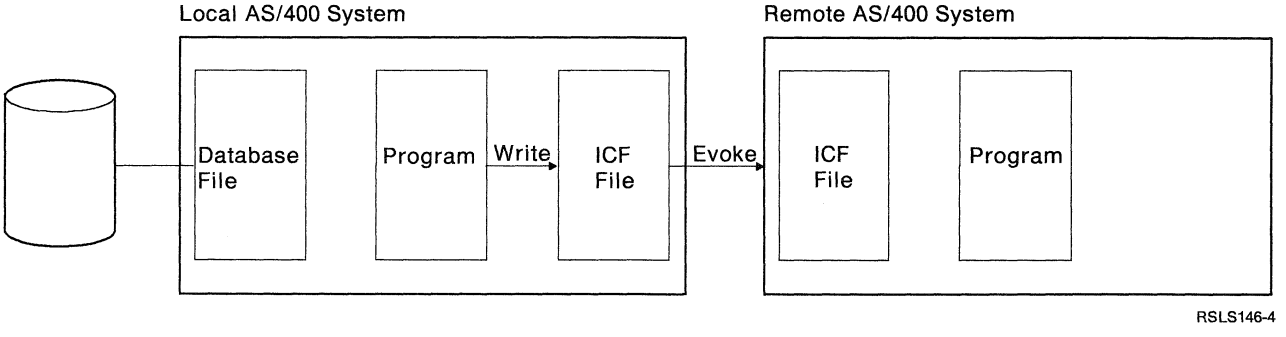


Figure 11-3. Evoke Request Starts a Target Program

After issuing the evoke request, the source program sends a database file to the target program, which prints the records as shown in Figure 11-4.

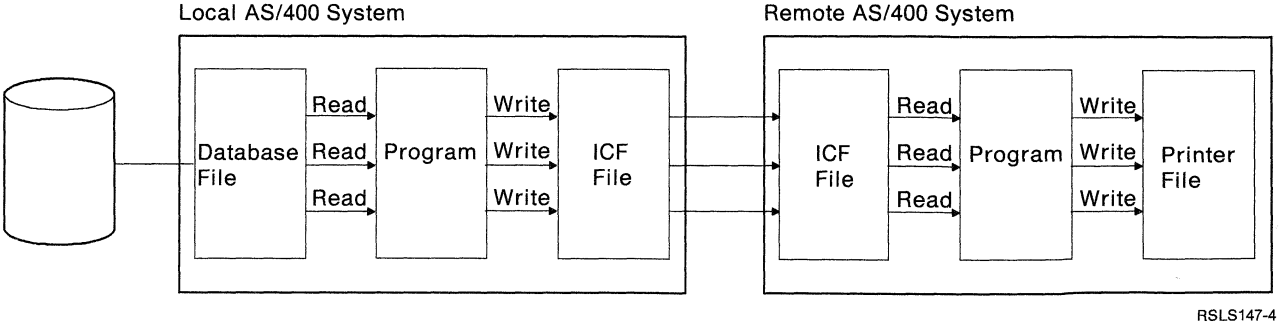


Figure 11-4. Target Program Prints Records

After the target program receives and prints the file, a database file is sent to the source program. The source program prints the records as they are received as shown in Figure 11-5.

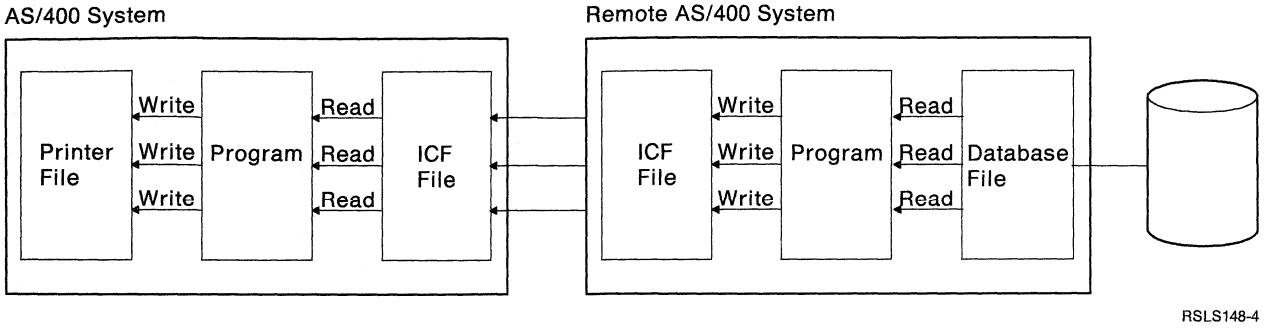


Figure 11-5. Source Program Prints the Received Records

Once all the records have been sent by the target program, the target program issues a detach to the source program to end the transaction, as shown in Figure 11-6.

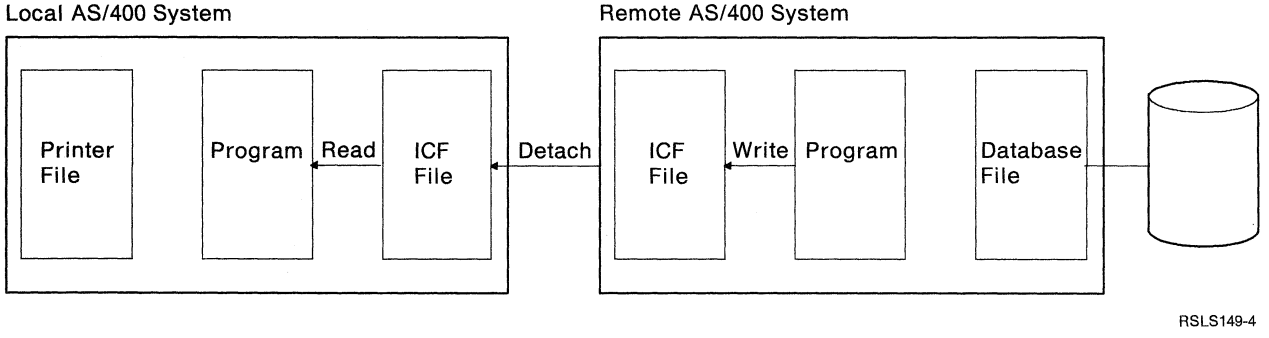


Figure 11-6. Target Program Ends the Transaction

Source Program Batch Transfer (Example I)

The following describes the RPG/400 batch data transfer source program.

Program Files: The RPG/400 batch data transfer source program uses the following files:

SRCICF	An ICF file used to send records to and receive records from the target program.
DBFILE	A database file that contains the records to be sent to the target program.
QPRINT	A printer file that is used to print the records received from the target program.

DDS Source: The DDS used in the ICF file is illustrated in Figure 11-7. The other files (DBFILE and QPRINT) are program-described and therefore do not require DDS.

```
A*************************************************************
A*                                                           *
A*                          ICF FILE                         *
A*              USED IN BATCH DATA TRANSFER PROGRAM          *
A*                                                           *
A*************************************************************
A*
A*  FILE LEVEL INDICATORS:
A*
A                                     INDARA
A*
A                                     RCVTRNRND(15 'END OF DATA')
A*
A  30                                 DETACH
A*
A                                     INDTXT(30 '30->DETACH TARG-
A                                     ET PROGRAM.')
A*
A                                     RCVDETACH(35 'RECEIVED    -
A                                     DETACHED.')
A*
A*
A*************************************************************
A*                       ICF RECORD FORMATS                  *
A*************************************************************
          R RCVDATA
            RCVFLD        80A
          R SNDDATA
            SNDFLD        80A
          R EVOKPGM
A  50                                 EVOKE(&LIB/&PGMID)
A  50                                 SECURITY(2 'PASSWRD' +
                                              3 'USERID')
A           PGMID         10A  P
A           LIB           10A  P
A         R ENDREC
A         R INVITE
A  45                                 INVITE
```

Figure 11-7. DDS Source for ICF File Used in Batch Data Transfer Program

ICF File Creation and Program Device Entry Definition: The command needed to create the ICF file is:

```
CRTICFF  FILE(ICFLIB/SRCICF)  SRCFILE(ICFLIB/QICFPUB)  SRCMBR(SRCICF)
   ACQPGMDEV(PGMDEVA)  TEXT('ICF FILE FOR BATCH DATA TRANSFER')
```

The command needed to define the program device entry is:

```
ADDICFDEVE FILE(ICFLIB/SRCICF) PGMDEV(PGMDEVA) RMTLOCNAME(CHICAGO)
```

Program Explanation: The following describes the structure of the program examples illustrated in Figure 11-8 on page 11-12 and Figure 11-9 on page 11-17. The ICF file used in the first example is defined by the user, and uses externally described data formats (DDS). The second example uses the same file, but uses program-described data and system-supplied formats. The reference numbers in the explanation below correspond to those in the following program examples.

Although the basic structure of the two examples provided is the same, there are differences because of the use of user-defined formats and system-supplied formats. All output operations to the ICF file in the first example are done using the WRITE statement. All output operations to the ICF file in the second example using system-supplied formats are done using the EXCPT statement.

Differences between the first and second example are described in notes in each of the descriptions.

1 The files used in this program are defined in the file specifications section. SRCICF is the ICF file used to send records to the target program.

The files used in the program are opened at the beginning of the RPG/400 cycle and the ICF program device is implicitly acquired because the ACQPGMDEV parameter was specified on the create ICF file (CRTICFF) command.

Note: The input records for SRCICF are explicitly coded in the program using system-supplied formats, because SRCICF is now treated as a program-described file. The system-supplied file QICDMF can be used instead of SRCICF. You can use the system-supplied file by specifying QICDMF in the file specification, or by using an OVRICFF command to change the file name from SRCICF to QICDMF. The OVRICFF command can also be used to change the ACQPGMDEV parameter of the file.

2 FEEDBK is the name of the file information data structure (INFDS) used with SRCICF. It contains the following information:

- Record format name (FMTNM)
- Program device name (PGMDEV)
- Major/minor return code (MAJMIN, MAJCOD, MINCOD)

3 This section builds and sends the evoke request to the remote system. Because the DDS for the record format only specifies the field identifiers with the record, the program moves the literal value RTDBATCL to the field *PGMID*, and ICFLIB to the field *LIB*. Indicator 50 is set to indicate that the program start request is to be sent.

When the program start request is received at the remote system, ICFLIB is searched for RTDBATCL and that program is then started. RTDBATCL is a control language (CL) program that contains the following:

```
ADDLIBLE ICFLIB
CALL ICFLIB/RTDBAT
```

Note: In the program using system-supplied formats, the library and program (ICFLIB/RTFBATCL) are specified as part of the $$EVOKNI format. RTFBATCL is a CL program that contains the following:

```
ADDLIBLE ICFLIB
CALL ICFLIB/RTFBAT
```

4 This section reads a record from the database file. If the record read is the end-of-file, the program sets indicator 98 on and then goes to **11**.

If it is not the last record, the data is moved to field *SNDFLD* and the program goes to **10** to write the record to the ICF program device. When control returns from **10**, the next database record is read.

5 Data is read from the program device associated with the ICF file (SRCICF).

6 If an error occurs on the read (return code greater than 03), the error is handled. Otherwise, if data is received (return code not = 03), the data is written to the printer file (QPRINT).

This section reads data records until the detach indication is received from the target program. When the detach is received, indicator 35 is set on, as defined by the RCVDETACH keyword in the DDS for the ICF file. Note that RCVDETACH is a file level-keyword.

Note: In the program using system-supplied formats, the minor return code of '08' is checked to verify whether the detach is received.

7 After the detach request has been received, the following message is written to the print file:

```
RSDBAT HAS COMPLETED NORMALLY
```

The session is ended in **9**.

Note: The program name is RSFBAT in the program using system-supplied formats.

8 When an I/O operation to the ICF file (SRCICF) completes unsuccessfully, the following message is written to the print file:

```
RSDBAT HAS COMPLETED ABNORMALLY
```

The session is ended in **9**.

Note: The program name is RSFBAT in the program using system-supplied formats.

9 The program ends the job by setting on last run (LR) indicator and returning to caller of the program. The ICF file is closed, and the session ends at the end of the RPG/400 cycle.

10 This subroutine is called to write data to the program device associated with the ICF file using the format SNDDATA. If an error occurs, the program goes to **8** and a message is printed.

Note: The $$SENDNI format is used instead of the user-defined SNDDATA format in the program using system-supplied formats.

11 This subroutine is called to perform an invite request to the ICF program device using format INVITE. If an error occurs, the program goes to 8 and a message is printed.

Note: The $$SEND format is used instead of the user-defined INVITE format in the program using system-supplied formats.

```
5728RG1 R01M02  881028          IBM AS/400 RPG/400              ICFLIB/RSDBAT       03/20/89  15:46:30      Page       1
Compiler . . . . . :   IBM AS/400 RPG/400
Command Options:
  Program . . . . . :   ICFLIB/RSDBAT
  Source file . . . . . :   ICFLIB/QICFPUB
  Source member . . . . . :   *PGM
  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
  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 . . . . . :   RSDBAT
  File . . . . . :   QICFPUB
  Library . . . . . :   ICFLIB
  Last Change . . . . . :   03/20/89  15:26:21
  Description . . . . . :   rpg batch file transfer using dds source
5728RG1 R01M02  881028          IBM AS/400 RPG/400              ICFLIB/RSDBAT       03/20/89  15:46:30      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  H*****************************************************************                  12/15/87
     200  H*                                                                                  03/20/89
     300  H*  THIS IS A BATCH FILE TRANSFER PROGRAM THAT READS A SEQUENTIAL                    10/14/87
     400  H*  FILE AND SENDS THE RECORDS TO THE REMOTE SYSTEM UNTIL THE END                    10/14/87
     500  H*  OF FILE IS REACHED. AT THIS TIME, THE PROGRAM STOPS SENDING                    10/14/87
     600  H*  AND STARTS RECEIVING RECORDS FROM THE REMOTE SYSTEM UNTIL A                    10/14/87
     700  H*  A DETACH INDICATION IS RECEIVED.                                               10/14/87
     800  H*                                                                                  03/20/89
     900  H*****************************************************************                  10/14/87
    1000   * 1                                                                               10/14/87
           H                                                                                            *****
    1100  FSRCICF  CF  E                    WORKSTN                                         10/14/87
    1200  F                                                KINFDS FEEDBK                    10/14/87
           RECORD FORMAT(S):  LIBRARY ICFLIB FILE SRCICF.
                    EXTERNAL FORMAT RCVDATA RPG NAME RCVDATA
                    EXTERNAL FORMAT SNDDATA RPG NAME SNDDATA
                    EXTERNAL FORMAT EVOKPGM RPG NAME EVOKPGM
                    EXTERNAL FORMAT ENDREC RPG NAME ENDREC
                    EXTERNAL FORMAT INVITE RPG NAME INVITE
    1300  FDBFILE  IF  F      80            DISK                                            10/14/87
    1400  FQPRINT  O   F     132            PRINTER                                         10/14/87
    1500  IDBFILE  NS  80                                                                   10/14/87
    1600  I                                        1  80 DBDATA                            10/14/87
    1700  I*                                                                                10/14/87
A000000   INPUT  FIELDS FOR RECORD RCVDATA FILE SRCICF FORMAT RCVDATA.
A000001                                          1  80 RCVFLD
B000000   INPUT  FIELDS FOR RECORD SNDDATA FILE SRCICF FORMAT SNDDATA.
B000001                                          1  80 SNDFLD
C000000   INPUT  FIELDS FOR RECORD EVOKPGM FILE SRCICF FORMAT EVOKPGM.
D000000   INPUT  FIELDS FOR RECORD ENDREC FILE SRCICF FORMAT ENDREC.
E000000   INPUT  FIELDS FOR RECORD INVITE FILE SRCICF FORMAT INVITE.
    1800  IFEEDBK 2      DS                                                                    10/14/87
    1900  I                                       38  47 FMTNM                             10/14/87
    2000  I                                      273 282 PGMDEV                            10/14/87
    2100  I                                      401 404 MAJMIN                            10/14/87
    2200  I                                      401 402 MAJCOD                            10/14/87
```

Figure 11-8 (Part 1 of 5). Source Program Example — RSDBAT (User-Defined Formats)

```
    2300  I                                        403 404 MINCOD                                  10/14/87
    2400  C*****************************************************************                      10/14/87
    2500  C*                                                                                      03/20/89
    2600  C*   EVOKE PROGRAM 'RTDBATCL' ON REMOTE SYSTEM IN LIBRARY ICFLIB.                       10/14/87
    2700  C*   INDICATOR 50 (*IN50) IS ASSOCIATED WITH THE EVOKE KEYWORD.                         10/14/87
    2800  C*                                                                                      03/20/89
    2900  C*****************************************************************                      10/14/87
    3000  C* 3                                                                                   10/14/87
5728RG1 R01M02  881028            IBM AS/400 RPG/400                    ICFLIB/RSDBAT        03/20/89  15:46:30      Page        3
SEQUENCE                                                                          IND    DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE     LINE   ID
    3100  C           ITMIN     TAG                                                             10/14/87
    3200  C                     MOVEL'RTDBATCL'PGMID                                            10/14/87
    3300  C                     MOVEL'ICFLIB  'LIB                                              10/14/87
    3400  C                     MOVE '1'       *IN50                                            10/14/87
    3500  C                     WRITEEVOKPGM                        ISSUE EVOKE                 10/14/87
    3600  C                     MOVE '0'       *IN50                                            10/14/87
    3700  C           MAJCOD    CABGT'03'      NOTOKR               ERROR?                      10/14/87
    3800  C*****************************************************************                      10/14/87
    3900  C*                                                                                      03/20/89
    4000  C*   AFTER SUCCESSFUL PROCESSING OF THE EVOKE OPERATION, A RECORD                       03/20/89
    4100  C*   IS READ FROM THE DATABASE FILE AND SENT TO THE REMOTE SYSTEM.                      03/20/89
    4200  C*   THIS IS REPEATED UNTIL END OF FILE IS REACHED ON THE DATABASE                      03/20/89
    4300  C*   FILE. AT END OF FILE, THE PROGRAM DEVICE IS INVITED AND                           03/20/89
    4400  C*   CONTROL GOES TO RECDTA TO GET DATA FROM THE REMOTE SYSTEM.                         10/16/87
    4500  C*                                                                                      03/20/89
    4600  C*****************************************************************                      10/14/87
    4700  C* 4                                                                                   10/14/87
    4800  C           SENDTA    TAG                                                             10/14/87
    4900  C                     READ DBFILE                   98                  3             10/14/87
    5000  C   98                EXSR INVSND                     INVITE                        10/14/87
    5100  C           EOFPSW    IFNE '1'                                                  B001  10/14/87
    5200  C                     MOVE DBDATA    SNDFLD                                      001  10/14/87
    5300  C                     EXSR WCFRTN                                                001  10/14/87
    5400  C                     GOTO SENDTA                     SEND DATA                  001  10/14/87
    5500  C                     END                                                       E001  10/14/87
    5600  C*****************************************************************                      10/14/87
    5700  C*                                                                                      03/20/89
    5800  C*   RECEIVE RECORDS FROM THE REMOTE SYSTEM UNTIL THE RCVDETACH                         03/20/89
    5900  C*   INDICATOR IS SET ON.  EACH RECORD RECEIVED IS PRINTED TO                           03/20/89
    6000  C*   THE PRINT FILE.                                                                   03/20/89
    6100  C*                                                                                      03/20/89
    6200  C*****************************************************************                      10/14/87
    6300  C* 5                                                                                   10/14/87
    6400  C           RECDTA    TAG                                                             10/14/87
    6500  C                     READ SRCICF                   98                  3             10/14/87
    6600  C*****************************************************************                      10/14/87
    6700  C*                                                                                      03/20/89
    6800  C*   IF ANY ICF FILE ERROR OCCURS, PRINT A LINE CONTAINING                              03/20/89
    6900  C*   INFORMATION ABOUT THE ERROR.                                                      10/14/87
    7000  C*                                                                                      03/20/89
    7100  C*****************************************************************                      10/14/87
    7200  C* 6                                                                                   10/14/87
    7300  C           MAJCOD    CABGT'03'      NOTOKR                                          10/14/87
    7400  C           MAJCOD    CABEQ'03'      CHKDET               NO DATA?                   10/14/87
    7500  C                     EXCPTPTREC                                                      10/14/87
    7600  C           CHKDET    TAG                                                             10/14/87
    7700  C           *IN35     CABNE'1'       RECDTA               DETACH?                    10/14/87
    7800  C*****************************************************************                      10/14/87
    7900  C*                                                                                      03/20/89
    8000  C*   AFTER A DETACH INDICATION IS RECEIVED, AN EOJ MESSAGE                              03/20/89
    8100  C*   IS PRINTED AND THE SESSION IS ENDED.                                              03/20/89
    8200  C*                                                                                      03/20/89
```

Figure 11-8 (Part 2 of 5). Source Program Example — RSDBAT (User-Defined Formats)

```
   8300  C*************************************************************                   10/14/87
   8400  C* 7                                                                              10/14/87
5728RG1 R01M02  881028              IBM AS/400 RPG/400                  ICFLIB/RSDBAT     03/20/89  15:46:30       Page      4
SEQUENCE                                                                          IND    DO    LAST       PAGE   PROGRAM
NUMBER   *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE     LINE   ID
   8500  C                        EXCPTOKEND                                                   10/14/87
   8600  C                        GOTO END                                                     10/14/87
   8700  C*************************************************************                   10/14/87
   8800  C*                                                                                03/20/89
   8900  C*   WHEN AN ERROR OCCURS ON AN ICF SESSION, INFORMATION                         03/20/89
   9000  C*   ABOUT THE ERROR IS PRINTED.                                                 03/20/89
   9100  C*                                                                                03/20/89
   9200  C*************************************************************                   10/14/87
   9300  C* 8                                                                              10/14/87
   9400  C           NOTOKR       TAG                                                      10/14/87
   9500  C                        EXCPTNOTOK                                               10/14/87
   9600  C*****************************************************************               10/14/87
   9700  C*                                                                                03/20/89
   9800  C*    WHEN PROCESSING IS FINISHED, THE LAST RECORD SWITCH                        10/14/87
   9900  C*    IS TURNED ON AND THE PROGRAM IS ENDED.                                     10/14/87
  10000  C*                                                                                03/20/89
  10100  C*****************************************************************               10/14/87
  10200  C* 9                                                                              10/14/87
  10300  C           END          TAG                                                      10/14/87
  10400  C                        SETON                    LR                1             10/14/87
  10500  C                        RETRN                                                    10/14/87
  10600  C*****************************************************************               10/14/87
  10700  C*                                                                                03/20/89
  10800  C*   THIS SUBROUTINE SENDS DATA TO THE REMOTE SYSTEM                             10/14/87
  10900  C*                                                                                03/20/89
  11000  C*****************************************************************               10/14/87
  11100  C* 10                                                                             10/14/87
  11200  C           WCFRTN       BEGSR                                                    10/14/87
  11300  C                        WRITESNDDATA                                             10/14/87
  11400  C           MAJCOD       CABGT'03'      NOTOKR                                    10/14/87
  11500  C                        ENDSR                                                    10/14/87
  11600  C*****************************************************************               10/14/87
  11700  C*                                                                                03/20/89
  11800  C* THIS SUBROUTINE IS CALLED AT END OF FILE TO REQUEST THE REMOTE                 10/14/87
  11900  C* PROGRAM TO START SENDING DATA. AN INVITE OPERATION IS ISSUED                   10/14/87
  12000  C* TO NOTIFY THE TARGET PROGRAM THAT IT CAN START SENDING DATA.                   10/14/87
  12100  C*                                                                                03/20/89
  12200  C*****************************************************************               10/14/87
  12300  C* 11                                                                             10/14/87
  12400  C           INVSND       BEGSR                                                    10/14/87
  12500  C                        MOVE '1'       EOFPSW  1                                 10/14/87
  12600  C                        MOVE '1'       *IN45                                     10/14/87
  12700  C                        WRITEINVITE                                              10/14/87
  12800  C           MAJCOD       CABGT'03'      NOTOKR                                    10/14/87
  12900  C                        ENDSR                                                    10/14/87
  13000  C*************************************************************                   10/14/87
  13100  OQPRINT  E 1                PTREC                                                 10/14/87
  13200  O                           RCVFLD    80                                          10/14/87
  13300  O        E 1                OKEND                                                 10/14/87
  13400  O                                     21 'RSDBAT HAS COMPLETED '                  10/14/87
  13500  O                                     30 'NORMALLY.'                              10/14/87
  13600  O        E 1                NOTOK                                                 10/14/87
5728RG1 R01M02  881028              IBM AS/400 RPG/400                  ICFLIB/RSDBAT     03/20/89  15:46:30       Page      5
SEQUENCE                                                                          IND    DO    LAST       PAGE   PROGRAM
NUMBER   *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE     LINE   ID
  13700  O                                     21 'RSDBAT HAS COMPLETED '                  10/14/87
  13800  O                                     32 'ABNORMALLY.'                            10/14/87
  13900  O                           MAJCOD    35                                          10/14/87
  14000  O                                     36 '/'                                      10/14/87
  14100  O                           MINCOD    39                                          10/14/87
```

Figure 11-8 (Part 3 of 5). Source Program Example — RSDBAT (User-Defined Formats)

```
   14200  O                                   46 'FORMAT:'                                    10/14/87
   14300  O                        FMTNM      56                                              10/14/87
   14400  O                                   63 'DEVICE:'                                    10/14/87
   14500  O                        PGMDEV     80                                              03/20/89
* 6103     14501   OVERFLOW INDICATOR OA ASSIGNED TO FILE QPRINT.
  F000000   OUTPUT FIELDS FOR RECORD SNDDATA FILE SRCICF FORMAT SNDDATA.
  F000001                           SNDFLD     80  CHAR   80
  G000000   OUTPUT FIELDS FOR RECORD EVOKPGM FILE SRCICF FORMAT EVOKPGM.
  G000001                           PGMID      10  CHAR   10
  G000002                           LIB        20  CHAR   10
  H000000   OUTPUT FIELDS FOR RECORD INVITE FILE SRCICF FORMAT INVITE.
          * * * * *  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
* 7089      1100   RPG PROVIDES SEPARATE INDICATOR AREA FOR FILE SRCICF.
* 7086      1300   RPG PROVIDES BLOCK OR UNBLOCK SUPPORT FOR FILE DBFILE.
 5728RG1 R01M02  881028            IBM AS/400 RPG/400                ICFLIB/RSDBAT        03/20/89  15:46:30      Page       6
                       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  DBFILE      DISK          1300D    1500     4900
    03  QPRINT      PRINTER       1400D   13100    13300    13600    14501
    01  SRCICF      WORKSTN       1100D    6500
          ENDREC                  1100D D000000
          EVOKPGM                 1100D C000000     3500  G000000
          INVITE                  1100D E000000    12700  H000000
          RCVDATA                 1100D A000000
          SNDDATA                 1100D B000000    11300  F000000
 Field References:
        FIELD       ATTR    REFERENCES (M=MODIFIED D=DEFINED)
        *IN35       A(1)       7700
        *IN45       A(1)      12600M
        *IN50       A(1)       3400M    3600M
        CHKDET      TAG        7400     7600D
        DBDATA      A(80)      1600D    5200
        END         TAG        8600    10300D
        EOFPSW      A(1)       5100    12500D
        FEEDBK      DS(404)    1100     1800D
        FMTNM       A(8)       1900D   14300
        INVSND      BEGSR      5000    12400D
* 7031  ITMIN       TAG        3100D
        LIB         A(10)      3300M G000002D
        MAJCOD      A(2)       2200D    3700     7300     7400    11400
                               12800    13900
* 7031  MAJMIN      A(4)       2100D
        MINCOD      A(2)       2300D   14100
        NOTOK       EXCPT      9500    13600
        NOTOKR      TAG        3700     7300     9400D   11400    12800
        OKEND       EXCPT      8500    13300
        PGMDEV      A(10)      2000D   14500
        PGMID       A(10)      3200M G000001D
        PTREC       EXCPT      7500    13100
        RCVFLD      A(80)    A000001D  13200
        RECDTA      TAG        6400D    7700
        SENDTA      TAG        4800D    5400
        SNDFLD      A(80)    B000001D   5200M F000001D
        WCFRTN      BEGSR      5300    11200D
        'ICFLIB  '  LITERAL    3300
        'RTDBATCL'  LITERAL    3200
        '0'         LITERAL    3600
        '03'        LITERAL    3700     7300     7400    11400    12800
        '1'         LITERAL    3400     5100     7700    12500    12600
 5728RG1 R01M02  881028            IBM AS/400 RPG/400                ICFLIB/RSDBAT        03/20/89  15:46:30      Page       7
 Indicator References:
        INDICATOR  REFERENCES (M=MODIFIED D=DEFINED)
        *IN          3400M    3600M    7700    12600M
        LR          10400M
        OA           1400D   14501
 * 7031  15
```

Figure 11-8 (Part 4 of 5). Source Program Example — RSDBAT (User-Defined Formats)

```
* 7031  30
       35           7700
* 7031  45          12600M
* 7031  50           3400M    3600M
* 7031  80           1500M
       98           4900M    5000     6500M
     * * * * *  E N D  O F  C R O S S  R E F E R E N C E  * * * * *
 5728RG1 R01M02  881028           IBM AS/400 RPG/400                ICFLIB/RSDBAT        03/20/89  15:46:30      Page        8
                        M e s s a g e  S u m m a r y
* QRG6103 Severity:  00   Number:    1
          Message . . . . :   No Overflow Indicator is specified but an
            indicator is assigned to a file and automatic skip to 6 is
            generated.
* QRG7031 Severity:  00   Number:    7
          Message . . . . :   The Name or indicator is not referenced.
* QRG7086 Severity:  00   Number:    1
          Message . . . . :   The RPG handles blocking function for file.
            INFDS contents updated only when blocks of data transferred.
* QRG7089 Severity:  00   Number:    1
          Message . . . . :   The RPG provides Separate-Indicator area for
            file.
     * * * * *  E N D  O F  M E S S A G E  S U M M A R Y  * * * * *
 5728RG1 R01M02  881028           IBM AS/400 RPG/400                ICFLIB/RSDBAT        03/20/89  15:46:30      Page        9
                         F i n a l  S u m m a r y
 Message Count:  (by Severity Number)
           TOTAL    00     10     20     30     40     50
            10      10      0      0      0      0      0
 Program Source Totals:
    Records . . . . . . . . . . :   145
    Specifications  . . . . . . :   66
    Table Records . . . . . . . :   0
    Comments  . . . . . . . . . :   79
 PRM has been called.
 Program RSDBAT is placed in library ICFLIB. 00 highest Error-Severity-Code.
         * * * * *  E N D  O F  C O M P I L A T I O N  * * * * *
```

Figure 11-8 (Part 5 of 5). Source Program Example — RSDBAT (User-Defined Formats)

```
5728RG1 R01M02  881028           IBM AS/400 RPG/400                ICFLIB/RSFBAT       03/20/89  15:47:16      Page      1
Compiler . . . . . :   IBM AS/400 RPG/400
Command Options:
  Program . . . . . :   ICFLIB/RSFBAT
  Source file . . . . . :   ICFLIB/QICFPUB
  Source member . . . . . :   *PGM
  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
  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 . . . . . :   RSFBAT
  File . . . . . :   QICFPUB
  Library  . . . . . :   ICFLIB
  Last Change  . . . . . :   03/20/89  15:30:19
  Description  . . . . . :   rpg batch file transfer using $$FORMAT
5728RG1 R01M02  881028           IBM AS/400 RPG/400                ICFLIB/RSFBAT       03/20/89  15:47:16      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  H*****************************************************************          10/16/87
     200  H*                                                                          03/20/89
     300  H*  THIS IS A BATCH FILE TRANSFER PROGRAM THAT READS A SEQUENTIAL           10/16/87
     400  H*  FILE AND SENDS THE RECORDS TO THE REMOTE SYSTEM UNTIL THE END           10/16/87
     500  H*  OF FILE IS REACHED. AT THIS TIME, THE PROGRAM STOPS SENDING            10/16/87
     600  H*  AND STARTS RECEIVING RECORDS FROM THE REMOTE SYSTEM UNTIL A            10/16/87
     700  H*  A DETACH INDICATION IS RECEIVED.                                       10/16/87
     800  H*                                                                          03/20/89
     900  H*****************************************************************          10/16/87
    1000  H* 1                                                                       10/16/87
    1100  FSRCICF  CF  F      84            WORKSTN                                   10/16/87
    1200  F                                               KINFDS FEEDBK             10/16/87
    1300  FDBFILE  IF  F      80            DISK                                      10/16/87
    1400  FQPRINT  O   F     132            PRINTER                                   10/16/87
    1500  ISRCICF  NS  82                                                            10/16/87
    1600  I                                       1  80 RCVFLD                       10/16/87
    1700  IDBFILE  NS  80                                                            10/16/87
    1800  I                                       1  80 DBDATA                       10/16/87
    1900  I* 2                                                                       10/16/87
    2000  IFEEDBK      DS                                                            10/16/87
    2100  I                                      38  47 FMTNM                        10/16/87
    2200  I                                     273 282 PGMDEV                       10/16/87
    2300  I                                     401 404 MAJMIN                       10/16/87
    2400  I                                     401 402 MAJCOD                       10/16/87
    2500  I                                     403 404 MINCOD                       10/16/87
    2600  C*****************************************************************          10/16/87
    2700  C*                                                                          03/20/89
    2800  C*   EVOKE PROGRAM 'RTFBATCL' ON REMOTE SYSTEM IN LIBRARY ICFLIB.           10/16/87
    2900  C*   THE USER ID AND PASSWORD ARE DEFINED AS PART OF THE $$EVOKNI           10/16/87
    3000  C*   FORMAT.                                                               03/20/89
    3100  C*                                                                         03/20/89
    3200  C*****************************************************************          10/16/87
    3300  C* 3                                                                       10/16/87
    3400  C           ITMIN     TAG                                                 10/16/87
    3500  C                     EXCPTEVOKE                    ISSUE EVOKE           10/16/87
    3600  C           MAJCOD    CABGT'03'     NOTOKR          ERROR?                10/16/87
```

Figure 11-9 (Part 1 of 5). Source Program Example — RSFBAT (System-Supplied Formats)

```
    3700  C*****************************************************************                10/16/87
    3800  C*                                                                               03/20/89
    3900  C*   AFTER THE SUCCESSFUL PROCESSING OF THE EVOKE OPERATION, A                    03/20/89
    4000  C*   RECORD IS READ FROM THE DATABASE FILE, AND SENT TO THE REMOTE                03/20/89
    4100  C*   SYSTEM. THIS IS REPEATED UNTIL AN END OF FILE IS REACHED ON                  03/20/89
    4200  C*   THE DATABASE FILE. AT THIS TIME, THE PROGRAM DEVICE IS INVI-                 03/20/89
    4300  C*   TED, AND CONTROL GOES TO RECDTA TO GET DATA FROM THE REMOTE                  03/20/89
    4400  C*   SYSTEM.                                                                     03/20/89
5728RG1 R01M02  881028            IBM AS/400 RPG/400                 ICFLIB/RSFBAT       03/20/89  15:47:16      Page        3
SEQUENCE                                                                    IND    DO     LAST       PAGE    PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM    UPDATE     LINE    ID
    4500  C*                                                                              03/20/89
    4600  C*****************************************************************             10/16/87
    4700  C* 4                                                                           10/16/87
    4800  C           SENDTA    TAG                                                     10/16/87
    4900  C                     READ DBFILE                 98                  3       10/16/87
    5000  C   98                EXSR INVSND                   INVITE                    10/16/87
    5100  C           EOFPSW    IFNE '1'                                          B001  10/16/87
    5200  C                     EXSR WCFRTN                                        001  10/16/87
    5300  C                     GOTO SENDTA                   SEND DATA            001  10/16/87
    5400  C                     END                                                E001  10/16/87
    5500  C*****************************************************************             10/16/87
    5600  C*                                                                              03/20/89
    5700  C*   THE PROGRAM STARTS RECEIVING RECORDS AT THIS POINT FROM THE                 03/20/89
    5800  C*   REMOTE SYSTEM UNTIL A DETACH INDICATION IS RECEIVED. EACH                   03/20/89
    5900  C*   RECORD RECEIVED IS PRINTED TO THE PRINT FILE.                               03/20/89
    6000  C*                                                                              03/20/89
    6100  C*****************************************************************             10/16/87
    6200  C* 5                                                                           10/16/87
    6300  C           RECDTA    TAG                                                     10/16/87
    6400  C                     READ SRCICF                 98                  3       10/16/87
    6500  C*****************************************************************             10/16/87
    6600  C*                                                                              03/20/89
    6700  C*   IF AN ICF FILE ERROR OCCURS, PRINT A LINE CONTAINING                       03/20/89
    6800  C*   INFORMATION ABOUT THE ERROR.                                               10/16/87
    6900  C*                                                                              03/20/89
    7000  C*****************************************************************             10/16/87
    7100  C* 6                                                                           10/16/87
    7200  C           MAJCOD    CABGT'03'     NOTOKR                                    10/16/87
    7300  C           MAJCOD    CABEQ'03'     CHKDET          NO DATA?                  10/16/87
    7400  C                     EXCPTPTREC                                              10/16/87
    7500  C           CHKDET    TAG                                                     10/16/87
    7600  C           MINCOD    CABNE'08'     RECDTA          DETACH?                   10/16/87
    7700  C*************************************************************               10/16/87
    7800  C*                                                                            03/20/89
    7900  C*   AFTER A DETACH INDICATION IS RECEIVED, AN EOJ MESSAGE IS                 03/20/89
    8000  C*   PRINTED AND THE SESSION IS ENDED.                                        03/20/89
    8100  C*                                                                            03/20/89
    8200  C*************************************************************               10/16/87
    8300  C* 7                                                                         10/16/87
    8400  C                     EXCPTOKEND                                            10/16/87
    8500  C                     GOTO END                                              10/16/87
    8600  C*************************************************************               10/16/87
    8700  C*                                                                            03/20/89
    8800  C*   WHEN AN ERROR OCCURS ON AN ICF SESSION, INFORMATION                      03/20/89
    8900  C*   ABOUT THE ERROR IS PRINTED.                                              03/20/89
    9000  C*                                                                            03/20/89
    9100  C*************************************************************               10/16/87
    9200  C* 8                                                                         10/16/87
    9300  C           NOTOKR    TAG                                                   10/16/87
    9400  C                     EXCPTNOTOK                                            10/16/87
    9500  C*****************************************************************           10/16/87
    9600  C*                                                                          03/20/89
    9700  C*    WHEN PROCESSING IS FINISHED, THE LAST RECORD SWITCH IS                 03/20/89
    9800  C*    TURNED ON AND THE PROGRAM IS ENDED.                                    03/20/89
```

Figure 11-9 (Part 2 of 5). Source Program Example — RSFBAT (System-Supplied Formats)

```
5728RG1 R01M02  881028            IBM AS/400 RPG/400                 ICFLIB/RSFBAT        03/20/89  15:47:16      Page       4
SEQUENCE                                                                                    IND    DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE     LINE   ID
    9900  C*                                                                                      03/20/89
   10000  C*****************************************************************                      10/16/87
   10100  C* 9                                                                                    10/16/87
   10200  C           END       TAG                                                               10/16/87
   10300  C                     SETON                     LR                    1                 10/16/87
   10400  C                     RETRN                                                             10/16/87
   10500  C*****************************************************************                      10/16/87
   10600  C*                                                                                      03/20/89
   10700  C*   THIS SUBROUTINE SENDS DATA TO THE REMOTE SYSTEM                                    10/16/87
   10800  C*                                                                                      03/20/89
   10900  C*****************************************************************                      10/16/87
   11000  C* 10                                                                                   10/16/87
   11100  C           WCFRTN    BEGSR                                                           10/16/87
   11200  C                     EXCPTSNDATA                                                       10/16/87
   11300  C           MAJCOD    CABGT'03'      NOTOKR                                           10/16/87
   11400  C                     ENDSR                                                             10/16/87
   11500  C*****************************************************************                      10/16/87
   11600  C*                                                                                      03/20/89
   11700  C* THIS SUBROUTINE IS CALLED AT END OF FILE TO REQUEST THE REMOTE                       10/16/87
   11800  C* PROGRAM TO START SENDING DATA. AN INVITE OPERATION IS ISSUED                         10/16/87
   11900  C* TO NOTIFY THE TARGET PROGRAM THAT IT CAN START SENDING DATA.                        10/16/87
   12000  C*                                                                                      03/20/89
   12100  C*****************************************************************                      10/16/87
   12200  C* 11                                                                                   10/16/87
   12300  C           INVSND    BEGSR                                                           10/16/87
   12400  C                     MOVE '1'       EOFPSW  1                                        10/16/87
   12500  C                     EXCPTINVITE                                                       10/16/87
   12600  C           MAJCOD    CABGT'03'      NOTOKR                                           10/16/87
   12700  C                     ENDSR                                                             10/16/87
   12800  C*************************************************************                        10/16/87
   12900  OQPRINT  E 1              PTREC                                                       10/16/87
   13000  O                         RCVFLD    80                                                10/16/87
   13100  O        E 1              OKEND                                                       10/16/87
   13200  O                                   21 'RSFBAT HAS COMPLETED '                        10/16/87
   13300  O                                   30 'NORMALLY.'                                    10/16/87
   13400  O        E 1              NOTOK                                                       10/16/87
   13500  O                                   21 'RSFBAT HAS COMPLETED '                        10/16/87
   13600  O                                   32 'ABNORMALLY.'                                  10/16/87
   13700  O                         MAJCOD    35                                                10/16/87
   13800  O                                   36 '/'                                            10/16/87
   13900  O                         MINCOD    39                                                10/16/87
   14000  O                                   46 'FORMAT:'                                      10/16/87
   14100  O                         FMTNM     56                                                10/16/87
   14200  O                                   63 'DEVICE:'                                      10/16/87
   14300  O                         PGMDEV    80                                                03/20/89
   14400  OSRCICF  E                EVOKE                                                       10/16/87
   14500  O                                   K8 '$$EVOKNI'                                     10/16/87
   14600  O                                    8 'RTFBATCL'                                     10/16/87
   14700  O                                   16 'QSECOFR '                                     11/16/88
   14800  O                                   24 'QSECOFR '                                     11/16/88
   14900  O                                   32 'ICFLIB  '                                     10/16/87
   15000  O        E                SNDATA                                                      10/16/87
5728RG1 R01M02  881028            IBM AS/400 RPG/400                 ICFLIB/RSFBAT        03/20/89  15:47:16      Page       5
SEQUENCE                                                                                    IND    DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE     LINE   ID
   15100  O                                   K8 '$$SENDNI'                                     10/16/87
   15200  O                                    4 '0080'                                         10/16/87
   15300  O                         DBDATA    84                                                10/16/87
   15400  O        E                INVITE                                                      10/16/87
   15500  O                                   K6 '$$SEND'                                       10/16/87
   15600  O                                    4 '0000'                                         10/16/87
* 6103     15601   OVERFLOW INDICATOR OA ASSIGNED TO FILE QPRINT.
            * * * * *   E N D   O F   S O U R C E   * * * * *
```

Figure 11-9 (Part 3 of 5). Source Program Example — RSFBAT (System-Supplied Formats)

```
        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
* 7089      1100   RPG PROVIDES SEPARATE INDICATOR AREA FOR FILE SRCICF.
* 7086      1300   RPG PROVIDES BLOCK OR UNBLOCK SUPPORT FOR FILE DBFILE.
 5728RG1 R01M02  881028            IBM AS/400 RPG/400                 ICFLIB/RSFBAT        03/20/89  15:47:16      Page        6
                       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 DBFILE      DISK         1300D  1700   4900
    03 QPRINT      PRINTER      1400D 12900  13100  13400  15601
    01 SRCICF      WORKSTN      1100D  1500   6400  14400  15000  15400
         $$EVOKNI              14500
         $$SEND                15500
         $$SENDNI              15100
 Field References:
       FIELD       ATTR    REFERENCES (M=MODIFIED D=DEFINED)
       CHKDET      TAG      7300   7500D
       DBDATA      A(80)    1800D 15300
       END         TAG      8500  10200D
       EOFPSW      A(1)     5100  12400D
       EVOKE       EXCPT    3500  14400
       FEEDBK      DS(404)  1100   2000D
       FMTNM       A(8)     2100D 14100
       INVITE      EXCPT   12500  15400
       INVSND      BEGSR    5000  12300D
* 7031 ITMIN       TAG      3400D
       MAJCOD      A(2)     2400D  3600   7200   7300  11300  12600  13700
* 7031 MAJMIN      A(4)     2300D
       MINCOD      A(2)     2500D  7600  13900
       NOTOK       EXCPT    9400  13400
       NOTOKR      TAG      3600   7200   9300D 11300  12600
       OKEND       EXCPT    8400  13100
       PGMDEV      A(10)    2200D 14300
       PTREC       EXCPT    7400  12900
       RCVFLD      A(80)    1600D 13000
       RECDTA      TAG      6300D  7600
       SENDTA      TAG      4800D  5300
       SNDATA      EXCPT   11200  15000
       WCFRTN      BEGSR    5200  11100D
       '03'        LITERAL  3600   7200   7300  11300  12600
       '08'        LITERAL  7600
       '1'         LITERAL  5100  12400
 Indicator References:
       INDICATOR  REFERENCES (M=MODIFIED D=DEFINED)
       LR         10300M
 5728RG1 R01M02  881028            IBM AS/400 RPG/400                 ICFLIB/RSFBAT        03/20/89  15:47:16      Page        7
       OA          1400D 15601
* 7031 80          1700M
* 7031 82          1500M
       98          4900M  5000   6400M
     * * * * *  E N D   O F   C R O S S   R E F E R E N C E  * * * * *
 5728RG1 R01M02  881028            IBM AS/400 RPG/400                 ICFLIB/RSFBAT        03/20/89  15:47:16      Page        8
                     M e s s a g e   S u m m a r y
* QRG6103 Severity:  00   Number:    1
          Message . . . . :   No Overflow Indicator is specified but an
            indicator is assigned to a file and automatic skip to 6 is
            generated.
* QRG7031 Severity:  00   Number:    4
          Message . . . . :   The Name or indicator is not referenced.
* QRG7086 Severity:  00   Number:    1
          Message . . . . :   The RPG handles blocking function for file.
            INFDS contents updated only when blocks of data transferred.
* QRG7089 Severity:  00   Number:    1
          Message . . . . :   The RPG provides Separate-Indicator area for
            file.
     * * * * *  E N D   O F   M E S S A G E   S U M M A R Y  * * * * *
```

Figure 11-9 (Part 4 of 5). Source Program Example — RSFBAT (System-Supplied Formats)

```
5728RG1 R01M02  881028            IBM AS/400 RPG/400              ICFLIB/RSFBAT        03/20/89  15:47:16      Page      9
                              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 . . . . . . . . . . :   156
   Specifications  . . . . . . :   75
   Table Records . . . . . . . :   0
   Comments  . . . . . . . . . :   81
PRM has been called.
Program RSFBAT is placed in library ICFLIB. 00 highest Error-Severity-Code.
        * * * * *  E N D   O F   C O M P I L A T I O N   * * * * *
```

Figure 11-9 (Part 5 of 5). Source Program Example — RSFBAT (System-Supplied Formats)

Target Program Batch Transfer (Example I)

The following describes an RPG/400 batch data transfer target program.

Program Files: The RPG/400 batch transfer target program uses the following files:

TGTICF — An ICF file used to send records to and receive records from the source program.

DBFILE — A database file that contains the records to be sent to the source program.

QPRINT — A printer file used to print the records received from the source program.

DDS Source: The DDS used in the ICF file is illustrated in Figure 11-10. The other files (DBFILE and QPRINT) are program-described and therefore do not require DDS.

```
A*************************************************************
A*                                                           *
A*                          ICF FILE                         *
A*              USED IN BATCH DATA TRANSFER PROGRAM          *
A*                                                           *
A*************************************************************
A*
A*  FILE LEVEL INDICATORS:
A*
A                                     INDARA
A*
A                                     RCVTRNRND(15 'END OF DATA')
A*
A  30                                 DETACH
A*
A                                     INDTXT(30 '30->DETACH TARG-
A                                     ET PROGRAM.')
A*
A                                     RCVDETACH(35 'RECEIVED    -
A                                     DETACHED.')
A*
A*
A*************************************************************
A*                     ICF RECORD FORMATS                    *
A*************************************************************
          R RCVDATA
            RCVFLD        80A
          R SNDDATA
            SNDFLD        80A
          R EVOKPGM
A  50                                 EVOKE(&LIB/&PGMID)
A  50                                 SECURITY(2 'PASSWRD' +
                                              3 'USERID')
A           PGMID         10A  P
A           LIB           10A  P
A         R ENDREC
A         R INVITE
A  45                                 INVITE
```

Figure 11-10. DDS Source for ICF File Used in Batch Data Transfer Target Program

This example acquires all program devices at the beginning of the program. For performance considerations, you may not want to acquire program devices until they are actually needed in the program.

ICF File Creation and Program Device Entry Definition: The command needed to create the ICF file is:

```
CRTICFF  FILE(ICFLIB/TGTICF)  SRCFILE(ICFLIB/QICFPUB)
   SRCMBR(TGTICF)  ACQPGMDEV(PGMDEVB)
   TEXT('TARGET ICF FILE FOR BATCH DATA TRANSFER')
```

The command needed to define the program device entry is:

```
ADDICFDEVE FILE(ICFLIB/TGTICF) PGMDEV(PGMDEVB) RMTLOCNAME(*REQUESTER)
```

Program Explanation: The following describes the structure of the program examples illustrated in Figure 11-11 on page 11-25 and Figure 11-12 on page 11-29. The ICF file used in the first example is defined by the user and uses externally described data formats (DDS). The second example uses the same file, but uses program-described data and system-supplied formats. The reference letters in the explanation below correspond to those in the following program examples.

Although the basic structure of the two examples provided is the same, there are differences because of the use of user-defined formats and system-supplied formats. All output operations to the ICF file in the first example are done using the WRITE statement. All output operations to the ICF file in the second example using system-supplied formats are done using the EXCPT statement.

Differences between the first and second example are described in notes in each of the descriptions.

1 The file specification identifies the files used in the program. TGTICF is the ICF file used to send records to the source program.

The files used in the program are opened at the beginning of the RPG/400 cycle and the ICF program device is implicitly acquired because the ACQPGMDEV parameter was specified on the CRTICFF command.

Note: In the program using system-supplied formats, the input records for TGTICF are explicitly coded since TGTICF is treated as a program-described file. The system-supplied file, QICDMF, can be be used instead of TGTICF. Using the system-supplied file is done by specifying QICDMF in the file specification, or by using an OVRICFF command to change the file name from TGTICF to QICDMF. The OVRICFF command can also be used to change the ACQPGMDEV parameter of the file.

2 FEEDBK is the name of the file information data structure (INFDS) used with TGTICF. It contains the following information:

- Record format-name (FMTNM)
- Program device name (PGMDEV)
- Major/minor return code (MAJMIN, MAJCOD, MINCOD)

3 Read data from the ICF program device (TGTICF) file.

If an error occurs on the read (major return code greater than 03), control passes to **6**. Otherwise, if data is received (major return code not = 03), the data is written to the printer file (QPRINT).

Data records are read until the change-direction indication is received from the source program. When change direction is received, indicator 15 is set on, as defined by the RCVTRNRND keyword in the DDS for the ICF file, and control is passed to **4**.

Note: In the program using system-supplied formats, the minor return code of '00' is checked to verify whether change direction is received.

4 The database file is read and the records sent to the source program until the end of the database file. At this time, the program sets indicator 98 and goes to 9. After returning from 9, control is passed to 5.

If it is not the last record, the data is moved to field SNDFLD, and the program goes to 8 to write the record to the ICF program device. When control returns from 8, the next database record is read.

5 After the last database record has been read, the following message is written to the print file:

```
RTDBAT HAS COMPLETED NORMALLY
```

Control passes to 7.

Note: The program name is RSFBAT in the program using system-supplied formats.

6 When an I/O operation to the ICF file (TGTICF) completes unsuccessfully, the following message is written to the print file:

```
RTDBAT HAS COMPLETED ABNORMALLY
```

Control passes to 7.

Note: The program name is RTFBAT in the program using system-supplied formats.

7 The program ends the job by setting on LR indicator and returning to caller of the program. The ICF file is closed and the session is ended at the end of the RPG cycle.

8 This subroutine is called to write data to the ICF program device using the format SNDDATA. If an error occurs, the program goes to 6 and a message is printed.

Note: The $$SENDNI format is used instead of the user-defined SNDDATA format in the program using system-supplied formats.

9 This subroutine is called to issue a detach request to the ICF program device using format ENDREC. If an error occurs, the program goes to 6 and a message is printed.

Note: The $$SENDET format is used instead of the user-defined ENDREC format in the program using system-supplied formats.

```
5728RG1 R01M02  881028           IBM AS/400 RPG/400                ICFLIB/RTDBAT       03/20/89  15:46:53      Page       1
Compiler . . . . . :   IBM AS/400 RPG/400
Command Options:
  Program  . . . . . :   ICFLIB/RTDBAT
  Source file  . . . . . :   ICFLIB/QICFPUB
  Source member  . . . . . :   *PGM
  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
  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 . . . . . :   RTDBAT
  File . . . . . :   QICFPUB
  Library  . . . . . :   ICFLIB
  Last Change  . . . . . :   03/20/89  15:40:57
  Description  . . . . . :   rpg batch file transfer using dds source
5728RG1 R01M02  881028           IBM AS/400 RPG/400                ICFLIB/RTDBAT       03/20/89  15:46:53      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  H*****************************************************************                 10/16/87
     200  H*                                                                               03/20/89
     300  H*   THIS PROGRAM IS EVOKED BY THE SOURCE PROGRAM AND RECEIVES                    03/20/89
     400  H*   RECORDS FROM IT. WHEN THE SOURCE PROGRAM IS DONE SENDING                     03/20/89
     500  H*   DATA, THIS PROGRAM SENDS ITS OWN RECORDS.  WHEN FINISHED,                   03/20/89
     600  H*   THIS PROGRAM WILL SEND A DETACH REQUEST TO THE SOURCE                        03/20/89
     700  H*   PROGRAM TO END THE SESSION AND JOB.                                          03/20/89
     800  H*                                                                               03/20/89
     900  H*****************************************************************                 10/16/87
    1000  H* 1                                                                             10/16/87
    1100  FTGTICF  CF  E                    WORKSTN                                       10/16/87
    1200  F                                                KINFDS FEEDBK                  10/16/87
           RECORD FORMAT(S):  LIBRARY ICFLIB FILE TGTICF.
                    EXTERNAL FORMAT RCVDATA RPG NAME RCVDATA
                    EXTERNAL FORMAT SNDDATA RPG NAME SNDDATA
                    EXTERNAL FORMAT EVOKPGM RPG NAME EVOKPGM
                    EXTERNAL FORMAT ENDREC RPG NAME ENDREC
                    EXTERNAL FORMAT INVITE RPG NAME INVITE
    1300  FDBFILE  IF  F      80            DISK                                          10/16/87
    1400  FQPRINT  O   F     132            PRINTER                                       10/16/87
    1500  IDBFILE  NS  80                                                                10/16/87
    1600  I                                       1  80 DBDATA                           10/16/87
    1700  I*                                                                             10/16/87
 A000000   INPUT  FIELDS FOR RECORD RCVDATA FILE TGTICF FORMAT RCVDATA.
 A000001                                         1  80 RCVFLD
 B000000   INPUT  FIELDS FOR RECORD SNDDATA FILE TGTICF FORMAT SNDDATA.
 B000001                                         1  80 SNDFLD
 C000000   INPUT  FIELDS FOR RECORD EVOKPGM FILE TGTICF FORMAT EVOKPGM.
 D000000   INPUT  FIELDS FOR RECORD ENDREC FILE TGTICF FORMAT ENDREC.
 E000000   INPUT  FIELDS FOR RECORD INVITE FILE TGTICF FORMAT INVITE.
    1800  IFEEDBK 2       DS                                                                 10/16/87
    1900  I                                      38  47 FMTNM                            10/16/87
    2000  I                                     273 282 PGMDEV                           10/16/87
    2100  I                                     401 404 MAJMIN                           10/16/87
    2200  I                                     401 402 MAJCOD                           10/16/87
    2300  I                                     403 404 MINCOD                           10/16/87
```

Figure 11-11 (Part 1 of 4). Target Program Example — RTDBAT (User-Defined Formats)

```
    2400 C*************************************************************                10/16/87
    2500 C*                                                                            03/20/89
    2600 C*   THIS PROGRAM ISSUES A READ OPERATION TO THE PROGRAM DEVICE                03/20/89
    2700 C*   TO RECEIVE RECORDS FROM THE SOURCE PROGRAM UNTIL THE                      03/20/89
    2800 C*   RCVTRNRND INDICATOR (*IN15) IS SET. EACH RECORD RECEIVED IS               03/20/89
    2900 C*   PRINTED TO THE PRINT FILE.                                               03/20/89
    3000 C*                                                                            10/16/87
5728RG1 R01M02  881028            IBM AS/400 RPG/400                 ICFLIB/RTDBAT      03/20/89  15:46:53      Page        3
SEQUENCE                                                                    IND    DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE     LINE   ID
    3100 C*   IF AN ERROR OCCURS, AN ERROR MESSAGE IS PRINTED AND THE                   10/16/87
    3200 C*   JOB IS ENDED.                                                            10/16/87
    3300 C*                                                                            03/20/89
    3400 C*************************************************************                10/16/87
    3500 C* 3                                                                         10/16/87
    3600 C           RECDTA    TAG                                                   10/16/87
    3700 C                     READ TGTICF                 98                3       10/16/87
    3800 C           MAJCOD    CABGT'03'      NOTOKR         ERROR?                  10/16/87
    3900 C           MAJCOD    CABEQ'03'      CHKTRN         NO DATA ?               10/16/87
    4000 C                     EXCPTPTREC                                           10/16/87
    4100 C           CHKTRN    TAG                                                   10/16/87
    4200 C           *IN15     CABNE'1'       RECDTA         RCVTRNRND ?             10/16/87
    4300 C*************************************************************                10/16/87
    4400 C*                                                                            03/20/89
    4500 C*   WHEN A RCVTRNRND INDICATION IS RECEIVED, THE PROGRAM STARTS              10/16/87
    4600 C*   SENDING THE RECORDS TO THE SOURCE PROGRAM. RECORDS ARE SENT              10/16/87
    4700 C*   UNTIL AN END OF FILE IS REACHED ON THE DATABASE FILE. AT                 03/20/89
    4800 C*   THIS TIME, A DETACH REQUEST IS SENT TO THE SOURCE PROGRAM.               03/20/89
    4900 C*                                                                            03/20/89
    5000 C*************************************************************                10/16/87
    5100 C* 4                                                                         10/16/87
    5200 C           SENDTA    TAG                                                   10/16/87
    5300 C                     READ DBFILE                 98                3       10/16/87
    5400 C   98                EXSR ENDSES                   SEND DETACH             10/16/87
    5500 C           EOFPSW    IFNE '1'                                        B001   10/16/87
    5600 C                     MOVE DBDATA    SNDFLD                            001   10/16/87
    5700 C                     EXSR WCFRTN                                      001   10/16/87
    5800 C                     GOTO SENDTA                   SEND DATA          001   10/16/87
    5900 C                     END                                             E001   10/16/87
    6000 C*************************************************************                10/16/87
    6100 C*                                                                            03/20/89
    6200 C*   WHEN THE END OF FILE IS REACHED, AN EOJ MESSAGE IS                       03/20/89
    6300 C*   PRINTED AND THE PROGRAM GOES TO END.                                     03/20/89
    6400 C*                                                                            03/20/89
    6500 C*************************************************************                10/16/87
    6600 C* 5                                                                         10/16/87
    6700 C                     EXCPTOKEND                                           10/16/87
    6800 C                     GOTO END                                             10/16/87
    6900 C*************************************************************                10/16/87
    7000 C*                                                                            03/20/89
    7100 C* WHEN AN I/O OPERATION ERROR IS DETECTED, AN ABNORMAL                       10/16/87
    7200 C* TERMINATION MESSAGE IS PRINTED AND THE PROGRAM ENDS.                      10/16/87
    7300 C*                                                                            03/20/89
    7400 C*************************************************************                10/16/87
    7500 C* 6                                                                         10/16/87
    7600 C           NOTOKR    TAG                                                   10/16/87
    7700 C                     EXCPTNOTOK                                           10/16/87
    7800 C*************************************************************                10/16/87
    7900 C*                                                                            03/20/89
    8000 C*   WHEN PROCESSING IS FINISHED, THE LAST RECORD SWITCH IS SET               10/16/87
    8100 C*   AND THE PROGRAM IS ENDED.                                                10/16/87
    8200 C*                                                                            03/20/89
    8300 C*************************************************************                10/16/87
```

Figure 11-11 (Part 2 of 4). Target Program Example — RTDBAT (User-Defined Formats)

```
    8400  C* 7                                                                  10/16/87
5728RG1 R01M02  881028            IBM AS/400 RPG/400                 ICFLIB/RTDBAT        03/20/89  15:46:53      Page        4
SEQUENCE                                                                        IND    DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE     LINE   ID
    8500  C           END       TAG                                             10/16/87
    8600  C                     SETON                     LR                1   10/16/87
    8700  C                     RETRN                                           10/16/87
    8800  C*****************************************************************   10/16/87
    8900  C*                                                                    03/20/89
    9000  C*   THIS SUBROUTINE IS CALLED TO SEND DATA TO THE SOURCE PRO-         03/20/89
    9100  C*   GRAM.  IF A SESSION ERROR OCCURS, AN ABNORMAL END                 03/20/89
    9200  C*   MESSAGE IS PRINTED, THE LR SWITCH IS SET, AND THE JOB ENDS.       03/20/89
    9300  C*                                                                    03/20/89
    9400  C*****************************************************************   10/16/87
    9500  C* 8                                                                  10/16/87
    9600  C           WCFRTN    BEGSR                                           10/16/87
    9700  C                     WRITESNDDATA                                    10/16/87
    9800  C           MAJCOD    CABGT'03'      NOTOKR           ERROR?          10/16/87
    9900  C                     ENDSR                                           10/16/87
   10000  C*****************************************************************   10/16/87
   10100  C*                                                                    03/20/89
   10200  C* THIS SUBROUTINE IS CALLED AT END OF FILE TO SEND AN                 10/16/87
   10300  C* INDICATION TO THE SOURCE SYSTEM THAT TRANSMISSION IS ENDED.         03/20/89
   10400  C* THE END OF FILE SWITCH IS ALSO SET TO END THE JOB.                  10/16/87
   10500  C*                                                                    03/20/89
   10600  C*****************************************************************   10/16/87
   10700  C* 9                                                                  10/16/87
   10800  C           ENDSES    BEGSR                                           10/16/87
   10900  C                     MOVE '1'       *IN30            ACTV DETACH     10/16/87
   11000  C                     MOVE '1'       EOFPSW  1                       10/16/87
   11100  C                     WRITEENDREC                     SEND DETACH     10/16/87
   11200  C           MAJCOD    CABGT'03'      NOTOKR           ERROR?          10/16/87
   11300  C                     ENDSR                                           10/16/87
   11400  C*****************************************************************   10/16/87
   11500  OQPRINT  E 1              PTREC                                       10/16/87
   11600  O                         RCVFLD    80                                10/16/87
   11700  O        E 1              OKEND                                       10/16/87
   11800  O                                   21 'RTDBAT HAS COMPLETED '        10/16/87
   11900  O                                   30 'NORMALLY.'                    10/16/87
   12000  O        E 1              NOTOK                                       10/16/87
   12100  O                                   21 'RTDBAT HAS COMPLETED '        10/16/87
   12200  O                                   32 'ABNORMALLY.'                  10/16/87
   12300  O                         MAJCOD    37                                10/16/87
   12400  O                                   38 '/'                            10/16/87
   12500  O                         MINCOD    40                                10/16/87
   12600  O                                   49 'FORMAT:'                      10/16/87
   12700  O                         FMTNM     60                                10/16/87
   12800  O                                   69 'DEVICE:'                      10/16/87
   12900  O                         PGMDEV    80                                03/20/89
* 6103     12901   OVERFLOW INDICATOR OA ASSIGNED TO FILE QPRINT.
  F000000   OUTPUT FIELDS FOR RECORD SNDDATA FILE TGTICF FORMAT SNDDATA.
  F000001                           SNDFLD    80  CHAR   80
  G000000   OUTPUT FIELDS FOR RECORD ENDREC FILE TGTICF FORMAT ENDREC.
          * * * * *  E N D  O F  S O U R C E  * * * * *
 5728RG1 R01M02  881028            IBM AS/400 RPG/400                 ICFLIB/RTDBAT        03/20/89  15:46:53      Page        5
        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
* 7089      1100   RPG PROVIDES SEPARATE INDICATOR AREA FOR FILE TGTICF.
* 7086      1300   RPG PROVIDES BLOCK OR UNBLOCK SUPPORT FOR FILE DBFILE.
 5728RG1 R01M02  881028            IBM AS/400 RPG/400                 ICFLIB/RTDBAT        03/20/89  15:46:53      Page        6
                      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  DBFILE      DISK          1300D    1500     5300
    03  QPRINT      PRINTER       1400D   11500    11700    12000    12901
    01  TGTICF      WORKSTN       1100D    3700
          ENDREC                  1100D D000000    11100  G000000
          EVOKPGM                 1100D C000000
          INVITE                  1100D E000000
          RCVDATA                 1100D A000000
          SNDDATA                 1100D B000000     9700  F000000
 Field References:
        FIELD       ATTR    REFERENCES (M=MODIFIED D=DEFINED)
        *IN15       A(1)       4200
```

Figure 11-11 (Part 3 of 4). Target Program Example — RTDBAT (User-Defined Formats)

```
        *IN30       A(1)      10900M
        CHKTRN      TAG        3900     4100D
        DBDATA      A(80)      1600D    5600
        END         TAG        6800     8500D
        ENDSES      BEGSR      5400    10800D
        EOFPSW      A(1)       5500    11000D
        FEEDBK      DS(404)    1100     1800D
        FMTNM       A(8)       1900D   12700
        MAJCOD      A(2)       2200D    3800     3900     9800    11200
                               12300
* 7031  MAJMIN      A(4)       2100D
        MINCOD      A(2)       2300D   12500
        NOTOK       EXCPT      7700    12000
        NOTOKR      TAG        3800     7600D    9800    11200
        OKEND       EXCPT      6700    11700
        PGMDEV      A(10)      2000D   12900
        PTREC       EXCPT      4000    11500
        RCVFLD      A(80)   A000001D   11600
        RECDTA      TAG         3600D    4200
        SENDTA      TAG         5200D    5800
        SNDFLD      A(80)   B000001D    5600M F000001D
        WCFRTN      BEGSR      5700     9600D
        '03'        LITERAL    3800     3900     9800    11200
        '1'         LITERAL    4200     5500    10900    11000
 Indicator References:
        INDICATOR  REFERENCES (M=MODIFIED D=DEFINED)
 5728RG1 R01M02  881028             IBM AS/400 RPG/400                 ICFLIB/RTDBAT        03/20/89  15:46:53      Page        7
        *IN          4200    10900M
        LR           8600M
        OA           1400D   12901
        15           4200
* 7031  30          10900M
* 7031  35
* 7031  45
* 7031  50
* 7031  80           1500M
        98           3700M    5300M    5400
      * * * * *  E N D  O F  C R O S S  R E F E R E N C E  * * * * *
 5728RG1 R01M02  881028             IBM AS/400 RPG/400                 ICFLIB/RTDBAT        03/20/89  15:46:53      Page        8
                        M e s s a g e  S u m m a r y
* QRG6103 Severity:  00   Number:    1
          Message . . . . :   No Overflow Indicator is specified but an
            indicator is assigned to a file and automatic skip to 6 is
            generated.
* QRG7031 Severity:  00   Number:    6
          Message . . . . :   The Name or indicator is not referenced.
* QRG7086 Severity:  00   Number:    1
          Message . . . . :   The RPG handles blocking function for file.
            INFDS contents updated only when blocks of data transferred.
* QRG7089 Severity:  00   Number:    1
          Message . . . . :   The RPG provides Separate-Indicator area for
            file.
      * * * * *  E N D  O F  M E S S A G E  S U M M A R Y  * * * * *
 5728RG1 R01M02  881028             IBM AS/400 RPG/400                 ICFLIB/RTDBAT        03/20/89  15:46:53      Page        9
                          F i n a l  S u m m a r y
 Message Count:  (by Severity Number)
            TOTAL    00     10     20     30     40     50
              9       9      0      0      0      0      0
 Program Source Totals:
    Records . . . . . . . . . . :   129
    Specifications  . . . . . . :   59
    Table Records . . . . . . . :   0
    Comments  . . . . . . . . . :   70
 PRM has been called.
 Program RTDBAT is placed in library ICFLIB. 00 highest Error-Severity-Code.
         * * * * *  E N D  O F  C O M P I L A T I O N  * * * * *
```

Figure 11-11 (Part 4 of 4). Target Program Example — RTDBAT (User-Defined Formats)

```
5728RG1 R01M02  881028           IBM AS/400 RPG/400                ICFLIB/RTFBAT       03/20/89  15:47:37      Page        1
Compiler . . . . . :   IBM AS/400 RPG/400
Command Options:
  Program  . . . . . :   ICFLIB/RTFBAT
  Source file  . . . . . :   ICFLIB/QICFPUB
  Source member  . . . . . :   *PGM
  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
  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 . . . . . :   RTFBAT
  File . . . . . :   QICFPUB
  Library  . . . . . :   ICFLIB
  Last Change  . . . . . :   03/20/89  15:15:51
  Description  . . . . . :   rpg batch file transfer using $$FORMAT
5728RG1 R01M02  881028           IBM AS/400 RPG/400                ICFLIB/RTFBAT       03/20/89  15:47:37      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  H*****************************************************************          10/16/87
     200  H*                                                                         03/20/89
     300  H*   THIS PROGRAM IS EVOKED BY A SOURCE PROGRAM AND RECEIVES               03/20/89
     400  H*   RECORDS FROM IT. WHEN THE SOURCE PROGRAM IS DONE SENDING              03/20/89
     500  H*   DATA, THIS PROGRAM SENDS ITS OWN RECORDS TO THE SOURCE.               03/20/89
     600  H*   WHEN FINISHED, THIS PROGRAM SENDS A DETACH REQUEST TO THE             03/20/89
     700  H*   SOURCE PROGRAM TO END THE SESSION AND JOB.                            03/20/89
     800  H*                                                                         03/20/89
     900  H*****************************************************************          10/16/87
    1000  H* 1                                                                      10/16/87
    1100  FTGTICF  CF  F     84           WORKSTN                                    10/16/87
    1200  F                                        KINFDS FEEDBK                    10/16/87
    1300  FDBFILE  IF  F     80           DISK                                       10/16/87
    1400  FQPRINT  O   F    132           PRINTER                                    10/16/87
    1500  ITGTICF  NS  80                                                           10/16/87
    1600  I                                     1  80 RCVFLD                        10/16/87
    1700  IDBFILE  NS  80                                                           10/16/87
    1800  I                                     1  80 DBDATA                        10/16/87
    1900  I*                                                                        10/16/87
    2000  I* 2                                                                      10/16/87
    2100  IFEEDBK      DS                                                           10/16/87
    2200  I                                    38  47 FMTNM                         10/16/87
    2300  I                                   273 282 PGMDEV                        10/16/87
    2400  I                                   401 404 MAJMIN                        10/16/87
    2500  I                                   401 402 MAJCOD                        10/16/87
    2600  I                                   403 404 MINCOD                        10/16/87
    2700  C*****************************************************************          10/16/87
    2800  C*                                                                        03/20/89
    2900  C*   THIS PROGRAM ISSUES THE READ OPERATION TO THE PROGRAM DEVICE          03/20/89
    3000  C*   TO RECEIVE RECORDS FROM THE SOURCE PROGRAM UNTIL THE CHANGE           03/20/89
    3100  C*   DIRECTION INDICATION IS RECEIVED. EACH RECORD RECEIVED IS             03/20/89
    3200  C*   PRINTED TO THE PRINT FILE.                                           03/20/89
    3300  C*                                                                        10/16/87
    3400  C*   IF AN ERROR OCCURS, AN ERROR MESSAGE IS PRINTED AND THE               10/16/87
    3500  C*   JOB IS ENDED.                                                        10/16/87
    3600  C*                                                                        03/20/89
```

Figure 11-12 (Part 1 of 4). Target Program Example — RTFBAT (System-Supplied Formats)

```
    3700  C*****************************************************************                10/16/87
    3800  C* 3                                                                              10/16/87
    3900  C           RECDTA    TAG                                                        10/16/87
    4000  C                     READ TGTICF                 98                   3         10/16/87
    4100  C           MAJCOD    CABGT'03'      NOTOKR         ERROR?                      10/16/87
    4200  C           MAJCOD    CABEQ'03'      CHKTRN         NO DATA ?                   10/16/87
    4300  C                     EXCPTPTREC                                                 10/16/87
    4400  C           CHKTRN    TAG                                                        10/16/87
5728RG1 R01M02  881028            IBM AS/400 RPG/400                 ICFLIB/RTFBAT        03/20/89  15:47:37      Page        3
SEQUENCE                                                                      IND    DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE     LINE   ID
    4500  C           MINCOD    CABNE'00'      RECDTA         RCVTRNRND ?                 10/16/87
    4600  C*****************************************************************                10/16/87
    4700  C*                                                                              03/20/89
    4800  C*   WHEN A RCVTRNRND INDICATION IS RECEIVED, THE PROGRAM STARTS                10/16/87
    4900  C*   SENDING RECORDS TO THE SOURCE PROGRAM. RECORDS ARE SENT UNTIL              10/16/87
    5000  C*   THE END OF FILE IS REACHED ON THE DATABASE FILE. AT THIS TIME              03/20/89
    5100  C*   A DETACH REQUEST IS SENT TO THE SOURCE PROGRAM.                            03/20/89
    5200  C*                                                                              03/20/89
    5300  C*****************************************************************                10/16/87
    5400  C* 4                                                                            10/16/87
    5500  C           SENDTA    TAG                                                        10/16/87
    5600  C                     READ DBFILE                 98                   3         10/16/87
    5700  C   98                EXSR ENDSES                   SEND DETACH               10/16/87
    5800  C           EOFPSW    IFNE '1'                                           B001    10/16/87
    5900  C                     EXSR WCFRTN                                         001    10/16/87
    6000  C                     GOTO SENDTA                   SEND DATA             001    10/16/87
    6100  C                     END                                                E001    10/16/87
    6200  C*************************************************************                   10/16/87
    6300  C*                                                                              03/20/89
    6400  C*   WHEN THE END OF FILE IS REACHED, AN EOJ MESSAGE IS                         10/16/87
    6500  C*   PRINTED, AND CONTROL GOES TO END.                                          03/20/89
    6600  C*                                                                              03/20/89
    6700  C*************************************************************                   10/16/87
    6800  C* 5                                                                            10/16/87
    6900  C                     EXCPTOKEND                                                 10/16/87
    7000  C                     GOTO END                                                   10/16/87
    7100  C*************************************************************                   10/16/87
    7200  C*                                                                              03/20/89
    7300  C* WHEN AN I/O OPERATION ERROR IS DETECTED, AN ABNORMAL                         10/16/87
    7400  C* TERMINATION MESSAGE IS PRINTED AND THE PROGRAM ENDS.                         10/16/87
    7500  C*                                                                              03/20/89
    7600  C*************************************************************                   10/16/87
    7700  C* 6                                                                            10/16/87
    7800  C           NOTOKR    TAG                                                        10/16/87
    7900  C                     EXCPTNOTOK                                                 10/16/87
    8000  C*****************************************************************                10/16/87
    8100  C*                                                                              03/20/89
    8200  C*   WHEN PROCESSING IS FINISHED, THE LAST RECORD SWITCH IS SET                 10/16/87
    8300  C*   AND THE PROGRAM IS ENDED.                                                  10/16/87
    8400  C*                                                                              03/20/89
    8500  C*****************************************************************                10/16/87
    8600  C* 7                                                                            10/16/87
    8700  C           END       TAG                                                        10/16/87
    8800  C                     SETON                       LR                   1         10/16/87
    8900  C                     RETRN                                                      10/16/87
    9000  C*****************************************************************                10/16/87
    9100  C*                                                                              03/20/89
    9200  C*   THIS SUBROUTINE IS CALLED TO SEND DATA TO THE SOURCE PROGRAM.              03/20/89
    9300  C*   IF A SESSION ERROR OCCURS, AN ABNORMAL END MESSAGE IS                      03/20/89
    9400  C*   PRINTED, THE LR SWITCH IS SET, AND THE JOB IS ENDED.                       03/20/89
    9500  C*                                                                              03/20/89
    9600  C*****************************************************************                10/16/87
    9700  C* 8                                                                            10/16/87
    9800  C           WCFRTN    BEGSR                                                      10/16/87
```

Figure 11-12 (Part 2 of 4). Target Program Example — RTFBAT (System-Supplied Formats)

```
5728RG1 R01M02  881028            IBM AS/400 RPG/400                 ICFLIB/RTFBAT       03/20/89  15:47:37      Page       4
SEQUENCE                                                                                  IND    DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE     LINE   ID
    9900  C                   EXCPTSNDATA                                                      10/16/87
   10000  C           MAJCOD  CABGT'03'      NOTOKR          ERROR?                           10/16/87
   10100  C                   ENDSR                                                            10/16/87
   10200  C*************************************************************                     10/16/87
   10300  C*                                                                                   03/20/89
   10400  C* THIS SUBROUTINE IS CALLED AT END OF FILE TO SEND AN                               10/16/87
   10500  C* INDICATION TO THE LOCAL SYSTEM THAT TRANSMISSION IS ENDED.                        10/16/87
   10600  C* THE END OF FILE SWITCH IS SET TO END THE JOB.                                    03/20/89
   10700  C*                                                                                   03/20/89
   10800  C*************************************************************                     10/16/87
   10900  C* 9                                                                               10/16/87
   11000  C           ENDSES  BEGSR                                                          10/16/87
   11100  C                   MOVE '1'       *IN30           ACTV DETACH                     10/16/87
   11200  C                   MOVE '1'       EOFPSW  1                                       10/16/87
   11300  C                   EXCPTENDREC                    SEND DETACH                     10/16/87
   11400  C           MAJCOD  CABGT'03'      NOTOKR          ERROR?                          10/16/87
   11500  C                   ENDSR                                                          10/16/87
   11600  C*****************************************************************               10/16/87
   11700  OQPRINT  E 1            PTREC                                                     10/16/87
   11800  O                       RCVFLD    80                                              10/16/87
   11900  O        E 1            OKEND                                                     10/16/87
   12000  O                                  21 'RTFBAT HAS COMPLETED '                     10/16/87
   12100  O                                  30 'NORMALLY.'                                 10/16/87
   12200  O        E 1            NOTOK                                                     10/16/87
   12300  O                                  21 'RTFBAT HAS COMPLETED '                     10/16/87
   12400  O                                  32 'ABNORMALLY.'                               10/16/87
   12500  O                       MAJCOD    37                                              10/16/87
   12600  O                                  38 '/'                                         10/16/87
   12700  O                       MINCOD    40                                              10/16/87
   12800  O                                  49 'FORMAT:'                                   10/16/87
   12900  O                       FMTNM     60                                              10/16/87
   13000  O                                  69 'DEVICE:'                                   10/16/87
   13100  O                       PGMDEV    80                                              03/20/89
   13200  OTGTICF  E              SNDATA                                                    10/16/87
   13300  O                                  K8 '$$SENDNI'                                  10/16/87
   13400  O                                   4 '0080'                                      10/16/87
   13500  O                       DBDATA    84                                              10/16/87
   13600  O        E              ENDREC                                                    10/16/87
   13700  O                                  K8 '$$SENDET'                                  10/16/87
   13800  O                                   4 '0000'                                      10/16/87
* 6103     13801   OVERFLOW INDICATOR OA ASSIGNED TO FILE QPRINT.
          * * * * *  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
* 7089      1100   RPG PROVIDES SEPARATE INDICATOR AREA FOR FILE TGTICF.
* 7086      1300   RPG PROVIDES BLOCK OR UNBLOCK SUPPORT FOR FILE DBFILE.
 5728RG1 R01M02  881028            IBM AS/400 RPG/400                 ICFLIB/RTFBAT       03/20/89  15:47:37      Page       5
                      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  DBFILE      DISK         1300D  1700   5600
    03  QPRINT      PRINTER      1400D 11700  11900  12200  13801
    01  TGTICF      WORKSTN      1100D  1500   4000  13200  13600
          $$SENDET              13700
          $$SENDNI              13300
 Field References:
        FIELD       ATTR    REFERENCES (M=MODIFIED D=DEFINED)
        *IN30       A(1)    11100M
        CHKTRN      TAG      4200   4400D
        DBDATA      A(80)    1800D 13500
        END         TAG      7000   8700D
        ENDREC      EXCPT   11300  13600
        ENDSES      BEGSR    5700  11000D
        EOFPSW      A(1)     5800  11200D
        FEEDBK      DS(404)  1100   2100D
        FMTNM       A(8)     2200D 12900
```

Figure 11-12 (Part 3 of 4). Target Program Example — RTFBAT (System-Supplied Formats)

```
       MAJCOD     A(2)     2500D  4100   4200  10000  11400  12500
* 7031 MAJMIN     A(4)     2400D
       MINCOD     A(2)     2600D  4500  12700
       NOTOK      EXCPT    7900  12200
       NOTOKR     TAG      4100   7800D 10000  11400
       OKEND      EXCPT    6900  11900
       PGMDEV     A(10)    2300D 13100
       PTREC      EXCPT    4300  11700
       RCVFLD     A(80)    1600D 11800
       RECDTA     TAG      3900D  4500
       SENDTA     TAG      5500D  6000
       SNDATA     EXCPT    9900  13200
       WCFRTN     BEGSR    5900   9800D
       '00'       LITERAL  4500
       '03'       LITERAL  4100   4200  10000  11400
       '1'        LITERAL  5800  11100  11200
 Indicator References:
       INDICATOR  REFERENCES (M=MODIFIED D=DEFINED)
       *IN       11100M
       LR         8800M
       OA         1400D 13801
 5728RG1 R01M02  881028            IBM AS/400 RPG/400                 ICFLIB/RTFBAT        03/20/89  15:47:37      Page       6
* 7031  30        11100M
* 7031  80         1500M  1700M
       98         4000M  5600M  5700
     * * * * *  E N D   O F   C R O S S   R E F E R E N C E  * * * * *
 5728RG1 R01M02  881028            IBM AS/400 RPG/400                 ICFLIB/RTFBAT        03/20/89  15:47:37      Page       7
                      M e s s a g e   S u m m a r y
* QRG6103 Severity:  00   Number:    1
          Message . . . . :   No Overflow Indicator is specified but an
            indicator is assigned to a file and automatic skip to 6 is
            generated.
* QRG7031 Severity:  00   Number:    3
          Message . . . . :   The Name or indicator is not referenced.
* QRG7086 Severity:  00   Number:    1
          Message . . . . :   The RPG handles blocking function for file.
            INFDS contents updated only when blocks of data transferred.
* QRG7089 Severity:  00   Number:    1
          Message . . . . :   The RPG provides Separate-Indicator area for
            file.
     * * * * *  E N D   O F   M E S S A G E   S U M M A R Y  * * * * *
 5728RG1 R01M02  881028            IBM AS/400 RPG/400                 ICFLIB/RTFBAT        03/20/89  15:47:37      Page       8
                        F i n a l   S u m m a r y
 Message Count:  (by Severity Number)
           TOTAL    00     10     20     30     40     50
             6       6      0      0      0      0      0
 Program Source Totals:
    Records . . . . . . . . . . :   138
    Specifications  . . . . . . :   67
    Table Records . . . . . . . :   0
    Comments  . . . . . . . . . :   71
 PRM has been called.
 Program RTFBAT is placed in library ICFLIB. 00 highest Error-Severity-Code.
         * * * * *  E N D   O F   C O M P I L A T I O N  * * * * *
```

Figure 11-12 (Part 4 of 4). Target Program Example — RTFBAT (System-Supplied Formats)

Multiple-Session Inquiry (Example II)

This example illustrates an interactive inquiry application that communicates with multiple ICF sessions. A source AS/400 system program accepts inquiries from a display device and sends a request to one of four AS/400 systems. The source program communicates with the display device through a display file, and with the four remote systems through a single ICF file.

The purpose of this example is to show multiple sessions from a single ICF file. The source program communicates with four sessions. From the viewpoint of each of the four target programs, the requester is the only session. Therefore, the target programs do not require any unique logic to support the multiple-session source.

Both the source program and the target program are described. The same target program is evoked in each of the four separate remote systems. Therefore, only one target program is shown in the programming example.

Transaction Flow of the Multiple-Session Inquiry (Example II)

The program shown in Figure 11-13 on page 11-34 is started from a display station. Both the display and the ICF files are opened. CIWS00 is the *REQUESTER device, and is acquired when the display file opens. CIWS00 is acquired because DEV(*REQUESTER) was specified when the display file was created. Since the ICF file was created with ACQPGMDEV(*NONE), no ICF devices are acquired during open processing.

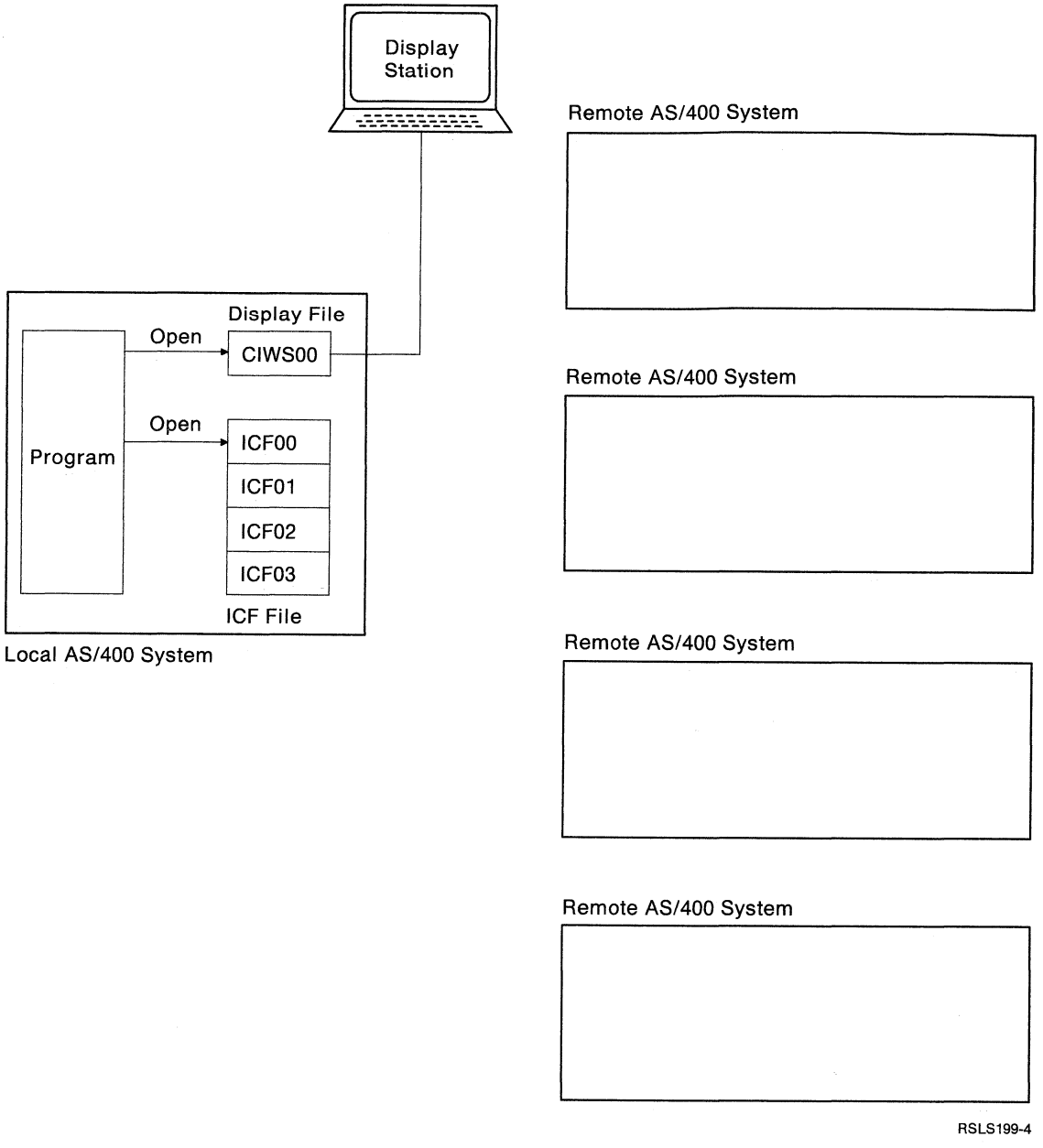


Figure 11-13. Program Starts at Display Station

All other program devices must be explicitly acquired by the program, as shown in Figure 11-14.

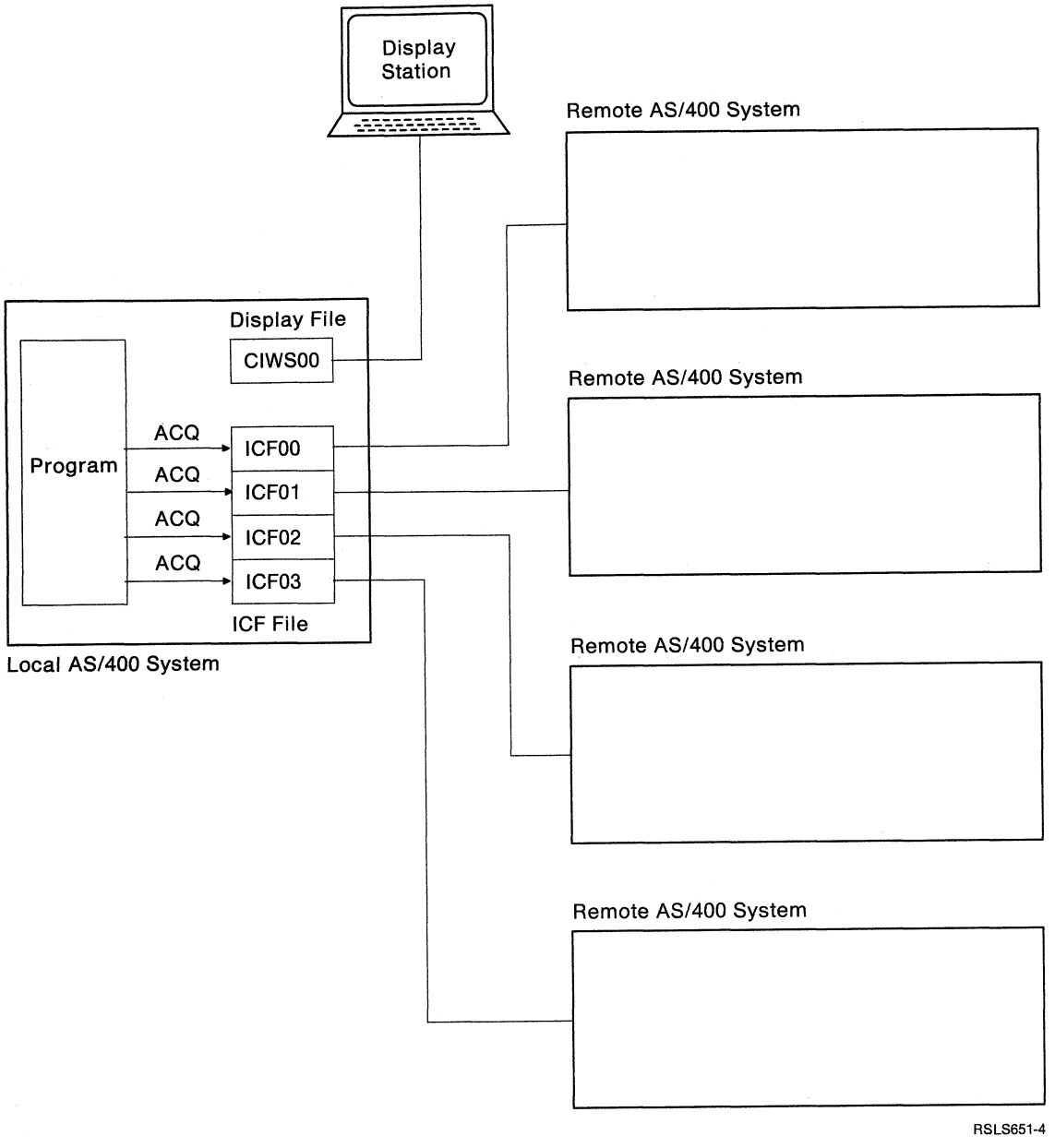


Figure 11-14. Program Devices Explicitly Acquired

All target programs are started with an evoke, as shown in Figure 11-15.

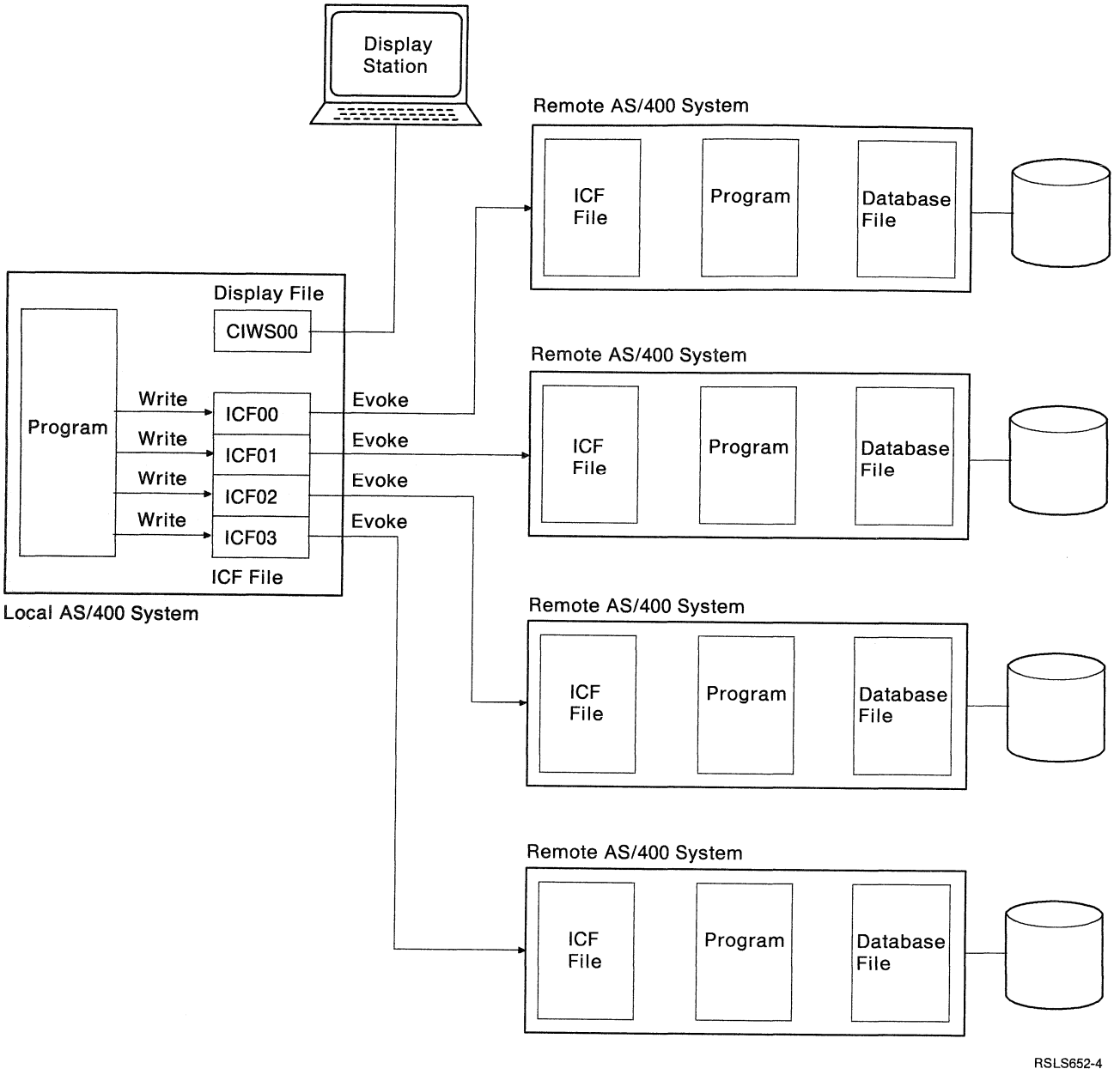


Figure 11-15. Evoke Starts Target Programs

The source program uses a specific program device name. Each target program uses an ICF file with a program device name that is associated with the requester. The target program's only session is the one used to communicate with the source program. The ICF file is implicitly opened by the RPG/400 language support when the target program is started. Since the file was created with the requesting program device specified on the ACQPGMDEV parameter, the requesting program device is acquired with the implicit open.

The main menu is written to the display station on the local system and the program waits for a request from the display station, as shown in Figure 11-16.

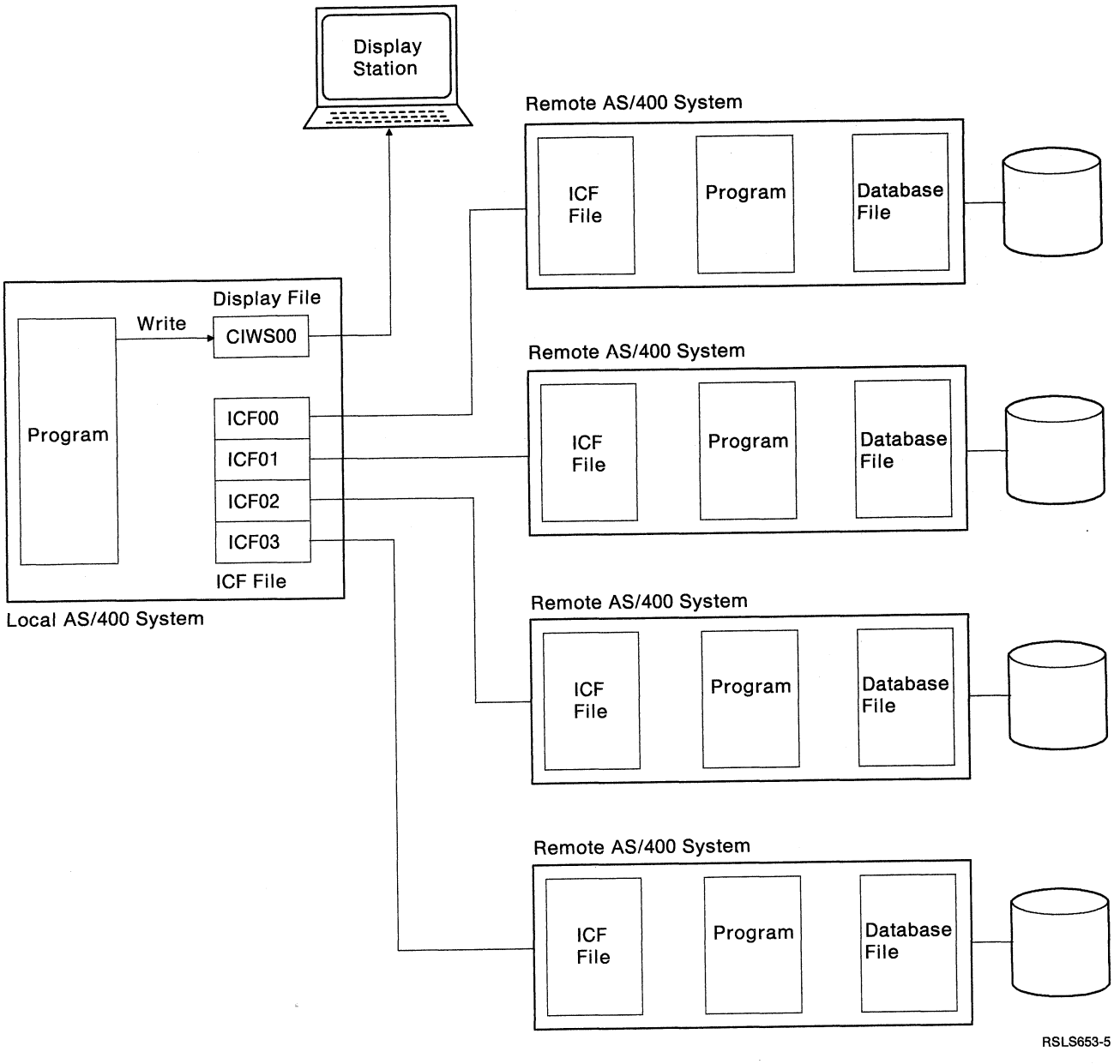


Figure 11-16. Main Menu Written to Display Station

The source program sends an inquiry request to one of the remote systems based on the request made from the display station, as shown in Figure 11-17.

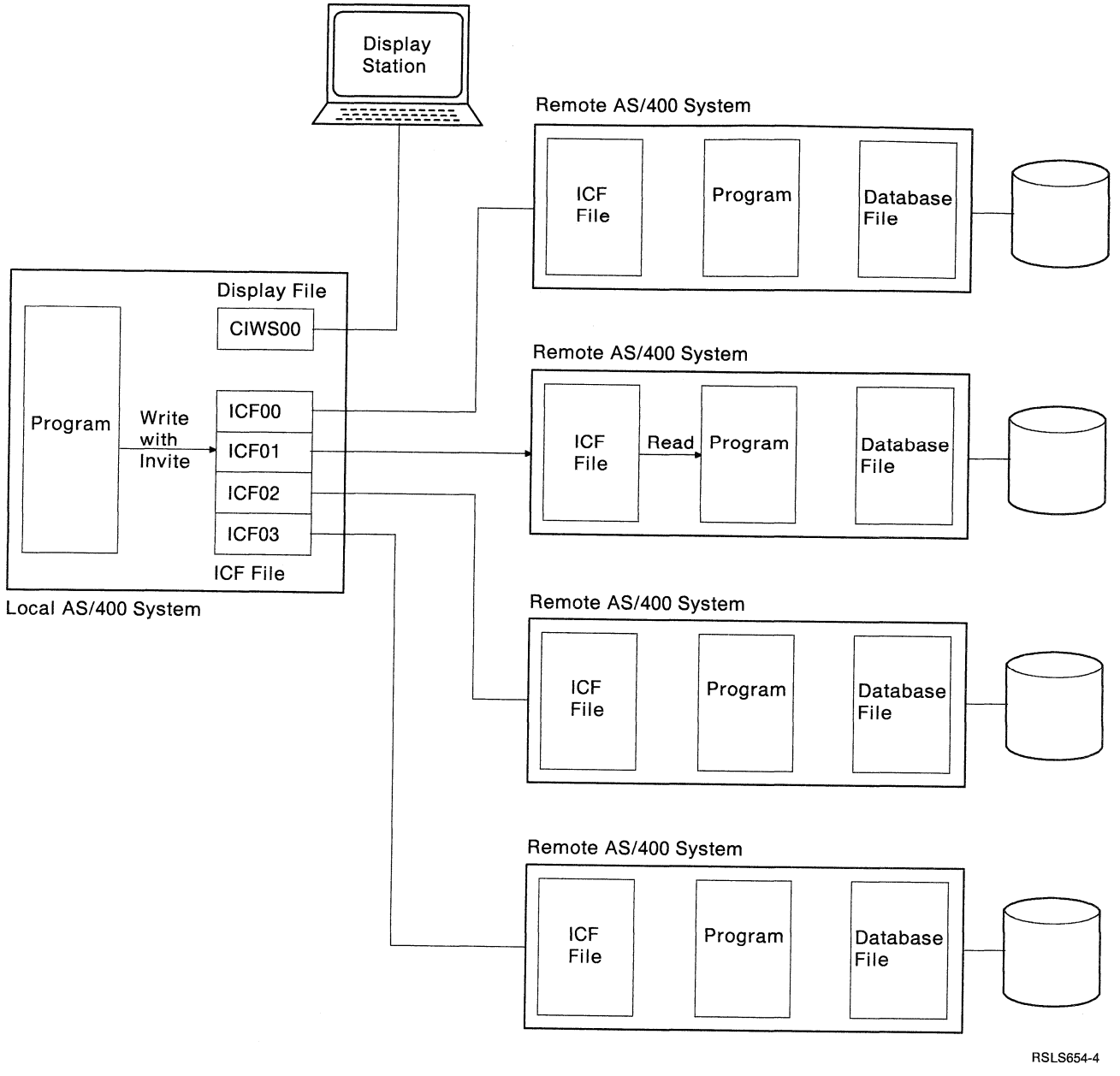


Figure 11-17. Program Sends Inquiry Request to Remote System

The target program responds to the inquiry by sending a reply, as shown in Figure 11-18.

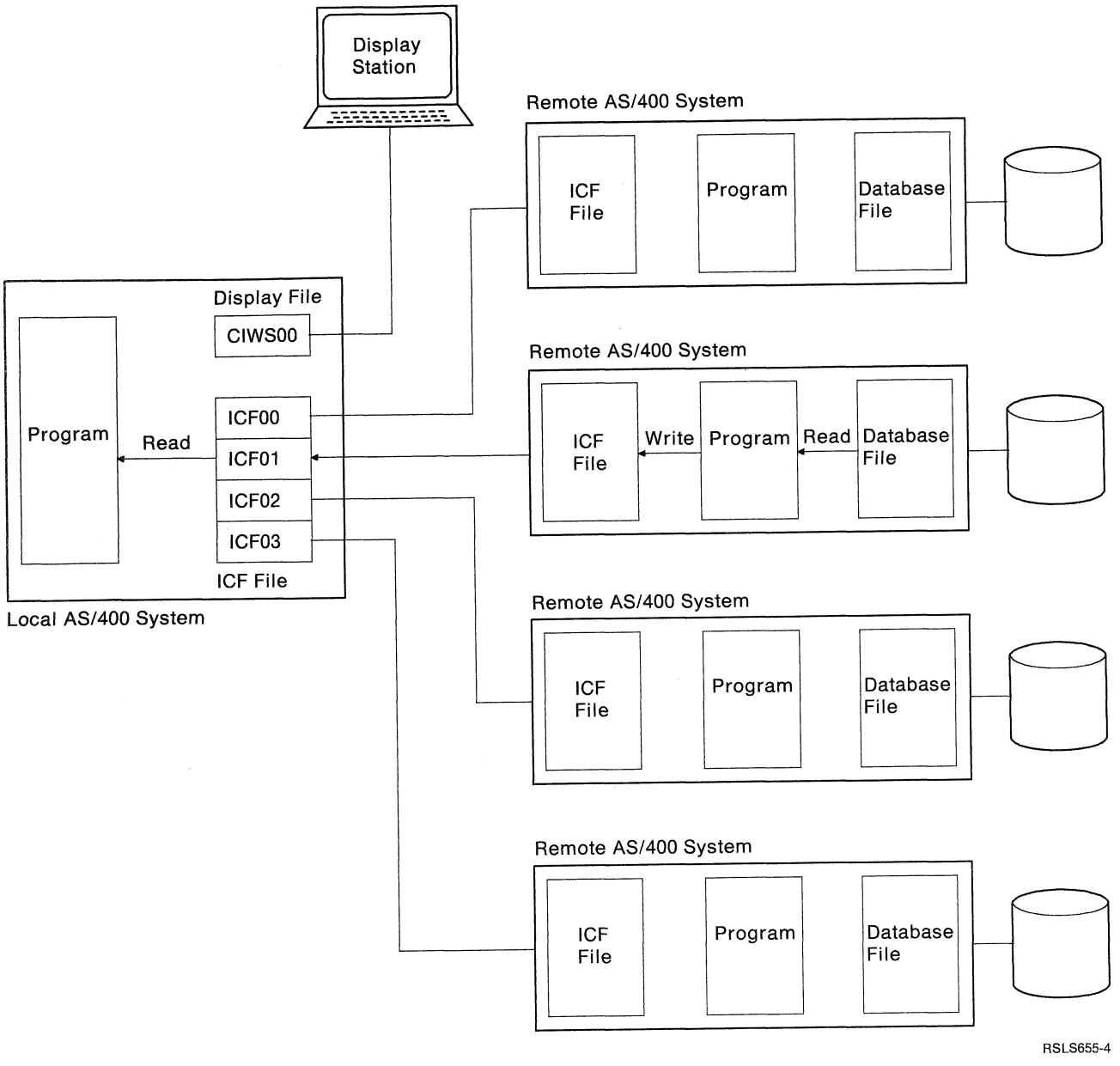


Figure 11-18. Target Program Sends a Reply

The program sends a detach request and ends the session when command function key 1 is pressed (while the main inquiry menu is present), as shown in Figure 11-19.

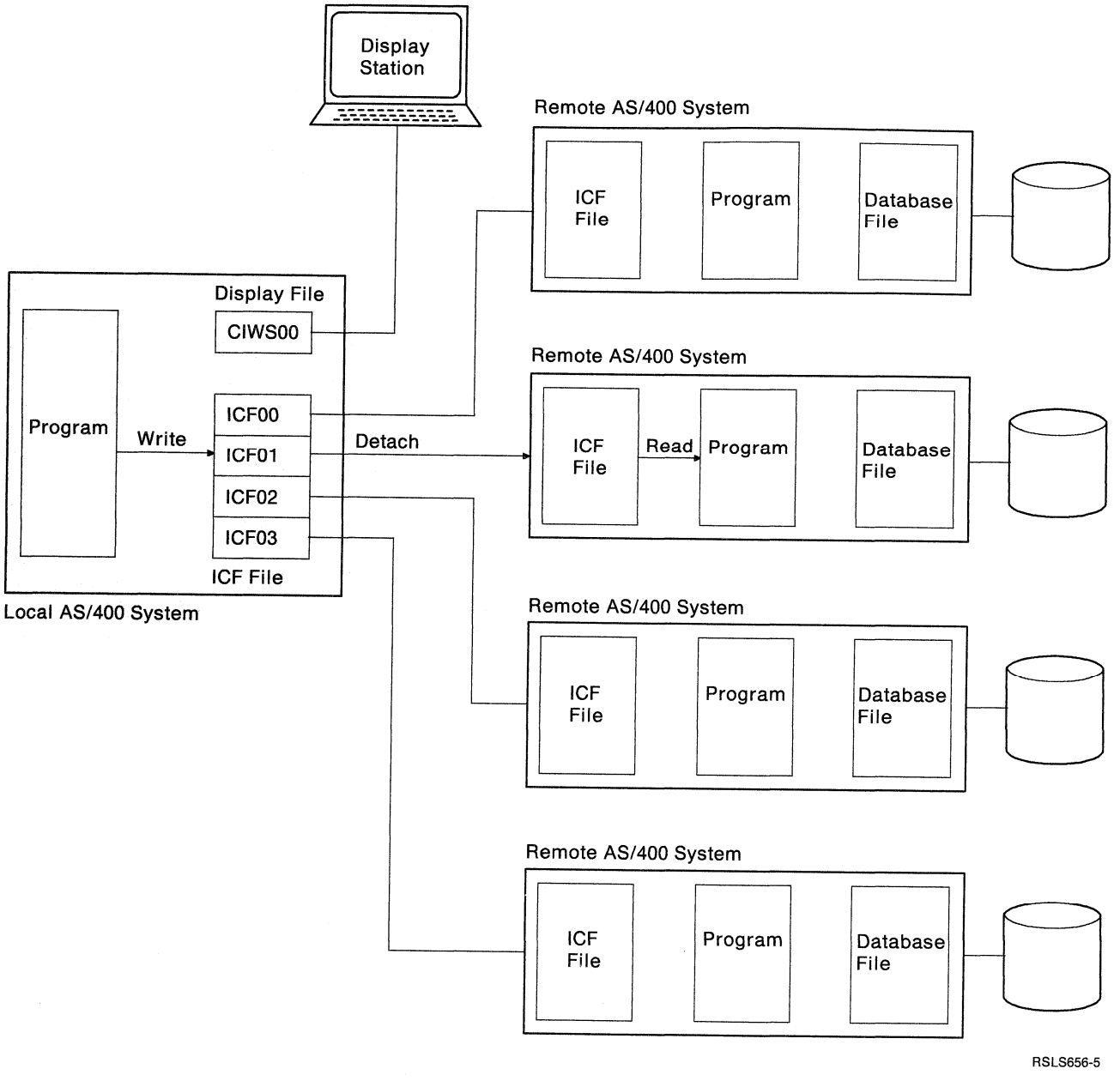


Figure 11-19. Program Ends the Session

Source Program Multiple-Session Inquiry (Example II)

The following describes a source program multiple-session inquiry.

Program Files: The RPG/400 multiple-session source program uses the following files:

CMNFIL	A ICF file used to send records to and receive records from the target program.
DSPFIL	A display file used to enter requests to be sent to the target program.
QPRINT	A printer file used to print error messages resulting from communications errors.

DDS Source: The DDS for the ICF file (CMNFIL) is illustrated in Figure 11-20 on page 11-42.

```
5714PW1 R01M00  880301                SEU SOURCE LISTING                    10/14/87 17:20:41          PAGE 1
SOURCE FILE . . . . . .  QICFPUB/ICFLIB
MEMBER  . . . . . . . .  CMNFIL
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
  100                                                                                                  10/06/87
  200                                                                                                  10/06/87
  300      A*****************************************************************                         10/14/87
  400      A*                                                               *                         10/14/87
  500      A*                          ICF FILE                           *                         10/14/87
  600      A*         USED IN SOURCE MULTIPLE SESSION PROGRAM               *                         10/14/87
  700      A*                                                               *                         10/14/87
  800      A*****************************************************************                         10/14/87
  900      A                                     INDARA                                                10/07/87
 1000      A          R ITMRSP                                                                         10/06/87
 1100      A                                     RECID(1 'I')                                          10/06/87
 1200      A            RECITM         1                                                             10/06/87
 1300      A            ITEMNO         6  0                                                          10/13/87
 1400      A            DESC          30                                                             10/06/87
 1500      A            QTYLST         7  0                                                          10/06/87
 1600      A            QTYOH          7  0                                                          10/06/87
 1700      A            QTYOO          7  0                                                          10/06/87
 1800      A            QTYBO          7  0                                                          10/06/87
 1900      A            UNITQ          2                                                             10/06/87
 2000      A            PR01           7  2                                                          10/06/87
 2100      A            PR05           7  0                                                          10/06/87
 2200      A            UFRT           5  2                                                          10/06/87
 2300      A            SLSTM          9  2                                                          10/06/87
 2400      A            SLSTY         11  2                                                          10/06/87
 2500      A            CSTTM          9  2                                                          10/06/87
 2600      A            CSTTY         11  2                                                          10/06/87
 2700      A            PRO            5  2                                                          10/06/87
 2800      A            LOS            9  2                                                          10/06/87
 2900      A            FILL1         56                                                             10/06/87
 3000      A          R DTLRSP                                                                         10/06/87
 3100      A                                     RECID(1 'C')                                          10/06/87
 3200      A                                     RCVTRNRND(90)                                         10/06/87
 3300      A            RECCUS         1                                                             10/06/87
 3400      A            CUSTNO         6  0                                                          10/13/87
 3500      A            DNAME         30                                                             10/06/87
 3600      A            DLSTOR         6  0                                                          10/06/87
 3700      A            DSLSTM         9  0                                                          10/06/87
 3800      A            DSPM01         9  0                                                          10/06/87
 3900      A            DSPM02         9  0                                                          10/06/87
 4000      A            DSPM03         9  0                                                          10/06/87
 4100      A            DSTTYD        11  0                                                          10/06/87
 4200      A            IDEPT          3  0                                                          10/06/87
 4300      A            FILL2         57                                                             10/06/87
 4400      A          R DETACH                                                                         10/06/87
 4500      A                                     DETACH                                                10/06/87
 4600      A          R EOS                                                                            10/06/87
 4700      A                                     EOS                                                   10/06/87
 4800      A          R EVKREQ                                                                         10/06/87
 4900      A                                     EVOKE(&LIB/&PGMID)                                    10/12/87
 5000      A            PGMID         10A  P                                                         10/06/87
 5100      A            LIB           10A  P                                                         10/06/87
 5200      A          R ITMREQ                                                                         10/06/87
 5300      A                                     INVITE                                                10/06/87
5714PW1 R01M00  880301                SEU SOURCE LISTING                    10/14/87 17:20:41          PAGE 2
SOURCE FILE . . . . . .  QICFPUB/ICFLIB
MEMBER  . . . . . . . .  CMNFIL
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
 5400      A            ITEMNO         6  0                                                          10/13/87
 5500      A          R DTLREQ                                                                         10/06/87
 5600      A                                     INVITE                                                10/06/87
 5700      A            CUSTNO         6  0                                                          10/13/87
                              * * * *  E N D  O F  S O U R C E  * * * *
```

Figure 11-20. DDS Source for Source Program Multiple Session Inquiry Using CMNFIL

The DDS source file for the display file (DSPFIL) is shown in Figure 11-21.

```
000100871007     A*****************************************************************
000200871007     A*                                                               *
000300871007     A*                          DISPLAY FILE                         *
000400871007     A*            USED IN SOURCE MULTIPLE SESSION PROGRAM            *
000500871007     A*                                                               *
000600871007     A*****************************************************************
000700871008     A*  BEGINNING MENU
000800871008     A*********************
000900871007     A                                      DSPSIZ(*DS3)
001000871007     A                                      CF01(99) CF02(98) CF03(97)
001100871007     A          R CIMENU                    TEXT('MENU FOR INQUIRY')
001200871007     A                                    1 34'INQUIRY MENU'
001300871007     A                                    3  1'Select one of the following:'
001400871007     A                                    4  3'1. Item inquiry'
001500871007     A                                    5  3'2. Customer inquiry'
001600871007     A                                   11  1'Option:'
001700871007     A            OPTION         1N  I 11  9VALUES('1' '2')
001800871008     A                                   19  5DFT('CMD KEY 1 - END ')
001900871008     A          R DTLMNU                      TEXT('CUSTOMER INQUIRY SCREEN 1')
002000871007     A                                    2  2DFT('ENTER CUSTOMER')
002100871013     A            CUSTNO         6N 0I  2 20
002200871008     A                                   19  5DFT('CMD KEY 1 - END ')
002300871008     A                                   19 23DFT(' 2 - MAIN MENU ')
002400871008     A*
002500871008     A*****************************
002600871007     A*  CUSTOMER INQUIRY SCREEN
002700871008     A*****************************
002800871007     A          R DTLSCR                      TEXT('CUSTOMER INQUIRY SCR. #2')
002900871007     A                                    1  3DFT('CUST  DPT LAST ORD   & THIS   +
003000871007     A                                         $MTH1     &MTH2     $MTH3     THIS+
003100871008     A                                         YTD NAME')
003200871008     A            CUSTN           6N     2  2
003300871007     A            DEPT            3N 0   2  9
003400871007     A            DLSTR           6N 0   2 13
003500871007     A            DSLSM           9N 0   2 22
003600871007     A            DSPM1           9N 0   2 32
003700871007     A            DSPM2           9N 0   2 42
003800871007     A            DSPM3           9N 0   2 52
003900871007     A            DSTYD          11N 0   2 62
004000890321     A            CNAME           5      2 74
004100871008     A                                   19  5DFT('CMD KEY 1 - END ')
004200871008     A                                   19 23DFT(' 2 - MAIN MENU ')
004300871007     A*
004400871008     A*************************
004500871007     A*  ITEM INQUIRY SCREEN
004600871008     A*************************
004700871007     A          R ITMMNU                      TEXT('ITEM INQUIRY SCREEN ONE')
004800871008     A                                    2  2DFT('ENTER ITEM NUMBER')
004900871013     A            ITEMNO         6N 0I  2 20
005000871008     A                                   19  5DFT('CMD KEY 1 - END ')
005100871008     A                                   19 23DFT(' 2 - MAIN MENU ')
005200871008     A*************************
005300871008     A*  ITEM DISPLAY
005400871008     A*************************
005500871007     A          R ITMSC2                      TEXT('ITEM INQUIRY SCREEN TWO') OVE+
005600871007     A                                        RLAY
005700871007     A                                    4  2DFT('DESC-')
005800871007     A            DSC            30      4  8
005900871007     A                                    5  2DFT('QUANTITY AVAILABLE')
006000871007     A            QAVAIL          7N 0   5 25
006100871007     A                                    6 11DFT('ON HAND')
006200871007     A            QTYH            7N 0   6 25
006300871007     A                                    7 11DFT('ON ORDER')
006400871007     A            QTYO            7N 0   7 25
006500871007     A                                    8 11DFT('BACK ORDER')
006600871007     A            QTYB            7N 0   8 25
006700871007     A                                    9  2DFT('UNIT OF MEASURE')
006800871007     A            UNT             2      9 30
```

Figure 11-21 (Part 1 of 2). DDS Source for Source Program Multiple-Session Inquiry Using DSPFIL

```
006900871007     A                                      10  2DFT('PRICE PER UNIT')
007000871007     A            PR1            7Y 2  10 24EDTCDE(3)
007100871007     A                                   11  8DFT('QUANTITY')
007200871007     A            PR5            7Y 0  11 25EDTCDE(3)
007300871007     A                                   12  8DFT('FREIGHT')
007400871007     A            UFR            5Y 2  12 26EDTCDE(3)
007500871008     A                                   13 32DFT('MORE... ')
007600871008     A                                   19  5DFT('CMD KEY 1 - END ')
007700871008     A                                   19 23DFT(' 2 - MAIN MENU ')
007800871008     A                                   19 40DFT(' 3 - ITEM MENU ')
007900871008     A*****************************
008000871008     A*  ITEM ADDITIONAL DISPLAY
008100871008     A*****************************
008200871007     A          R ITMSC3                      TEXT('ITEM INQUIRY SCREEN 3  ') OVE+
008300871007     A                                        RLAY
008400871007     A                                    5  2DFT('SALES MONTH')
008500871007     A            SLSM           9Y 2   5 16EDTCDE(1)
008600871007     A                                    6  8DFT('Y-T-D')
008700871007     A            SLSY          11Y 2   6 14EDTCDE(1)
008800871007     A                                    7  2DFT('COSTS MONTH')
008900871007     A            CSTM           9Y 2   7 16EDTCDE(1)
009000871007     A                                    8  8DFT('Y-T-D')
009100871007     A            CSTY          11Y 2   8 14EDTCDE(1)
009200871007     A                                    9  2DFT('PROFIT PCT')
009300871007     A            PROFIT         5Y 2   9 22EDTCDE(1)
009400871007     A                                   10  2DFT('LOST SALES')
009500871007     A            LOSTS          9Y 2  10 16EDTCDE(1)
009600871008     A                                   19  5DFT('CMD KEY 1 - END ')
009700871008     A                                   19 23DFT(' 2 - MAIN MENU ')
009800871008     A*************************
009900871007     A*   TIMOUT SCREEN.
010000871008     A*************************
010100871007     A          R TIMOUT                      TEXT('TIME OUT SCREEN')         OVE+
010200871007     A                                        RLAY
010300871007     A                                   20  2DFT('REMOTE SYSTEM TIMED OUT. ENTER
010400871007                                              1 TO TRY AGAIN OR 2 TO END.')
010500871007     A            TIMRSP         1   I 20 61
```

Figure 11-21 (Part 2 of 2). DDS Source for Source Program Multiple-Session Inquiry Using DSPFIL

ICF File Creation and Program Device Entry Definition: The command needed to create the ICF file is:

```
CRTICFF  FILE(ICFLIB/CMNFIL)  SRCFILE(ICFLIB/QICFPUB)  SRCMBR(CMNFIL)
   ACQPGMDEV(*NONE)  MAXPGMDEV(4)  WAITRCD(30)
   TEXT("SOURCE ICF FILE FOR MULTIPLE SESSION PROGRAM")
```

The commands needed to define the four program device entries are:

```
OVRICFDEVE PGMDEV(ICF00) RMTLOCNAME(CHICAGO) FMTSLT(*RECID)

OVRICFDEVE PGMDEV(ICF01) RMTLOCNAME(NEWYORK) FMTSLT(*RECID)

OVRICFDEVE PGMDEV(ICF02) RMTLOCNAME(DETROIT) FMTSLT(*RECID)

OVRICFDEVE PGMDEV(ICF03) RMTLOCNAME(MADISON) FMTSLT(*RECID)
```

Program Explanation: The following explains the structure of the program examples illustrated in Figure 11-22 on page 11-49 and in Figure 11-23 on page 11-63. The ICF file used in the first example is defined by the user, and uses externally described data formats (DDS). The second example uses the same file, but uses program-described data and system-supplied formats. The reference numbers in the explanation below correspond to the numbers in the following program examples.

Although the basic structure of the two examples provided is the same, there are differences because of the way the user-defined formats and the system-supplied formats are used. All output operations to the ICF file in the first example are done using the WRITE statement. All output operations in the ICF file in the second example using system-supplied formats are done using the EXCPT statement.

Differences between the first and second example are described as notes in each of the following descriptions where necessary.

1 The file specifications define the ICF file (CMNFIL) and the display file (DSPFIL) used in the program.

CMNFIL is the ICF file used to send records to and receive records from each of the four target programs.

DSPFIL is the display file used to receive user's requests and to report the information received based on the request.

The files used in the program are opened at the beginning of the RPG/400 cycle.

Note: In the program using system-supplied formats, the input records for CMNFIL are explicitly coded in the program since CMNFIL is now treated as a program-described file. The system-supplied file, QICDMF, can be used instead of CMNFIL. To use QICDMF, specify QICDMF in the file specification, or use an OVRICFF command to change the file name from CMNFIL to QICDMF.

The continuation lines on the file specification define the following:

- The data structure names, IOFB and IODS, used for the feedback area (INFDS) for CMNFIL and DSPFIL respectively.
- The number of program devices that can be attached to the files (four for CMNFIL).

- The program device name in *CMID* field to which it issues the I/O operation.

2 The file information data structure (IOFB) is provided to receive the I/O feedback area following an ICF file I/O operation.

For the display file, the file information data structure (IODS) is used by the program to determine the record format used for the last display file I/O operation. The field name referenced in the program is *RECID*, found in positions 261 through 268 of the feedback area.

3 The four ICF program devices used by the program are explicitly acquired.

The work station is implicitly acquired when the DSPFIL file opens.

Also, the evoke requests are issued to the the remote systems by the subroutine at **13**.

When control returns from **13**, the main menu (record format CIMENU) is written to the work station.

4 A read operation is issued to the display program device and the program waits for an input request from the user. When a record is returned, the last record format used (as specified in the *RECID* field in the I/O feedback area) is checked. The program branches to the appropriate routine according to the value in RECID.

5 The request entered by the user from the main menu (CIMENU) is checked. If indicator 99 is set to 1, indicating that the operator pressed function key 1, the four transactions and sessions end and the program ends. If the operator entered option 1, the program writes the item inquiry menu (ITMMNU) to the work station and returns to **4**.

If the option is not 1, the customer inquiry menu (DTLMNU) is written to the work station and control is passed to **4**.

6 The item number requested by the user from the Item Inquiry Screen (record format ITMMNU) is processed here. If function key 1 is pressed (indicator 99), control passes to **12**, the four transactions and sessions are ended, and the program ends. If function key 2 is pressed, the inquiry request is canceled, the main menu (CIMENU) is written to the work station, and the program returns to **4**.

The item number read from the work station is checked for value range. If the range is from 0 to 399999, then the request is sent to the target program on program device ICF01.

If the range is from 400000 to 699999, then the request is sent to the target program on program device ICF02.

If the range is from 700000 to 899999, then the request is sent to the target program on program device ICF03.

The request is sent to the appropriate target program by writing data to the program device using format ITMREQ. The INVITE keyword is specified as part of the ITMREQ format to give the target program permission to send.

A read-from-invited-program-devices operation is issued to the invited program device to receive the response to the inquiry. The operation is interpreted as a read-from-invited-program-devices because the program device name field (*CMID*) is blank. Indicator 89 is set on after I/O operation, if the operation does not complete. Subroutine **14** gets control, and further checks are made.

The return codes are checked after every I/O request. If there are any errors, control is passed to 12.

The program returns to 4.

Note: In the program using system-supplied formats, the $$SEND format is used instead of the user-defined ITMREQ format. Also, the EXCPT statement is used instead of the WRITE statement.

7 The information received from the target program is processed. If the returned item number is 0 or less, the request is invalid, a new item inquiry menu (ITMMNU) is written to the work station, and control goes to 4.

The program then performs the calculations to set the quantity fields and writes the result to the requesting work station using record format ITMSC2.

The program then returns to 4.

8 This section processes the user requests for additional information (record format ITMSC2). If function key 2 (indicator 98) was pressed, the main menu (record format CIMENU) writes to the work station and control goes to 4.

If function key 2 was pressed (as indicated by indicator 98), the profit and loss figures are calculated. Those values are then written to the work station using format ITMSC3 (item inquiry work station 3). The program then returns to 4. If function key 1 (indicator 99) was pressed, control goes to 12.

If function key 3 (indicator 97) was pressed, the Item Inquiry Menu (ITMMNU) is written to the work station and the program returns to 4.

9 This section processes requests read from the customer menu (DTLMNU). If function key 2 (indicator 98) was pressed, the main menu (CIMENU) is written to the work station and the program returns to 4. If function key 1 (indicator 99) was pressed, control goes to 12.

The customer inquiry request is send to the target program by writing data to the program device (ICF00) using format DTLREQ. The INVITE keyword is specified as part of the DLTREQ format to give the target program permission to send.

A read operation is issued to the invited program device to receive the response to the inquiry. This is accomplished by blanking out *CMID*. Indicator 88 is set on if the I/O operation did not complete.

The return codes are checked after every I/O request. If there are any errors, control is passed to 12.

Note: In the program using system-supplied formats, the $$SEND format is used instead of the user-defined DTLREQ format. Also, the EXCPT operation is used instead of the WRITE operation. The READ operation is issued using ICF file CMNFIL in factor 2.

10 The information supplied by the target program in response to a request for a customer detail is processed. If the customer number is 0 or less, the request is invalid and the main menu (record format CIMENU) is written to the work station. The program then returns to 4.

The detail information is written to the work station using record format DTLSCR.

The program then returns to 4.

The return codes are checked after every I/O request to verify the success of the operation.

Note: The READ operation is issued using file name CMNFIL in factor 2 in the program using system-supplied formats.

11 Control is passed here if the customer detail record format (DLTSCR) is displayed. If function key 1 (indicator 99) was pressed, control goes to **12**. If function key 2 (indicator 98) was pressed, the main menu (CIMENU) is written to the work station and control is returned to **4**.

12 If the record format name is not found on a read operation, an error message prints. If an error occurs on any ICF operation, control is passed here and an error message is printed containing the program device and error that occurred.

For each of the four sessions, the transaction is ended by issuing a detach request to the appropriate program device using format DETACH, and the session is ended by the release operation. The last record indicator is turned on to end the program. The ICF file is implicitly closed at the end of the RPG/400 cycle.

13 This subroutine builds the evoke requests to send to the remote systems. Because the DDS keyword for the record format only specifies the field identifiers with the record, this code moves the literal value RTDMULCL to the field *PGMID*, and ICFLIB to the field *LIB*.

When the program start request is received at the remote system, ICFLIB is searched for RTDMULCL and that program is then started. RTDMULCL is a CL program that contains the following:

```
ADDLIBLE ICFLIB
CALL ICFLIB/RTDMUL
```

Note: In the program using system-supplied formats, the library and program (ICFLIB/RTFMULCL) are specified as part of the $$EVOKNI format. RTFMULCL is a CL program that contains the following:

```
ADDLIBLE ICFLIB
CALL ICFLIB/RTFMUL
```

14 This subroutine is called when the read operation to the program device does not complete. The indication that the timer has ended is checked (RC = 0310), and, if it is set, a message displays to the user. The message asks whether to try the read operation again, or to end the job. In this example, the time interval is specified at **10**.

15 This subroutine is called for I/O operation errors that are not handled by subroutine **14**. It checks whether the program device is already acquired when an acquire operation is requested, and, if it is, the second acquire is ignored. Otherwise the program ends.

```
5728RG1 R01M02  881028            IBM AS/400 RPG/400                 ICFLIB/RSDMUL        03/20/89  15:44:17      Page        1
Compiler . . . . . :   IBM AS/400 RPG/400
Command Options:
  Program  . . . . . :   ICFLIB/RSDMUL
  Source file  . . . . . :   ICFLIB/QICFPUB
  Source member  . . . . . :   *PGM
  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
  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 . . . . . :   RSDMUL
  File . . . . . :   QICFPUB
  Library  . . . . . :   ICFLIB
  Last Change  . . . . . :   03/20/89  15:34:43
  Description  . . . . . :   RPG Multi-Session example w/DDS (source)
5728RG1 R01M02  881028            IBM AS/400 RPG/400                 ICFLIB/RSDMUL        03/20/89  15:44:17      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  H*****************************************************************          10/13/87
     200  H*                                                               *          03/20/89
     300  H*  THIS PROGRAM ASSIGNS FOUR SESSIONS AS FOLLOWS:               *          10/13/87
     400  H*    'ICF00' TO INQUIRE ABOUT A CUSTOMER ACCOUNT BEFORE AN      *          10/13/87
     500  H*            ORDER IS PROCESSED.                                *          10/13/87
     600  H*    'ICF01' TO INQUIRE ABOUT THE INVENTORY STATUS OF AN ITEM   *          10/13/87
     700  H*            BEING ORDERED (ITEM 000001 THRU 399999).           *          10/13/87
     800  H*    'ICF02' TO INQUIRE ABOUT THE INVENTORY STATUS OF AN ITEM   *          10/13/87
     900  H*            BEING ORDERED (ITEM 400000 THRU 699999).           *          10/13/87
    1000  H*    'ICF03' TO INQUIRE ABOUT THE INVENTORY STATUS OF AN ITEM   *          10/13/87
    1100  H*            BEING ORDERED (ITEM 700000 THRU 999999).           *          10/13/87
    1200  H*  A DISPLAY DEVICE IS USED TO ENTER THE REQUEST (USING A       *          03/20/89
    1300  H*  CUSTOMER AND AN ITEM MENU) THAT IS SENT TO THE REMOTE        *          10/13/87
    1400  H*  SYSTEM.                                                      *          10/16/87
    1500  H*                                                               *          03/20/89
    1600  H*****************************************************************          10/13/87
    1700  F*****************************************************************          10/13/87
    1800  F*                                                                         10/13/87
    1900  F*                F I L E   S P E C I F I C A T I O N S                    10/13/87
    2000  F*                                                                         10/13/87
    2100  F*    CMNFIL :   ICF FILE USED TO SEND A REQUEST TO ONE                    01/18/88
    2200  F*               OF FOUR DIFFERENT TARGET PROGRAMS.  MULTIPLE             10/13/87
    2300  F*               SESSIONS ARE ACTIVE AT THE SAME TIME.                    10/13/87
    2400  F*                                                                       10/13/87
    2500  F*    DSPFIL :   DISPLAY FILE USED TO ENTER A REQUEST TO BE              10/13/87
    2600  F*               SENT TO A REMOTE SYSTEM.                                10/13/87
    2700  F*                                                                       10/13/87
    2800  F*    THE FOLLOWING INFORMATION IS SPECIFIED AS PART OF THE              10/13/87
    2900  F*    FILE SPECIFICATION:                                                10/13/87
    3000  F*                INFDS : I/O FEEDBACK AREA                            10/13/87
    3100  F*                NUM   : SPECIFIES THE MAXIMUM NUMBER OF              10/13/87
    3200  F*                        PROGRAM DEVICES THAT CAN BE ATTACHED         10/13/87
    3300  F*                        TO THIS FILE.  A VALUE OF 4 IS               10/13/87
    3400  F*                        SPECIFIED FOR THE ICF FILE.                  01/18/88
    3500  F*                        THIS DEFINES THE FILE AS A                   10/13/87
    3600  F*                        MULTIPLE DEVICE FILE.                        10/13/87
    3700  F*                ID    : 10 CHARACTER PROGRAM DEVICE NAME             10/13/87
    3800  F*                        FIELD WHICH SPECIFIES WHICH PROGRAM          10/13/87
```

Figure 11-22 (Part 1 of 14). Source Program Example — RSDMUL (User-Defined Formats)

```
   3900 F*                       DEVICE TO DIRECT THE OPERATION.                    10/13/87
   4000 F*                                                                          10/13/87
   4100 F*                                                                          10/13/87
   4200 F*****************************************************************          10/13/87
   4300 H* 1                                                                        10/13/87
   4400 FCMNFIL  CF  E                    WORKSTN                                   10/13/87
   4500 F                                            KINFDS IOFB                   10/13/87
   4600 F                                            KINFSR *PSSR                  10/14/87
   4700 F                                            KNUM       4                  10/13/87
   4800 F                                            KID    CMID                   10/13/87
5728RG1 R01M02  881028            IBM AS/400 RPG/400                 ICFLIB/RSDMUL      03/20/89  15:44:17      Page        3
SEQUENCE                                                                    IND   DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE   NUM   UPDATE     LINE   ID
          RECORD FORMAT(S):  LIBRARY ICFLIB FILE CMNFIL.
                   EXTERNAL FORMAT ITMRSP RPG NAME ITMRSP
                   EXTERNAL FORMAT DTLRSP RPG NAME DTLRSP
                   EXTERNAL FORMAT DETACH RPG NAME DETACH
                   EXTERNAL FORMAT EOS RPG NAME EOS
                   EXTERNAL FORMAT EVKREQ RPG NAME EVKREQ
                   EXTERNAL FORMAT ITMREQ RPG NAME ITMREQ
                   EXTERNAL FORMAT DTLREQ RPG NAME DTLREQ
   4900 FDSPFIL  CF  E                    WORKSTN                                   10/13/87
   5000 F                                            KINFDS IODS                   10/13/87
          RECORD FORMAT(S):  LIBRARY ICFLIB FILE DSPFIL.
                   EXTERNAL FORMAT CIMENU RPG NAME CIMENU
                   EXTERNAL FORMAT DTLMNU RPG NAME DTLMNU
                   EXTERNAL FORMAT DTLSCR RPG NAME DTLSCR
                   EXTERNAL FORMAT ITMMNU RPG NAME ITMMNU
                   EXTERNAL FORMAT ITMSC2 RPG NAME ITMSC2
                   EXTERNAL FORMAT ITMSC3 RPG NAME ITMSC3
                   EXTERNAL FORMAT TIMOUT RPG NAME TIMOUT
   5100 FQPRINT  O   F     132            PRINTER                                   10/13/87
   5200 I*****************************************************************          10/13/87
   5300 I*                                                                          10/13/87
   5400 I*            I N P U T   S P E C I F I C A T I O N S                       10/13/87
   5500 I*                                                                          10/13/87
   5600 I*  IODS    :  REDEFINES THE I/O FEEDBACK AREA OF THE DISPLAY                10/13/87
   5700 I*              FILE.  THIS AREA CONTAINS THE NAME OF THE LAST              10/13/87
   5800 I*              RECORD PROCESSED.  THIS FIELD IS CALLED RECID.              10/13/87
   5900 I*  IOFB    :  REDEFINES THE I/O FEEDBACK AREA FOR THE ICF                  01/18/88
   6000 I*              FILE.                                                       10/13/87
   6100 I*                                                                          10/13/87
   6200 I*****************************************************************          10/13/87
   6300 I*                                                                          10/13/87
 A000000   INPUT  FIELDS FOR RECORD ITMRSP FILE CMNFIL FORMAT ITMRSP.
 A000001                                       1   1 RECITM
 A000002                                       2   70ITEMNO
 A000003                                       8  37 DESC
 A000004                                      38  440QTYLST
 A000005                                      45  510QTYOH
 A000006                                      52  580QTYOO
 A000007                                      59  650QTYBO
 A000008                                      66  67 UNITQ
 A000009                                      68  742PR01
 A000010                                      75  810PR05
 A000011                                      82  862UFRT
 A000012                                      87  952SLSTM
 A000013                                      96 1062SLSTY
 A000014                                     107 1152CSTTM
 A000015                                     116 1262CSTTY
 A000016                                     127 1312PRO
 A000017                                     132 1402LOS
 A000018                                     141 196 FILL1
```

Figure 11-22 (Part 2 of 14). Source Program Example — RSDMUL (User-Defined Formats)

```
5728RG1 R01M02  881028            IBM AS/400 RPG/400                 ICFLIB/RSDMUL        03/20/89  15:44:17      Page       4
SEQUENCE                                                                                    IND    DO    LAST        PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE      LINE   ID
 B000000   INPUT  FIELDS FOR RECORD DTLRSP FILE CMNFIL FORMAT DTLRSP.
 B000001                                        1   1 RECCUS
 B000002                                        2   70CUSTNO
 B000003                                        8  37 DNAME
 B000004                                       38  430DLSTOR
 B000005                                       44  520DSLSTM
 B000006                                       53  610DSPM01
 B000007                                       62  700DSPM02
 B000008                                       71  790DSPM03
 B000009                                       80  900DSTTYD
 B000010                                       91  930IDEPT
 B000011                                       94 150 FILL2
 C000000   INPUT  FIELDS FOR RECORD DETACH FILE CMNFIL FORMAT DETACH.
 D000000   INPUT  FIELDS FOR RECORD EOS FILE CMNFIL FORMAT EOS.
 E000000   INPUT  FIELDS FOR RECORD EVKREQ FILE CMNFIL FORMAT EVKREQ.
 F000000   INPUT  FIELDS FOR RECORD ITMREQ FILE CMNFIL FORMAT ITMREQ.
 F000001                                        1   60ITEMNO
 G000000   INPUT  FIELDS FOR RECORD DTLREQ FILE CMNFIL FORMAT DTLREQ.
 G000001                                        1   60CUSTNO
 H000000   INPUT  FIELDS FOR RECORD CIMENU FILE DSPFIL FORMAT CIMENU.
 H000000        MENU FOR INQUIRY
 H000001                                        3   3 *IN97
 H000002                                        2   2 *IN98
 H000003                                        1   1 *IN99
 H000004                                        4   4 OPTION
 I000000   INPUT  FIELDS FOR RECORD DTLMNU FILE DSPFIL FORMAT DTLMNU.
 I000000        CUSTOMER INQUIRY SCREEN 1
 I000001                                        3   3 *IN97
 I000002                                        2   2 *IN98
 I000003                                        1   1 *IN99
 I000004                                        4   90CUSTNO
 J000000   INPUT  FIELDS FOR RECORD DTLSCR FILE DSPFIL FORMAT DTLSCR.
 J000000        CUSTOMER INQUIRY SCR. #2
 J000001                                        3   3 *IN97
 J000002                                        2   2 *IN98
 J000003                                        1   1 *IN99
 K000000   INPUT  FIELDS FOR RECORD ITMMNU FILE DSPFIL FORMAT ITMMNU.
 K000000        ITEM INQUIRY SCREEN ONE
 K000001                                        3   3 *IN97
 K000002                                        2   2 *IN98
 K000003                                        1   1 *IN99
 K000004                                        4   90ITEMNO
 L000000   INPUT  FIELDS FOR RECORD ITMSC2 FILE DSPFIL FORMAT ITMSC2.
 L000000        ITEM INQUIRY SCREEN TWO
 L000001                                        3   3 *IN97
 L000002                                        2   2 *IN98
 L000003                                        1   1 *IN99
 M000000   INPUT  FIELDS FOR RECORD ITMSC3 FILE DSPFIL FORMAT ITMSC3.
 M000000        ITEM INQUIRY SCREEN 3
 M000001                                        3   3 *IN97
 M000002                                        2   2 *IN98
 M000003                                        1   1 *IN99
 N000000   INPUT  FIELDS FOR RECORD TIMOUT FILE DSPFIL FORMAT TIMOUT.
 N000000        TIME OUT SCREEN
5728RG1 R01M02  881028            IBM AS/400 RPG/400                 ICFLIB/RSDMUL        03/20/89  15:44:17      Page       5
SEQUENCE                                                                                    IND    DO    LAST        PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE      LINE   ID
 N000001                                        3   3 *IN97
 N000002                                        2   2 *IN98
 N000003                                        1   1 *IN99
 N000004                                        4   4 TIMRSP
    6400  IIODS 2        DS                                                                    10/13/87
    6500  I                                     1 240 FILL01                                  10/13/87
    6600  I                                   261 268 RECID                                   10/13/87
    6700  I                                   271 415 FILL02                                  10/13/87
```

Figure 11-22 (Part 3 of 14). Source Program Example — RSDMUL (User-Defined Formats)

```
   6800 IIOFB       DS                                                      10/13/87
   6900 I                                    *ROUTINE LOC                   10/14/87
   7000 I                                    *STATUS  ERR                   10/14/87
   7100 I                                       1 240 FILL03                10/13/87
   7200 I                                      38  47 FMTNM                 10/13/87
   7300 I                                     273 282 CMID                  10/13/87
   7400 I                                     401 404 MAJMIN                10/13/87
   7500 I                                     401 402 MAJCOD                10/13/87
   7600 I                                     403 404 MINCOD                10/13/87
   7700 I                                     261 268 RECID2                10/13/87
   7800 I                                     271 415 FILL04                10/13/87
   7900 C*************************************************************      10/13/87
   8000 C*                                                                  10/13/87
   8100 C*       C A L C U L A T I O N   S P E C I F I C A T I O N S        10/13/87
   8200 C*                                                                  10/13/87
   8300 C*   THE DISPLAY PROGRAM DEVICE IS IMPLICITLY ACQUIRED WHEN THE     10/13/87
   8400 C*   FILE IS OPENED.                                                10/13/87
   8500 C*                                                                  10/13/87
   8600 C*   ALL OF THE ICF PROGRAM DEVICES ARE EXPLICITLY ACQUIRED.        01/18/88
   8700 C*                                                                  10/13/87
   8800 C*   EACH OF THE FOUR TARGET PROGRAMS ARE EVOKED TO ESTABLISH       10/13/87
   8900 C*   TRANSACTIONS WITH THE REMOTE SYSTEMS.                          10/13/87
   9000 C*                                                                  10/13/87
   9100 C*   THE MAIN INQUIRY MENU (CIMENU) IS WRITTEN TO THE USER'S        10/13/87
   9200 C*   DISPLAY.                                                       10/13/87
   9300 C*                                                                  10/13/87
   9400 C*************************************************************      10/13/87
   9500 C* 3                                                               10/13/87
   9600 C           ENTRY     TAG                                           10/13/87
   9700 C           'ICF00    'ACQ  CMNFIL                  1ST SESSION     10/13/87
   9800 C           'ICF01    'ACQ  CMNFIL                  2ND SESSION     10/13/87
   9900 C           'ICF02    'ACQ  CMNFIL                  3RD SESSION     10/13/87
  10000 C           'ICF03    'ACQ  CMNFIL                  4TH SESSION     10/13/87
  10100 C                     MOVEL'ICF00   'CMID           1ST PROGRAM     10/13/87
  10200 C                     EXSR EVKSR                    CALL EVOKE      10/13/87
  10300 C                     MOVEL'ICF01   'CMID           2ND PROGRAM     10/13/87
  10400 C                     EXSR EVKSR                    CALL EVOKE      10/13/87
  10500 C                     MOVEL'ICF02   'CMID           3RD PROGRAM     10/13/87
  10600 C                     EXSR EVKSR                    CALL EVOKE      10/13/87
  10700 C                     MOVEL'ICF03   'CMID           4TH PROGRAM     10/13/87
  10800 C                     EXSR EVKSR                    CALL EVOKE      10/13/87
  10900 C           MAIN      TAG                                           10/13/87
  11000 C                     WRITECIMENU                                   10/13/87
  11100 C*************************************************************      10/13/87
5728RG1 R01M02  881028            IBM AS/400 RPG/400              ICFLIB/RSDMUL       03/20/89  15:44:17      Page       6
SEQUENCE                                                                  IND    DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE     LINE   ID
  11200 C*                                                                                10/13/87
  11300 C*                    DETERMINE USER'S REQUEST                                    10/13/87
  11400 C*                                                                                10/13/87
  11500 C*   A READ TO THE DISPLAY DEVICE IS ISSUED TO RECEIVE THE USER'S                  03/20/89
  11600 C*   REQUEST. THE TYPE OF REQUEST MADE IS BASED ON THE DISPLAY                    03/20/89
  11700 C*   FORMAT CURRENTLY ON THE SCREEN.  THE RECORD FORMAT NAME IS                   03/20/89
  11800 C*   EXTRACTED FROM THE I/O FEEDBACK AREA AND IS USED TO DETER-                   03/20/89
  11900 C*   MINE WHAT ACTION SHOULD BE TAKEN NEXT.                                       03/20/89
  12000 C*                                                                                10/13/87
  12100 C*************************************************************                  10/13/87
  12200 C* 4                                                                            10/13/87
  12300 C           READRQ    TAG                                                       10/13/87
  12400 C                     SETOF                    8889  TIMEOUT IND     1 2        10/13/87
  12500 C                     READ DSPFIL                 87                    3        10/13/87
  12600 C           RECID     CABEQ'CIMENU  'MENU             MAIN MENU?              03/20/89
  12700 C           RECID     CABEQ'ITMMNU  'ITMIN            ITEM MENU?              03/20/89
  12800 C           RECID     CABEQ'ITMSC2  'ITMRTN           ITM SCR?                10/13/87
  12900 C           RECID     CABEQ'ITMSC3  'ITMRTN           ITM SCR?                10/13/87
  13000 C           RECID     CABEQ'DTLMNU  'DTLIN            DETAIL SCR?             10/13/87
  13100 C           RECID     CABEQ'DTLSCR  'DTLRTN           CUST SCR?               10/13/87
  13200 C                     WRITECIMENU                     MAIN MENU IF            10/13/87
  13300 C                     GOTO READRQ                     THERE IS ERR            10/13/87
```

Figure 11-22 (Part 4 of 14). Source Program Example — RSDMUL (User-Defined Formats)

```
13400 C*****************************************************************      10/13/87
13500 C*                                                                      10/13/87
13600 C*                            MAIN MENU                                 10/13/87
13700 C*                                                                      10/13/87
13800 C*    THE MAIN MENU IS READ TO DETERMINE THE REQUEST ENTERED            10/13/87
13900 C*    BY THE USER. IF CMD 1 (*IN99) IS PRESSED, THE PROGRAM             10/13/87
14000 C*    IS ENDED.  IF OPTION = 1, AN ITEM INQUIRY MENU IS WRITTEN         10/13/87
14100 C*    TO SCREEN.  IF OPTION = 2, A CUSTOMER INQUIRY MENU IS             10/13/87
14200 C*    WRITTEN TO THE SCREEN.                                            10/13/87
14300 C*                                                                      10/13/87
14400 C*****************************************************************      10/13/87
14500 C* 5                                                                    10/13/87
14600 C          MENU      TAG                                               10/13/87
14700 C          *IN99     CABEQ'1'       END             JOB ENDS           10/13/87
14800 C          OPTION    IFEQ '1'                                  B001    10/13/87
14900 C                    WRITEITMMNU                    ITEM MENU   001    10/13/87
15000 C                    ELSE                                      X001    10/13/87
15100 C                    WRITEDTLMNU                    CUST MENU   001    10/13/87
15200 C                    END                                       E001    10/13/87
15300 C                    GOTO READRQ                                       10/13/87
15400 C*****************************************************************      10/13/87
15500 C*                                                                      10/13/87
15600 C*                       ITEM INQUIRY                                   10/13/87
15700 C*                                                                      10/13/87
15800 C*    THE ITEM NUMBER REQUESTED BY THE USER ON THE ITEM INQUIRY         10/13/87
15900 C*    SCREEN IS CHECKED.  THIS IS DETERMINED BY THE DISPLAY             03/20/89
16000 C*    RECORD FORMAT BEING PROCESSED - IN THIS CASE ITMMNU.              03/20/89
16100 C*                                                                      10/13/87
16200 C*    IF CMD 1 (*IN99) IS PRESSED, THE PROGRAM IS ENDED.  IF CMD 2      10/13/87
16300 C*    IS PRESSED, THE ITEM INQUIRY REQUEST IS CANCELED, AND THE         03/20/89
16400 C*    MAIN MENU (CIMENU) IS WRITTEN TO THE SCREEN.                      10/13/87
16500 C*                                                                      10/13/87
5728RG1 R01M02  881028            IBM AS/400 RPG/400              ICFLIB/RSDMUL      03/20/89  15:44:17      Page       7
SEQUENCE                                                              IND   DO   LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE   NUM  UPDATE     LINE   ID
16600 C*    IF AN ITEM NUMBER IS ENTERED, A ITEM INQUIRY REQUEST IS           10/13/87
16700 C*    SENT TO THE APPROPRIATE REMOTE SYSTEM.  THE REMOTE SYSTEM         10/13/87
16800 C*    IS SELECTED BASED ON THE ITEM NUMBER REQUESTED.                   10/13/87
16900 C*                                                                      10/13/87
17000 C*    IF AN ERROR OCCURS, THE ERROR IS PRINTED AND THE JOB              10/13/87
17100 C*    IS ENDED.                                                         10/13/87
17200 C*                                                                      10/13/87
17300 C*****************************************************************      10/13/87
17400 C* 6                                                                    10/13/87
17500 C          ITMIN     TAG                                               10/13/87
17600 C          *IN99     CABEQ'1'       END             EXIT ON CMD3       10/13/87
17700 C          *IN98     IFEQ '1'                                  B001    10/13/87
17800 C                    WRITECIMENU                    MAIN MENU   001    10/13/87
17900 C                    GOTO READRQ                               001    10/13/87
18000 C                    END                                       E001    10/13/87
18100 C          ITEMNO    CABLE399999    XICF01                             10/13/87
18200 C          ITEMNO    CABLE699999    XICF02                             10/13/87
18300 C          ITEMNO    CABLE899999    XICF03                             10/13/87
18400 C          XICF01    TAG                                               10/13/87
18500 C                    MOVEL'ICF01    'CMID                              10/13/87
18600 C                    GOTO XITMIN                                       10/13/87
18700 C          XICF02    TAG                                               10/13/87
18800 C                    MOVEL'ICF02    'CMID                              10/13/87
18900 C                    GOTO XITMIN                                       10/13/87
19000 C          XICF03    TAG                                               10/13/87
19100 C                    MOVEL'ICF03    'CMID                              10/13/87
19200 C          XITMIN    TAG                                               10/13/87
19300 C                    WRITEITMREQ                    INQ W/INVITE       10/13/87
19400 C          MAJCOD    CABGE'04'      ERROR           ERROR RTN          10/13/87
19500 C          TRY89     TAG                                               10/13/87
19600 C                    SETOF                      89                3    10/13/87          OQ
19700 C                    MOVEL'         'CMID                              10/13/87
19800 C                    READ CMNFIL              8910RECV ITM INFO   2 3    10/13/87
```

Figure 11-22 (Part 5 of 14). Source Program Example — RSDMUL (User-Defined Formats)

```
  19900  C   89              EXSR ERRCHK                       CHCK ERR INFO              10/13/87
  20000  C          MAJMIN    CABGE'0300'    ITMIN             NODATATRYAGN               10/13/87
  20100  C          MAJCOD    CABGE'04'      ERROR             ERROR RTN                  10/13/87
  20200  C          RECID2    CABNE'ITMRSP'  RECERR            PRINT MSG                  10/13/87
  20300  C*****************************************************************               10/13/87
  20400  C*                                                                               10/13/87
  20500  C*                    PROCESS ITEM INFORMATION                                   10/13/87
  20600  C*                                                                               10/13/87
  20700  C*    THE ITEM RECORD RECEIVED FROM THE TARGET PROGRAM AND THE                   10/13/87
  20800  C*    INFORMATION ABOUT THE ITEM IS PROCESSED AND DISPLAYED.                     10/13/87
  20900  C*    IF ITEMNO IS 0 OR LESS, IT IS AN INVALID REQUEST AND A FRESH               10/13/87
  21000  C*    ITEM MENU IS WRITTEN TO THE SCREEN.  IF THE REQUEST IS                     10/13/87
  21100  C*    VALID, VALUES ARE CALCULATED BASED ON THE INFORMATION                      10/13/87
  21200  C*    RECEIVED.                                                                  10/13/87
  21300  C*                                                                               10/13/87
  21400  C*****************************************************************               10/13/87
  21500  C* 7                                                                             10/13/87
  21600  C          ITMOUT    TAG                                                         10/13/87
  21700  C          ITEMNO    IFLE 000000                                          B001   10/13/87
  21800  C                    WRITEITMMNU                      ITEM MENU            001   10/13/87
  21900  C                    GOTO READRQ                      READ DISPLY          001   10/13/87
5728RG1 R01M02  881028          IBM AS/400 RPG/400                ICFLIB/RSDMUL         03/20/89  15:44:17       Page      8
SEQUENCE                                                                     IND    DO     LAST       PAGE    PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM    UPDATE     LINE    ID
  22000  C                    ELSE                                                 X001   10/13/87
  22100  C                    Z-ADD0         QAVAIL  70        QTY AVAIL.           001   10/13/87
  22200  C                    ADD  QTYOH     QAVAIL                                 001   10/13/87
  22300  C                    SUB  QTYOO     QAVAIL                                 001   10/13/87
  22400  C                    ADD  QTYBO     QAVAIL                                 001   10/13/87
  22500  C                    MOVELDESC      DSC                                    001   10/13/87
  22600  C                    MOVE QTYOO     QTYO                                   001   10/13/87
  22700  C                    MOVE QTYOH     QTYH                                   001   10/13/87
  22800  C                    MOVE QTYBO     QTYB                                   001   10/13/87
  22900  C                    MOVE UNITQ     UNT                                    001   10/13/87
  23000  C                    MOVE PR01      PR1                                    001   10/13/87
  23100  C                    MOVE PR05      PR5                                    001   10/13/87
  23200  C                    MOVE UFRT      UFR                                    001   10/13/87
  23300  C                    WRITEITMSC2                      DSP DETAIL           001   10/13/87
  23400  C                    GOTO READRQ                                           001   10/13/87
  23500  C*****************************************************************               10/13/87
  23600  C*                                                                               10/13/87
  23700  C*                   ADDITIONAL ITEM INFORMATION                                 10/13/87
  23800  C*                                                                               10/13/87
  23900  C*    ADDITIONAL ITEM INFORMATION IS PROCESSED AND THE RESULT                    10/13/87
  24000  C*    DISPLAYED ON THE SCREEN WHEN A RESPONSE IS READ FROM THE                   03/20/89
  24100  C*    DISPLAY WITH AN ITEM SCREEN RECORD FORMAT.                                 03/20/89
  24200  C*                                                                               10/13/87
  24300  C*    IF CMD 1 (*IN99) IS PRESSED, THE PROGRAM IS ENDED.                         10/13/87
  24400  C*                                                                               10/13/87
  24500  C*    IF CMD 2 (*IN98) IS PRESSED, THE ITEM INQUIRY IS ENDED, AND                03/20/89
  24600  C*    THE MAIN MENU (CIMENU) IS WRITTEN TO THE SCREEN.                           03/20/89
  24700  C*                                                                               10/13/87
  24800  C*    IF CMD 3 (*IN97) IS PRESSED, THE ITEM INQUIRY MENU IS                      10/14/87
  24900  C*    WRITTEN ON THE SCREEN.                                                     10/14/87
  25000  C*                                                                               10/14/87
  25100  C*    IF 'ENTER' IS PRESSED WHILE SCREEN 2 FOR ITEM REQUESTED IS                 10/16/87
  25200  C*    CURRENTLY DISPLAYED, MORE INFORMATION IS CALCULATED AND                    10/16/87
  25300  C*    DISPLAYED.                                                                 10/16/87
  25400  C*                                                                               10/16/87
  25500  C*    IF 'ENTER' IS PRESSED WHILE SCREEN 3 FOR ITEM REQUESTED IS                 10/16/87
  25600  C*    CURRENTLY DISPLAYED, THEN THE ITEM INQUIRY MENU IS WRITTEN                 03/20/89
  25700  C*    TO THE SCREEN.                                                             03/20/89
  25800  C*                                                                               10/13/87
  25900  C*****************************************************************               10/13/87
  26000  C* 8                                                                             10/13/87
  26100  C          ITMRTN    TAG                                                   001   10/13/87
  26200  C          *IN99     CABEQ'1'       END               JOB ENDS             001   10/13/87
  26300  C          *IN98     IFEQ '1'                                            B002   10/13/87
```

Figure 11-22 (Part 6 of 14). Source Program Example — RSDMUL (User-Defined Formats)

```
   26400 C                   WRITECIMENU                    MAIN MENU              002   10/13/87
   26500 C                   GOTO READRQ                                           002   10/13/87
   26600 C                   END                                                  E002   10/13/87
   26700 C           *IN97   IFEQ '1'                       CMD 3 ?                B002   10/13/87
   26800 C           RECID   IFEQ 'ITMSC2  '                ITM SCR 2 ?            B003   10/13/87
   26900 C                   WRITEITMMNU                    YES,THEN ITS           003   10/13/87
   27000 C                   GOTO READRQ                    ITEM MENU              003   10/13/87
   27100 C                   END                                                  E003   10/13/87
   27200 C                   END                                                  E002   10/13/87
   27300 C           RECID   IFEQ 'ITMSC3  '                ITM SCR 3 ?            B002   10/13/87
5728RG1 R01M02  881028          IBM AS/400 RPG/400              ICFLIB/RSDMUL      03/20/89  15:44:17      Page        9
SEQUENCE                                                                    IND    DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE     LINE   ID
   27400 C                   WRITEITMMNU                    YES,THEN ITS           002   10/13/87
   27500 C                   GOTO READRQ                    ITEM MENU              002   10/13/87
   27600 C                   END                                                  E002   10/13/87
   27700 C           SLSTM   SUB  CSTTM     PROFM   92      PROF MONTH             001   10/13/87
   27800 C                   MULT 100       PROFM                                 001   10/13/87
   27900 C           SLSTM   COMP 0                        46                 3    001   10/13/87
   28000 C     N46   PROFM   DIV  SLSTM     PROFM           PROF PCT              001   10/13/87
   28100 C           QTYLST  MULT PR01      LOSTS           LOST SALES            001   10/13/87
   28200 C                   MOVE SLSTM     SLSM                                  001   10/13/87
   28300 C                   MOVE SLSTY     SLSY                                  001   10/13/87
   28400 C                   MOVE CSTTM     CSTM                                  001   10/13/87
   28500 C                   MOVE PROFM     PROFIT                                001   10/13/87
   28600 C                   MOVE CSTTY     CSTY                                  001   10/13/87
   28700 C                   WRITEITMSC3                    DET ITM INF            001   10/13/87
   28800 C                   GOTO READRQ                                           001   10/13/87
   28900 C*****************************************************************       10/13/87
   29000 C*                                                                      10/13/87
   29100 C*                      CUSTOMER INQUIRY                                10/13/87
   29200 C*                                                                      10/13/87
   29300 C*    THE REQUEST FROM THE CUSTOMER INQUIRY MENU IS PROCESSED.          10/13/87
   29400 C*                                                                      10/13/87
   29500 C*    IF CMD 1 (*IN99) IS PRESSED, THE PROGRAM IS ENDED.                10/13/87
   29600 C*                                                                      10/13/87
   29700 C*    IF CMD 2 (*IN98) IS PRESSED, THE CUSTOMER INQUIRY IS ENDED,       10/13/87
   29800 C*    AND THE MAIN MENU (CIMENU) IS WRITTEN TO THE SCREEN.              10/13/87
   29900 C*                                                                      10/13/87
   30000 C*    IF A CUSTOMER NUMBER IS ENTERED, THE CUSTOMER INQUIRY             10/13/87
   30100 C*    REQUEST IS SENT TO THE REMOTE SYSTEM.                             10/13/87
   30200 C*                                                                      10/13/87
   30300 C*    A READ TO THE ICF PROGRAM DEVICE IS ISSUED TO RECEIVE THE         03/20/89
   30400 C*    INFORMATION FROM THE TARGET PROGRAM.                              10/13/87
   30500 C*                                                                      10/13/87
   30600 C*    IF AN ERROR OCCURS, THE ERROR IS PRINTED AND THE JOB IS           10/13/87
   30700 C*    ENDED.                                                            10/13/87
   30800 C*                                                                      10/13/87
   30900 C*****************************************************************       10/13/87
   31000 C* 9                                                                    10/13/87
   31100 C           DTLIN   TAG                                           001   10/13/87
   31200 C           *IN99   CABEQ'1'       END             JOB ENDS       001   10/13/87
   31300 C           *IN98   IFEQ '1'                                      B002   10/13/87
   31400 C                   WRITECIMENU                    MAIN MENU              002   10/13/87
   31500 C                   GOTO READRQ                                           002   10/13/87
   31600 C                   END                                                  E002   10/13/87
   31700 C           EVDTL   TAG                                           001   10/13/87
   31800 C                   MOVEL'ICF00   'CMID                           001   10/13/87
   31900 C                   WRITEDTLREQ                    CUST INQ               001   10/13/87
   32000 C           MAJCOD  CABGE'04'      ERROR           ERROR RTN              001   10/13/87
   32100 C           TRY88   TAG                                           001   10/13/87
   32200 C                   SETOF                          88                 3   001   10/13/87
   32300 C                   MOVEL'        'CMID                           001   10/13/87
   32400 C                   READ CMNFIL                 8810RCV CUS INF       2 3  001   10/13/87
   32500 C   88              EXSR ERRCHK                    CHECK ERR              001   10/13/87
   32600 C           MAJMIN  CABGE'0300'    EVDTL           NODATATRYAGN           001   10/13/87
```

Figure 11-22 (Part 7 of 14). Source Program Example — RSDMUL (User-Defined Formats)

```
   32700 C           MAJCOD    CABGE'04'      ERROR            ERROR RTN             001   10/13/87
5728RG1 R01M02  881028            IBM AS/400 RPG/400                    ICFLIB/RSDMUL       03/20/89  15:44:17      Page       10
SEQUENCE                                                                                IND   DO    LAST        PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE   NUM   UPDATE      LINE   ID
   32800 C           RECID2    CABNE'DTLRSP'  RECERR           PRINT MSG             001   10/13/87
   32900 C*****************************************************************              10/13/87
   33000 C*                                                                             10/13/87
   33100 C*                      PROCESS CUSTOMER INFORMATION                           10/13/87
   33200 C*                                                                             10/13/87
   33300 C*     THE CUSTOMER DATA RECEIVED FROM THE TARGET PROGRAM IS                   03/20/89
   33400 C*     PROCESSED. IF CUSTOMER NUMBER IS ZERO OR LESS, IT IS AN                 03/20/89
   33500 C*     INVALID REQUEST AND THE MAIN MENU IS WRITTEN TO THE SCREEN.             03/20/89
   33600 C*     WHEN THE RCVTRNRND INDICATOR(IN90) IS RECEIVED, THE CUSTOMER            03/20/89
   33700 C*     INFORMATION IS WRITTEN TO THE SCREEN. IF DURING THE READ                03/20/89
   33800 C*     OPERATION AN ERROR IS RECEIVED, CONTROL GOES TO THE ERROR               03/20/89
   33900 C*     ROUTINE TO END THE JOB.                                                 03/20/89
   34000 C*                                                                             10/13/87
   34100 C*****************************************************************              10/13/87
   34200 C* 10                                                                         10/13/87
   34300 C           DTOUT     TAG                                             001   10/13/87
   34400 C           CUSTNO    IFEQ 000000                                     B002  10/13/87
   34500 C                     SETOF                        66                 3 002  10/13/87
   34600 C                     WRITECIMENU                    MAIN MENU           002  10/13/87
   34700 C                     GOTO READRQ                                        002  10/13/87
   34800 C                     END                                             E002  10/13/87
   34900 C                     MOVE CUSTNO    CUSTN                            001   10/13/87
   35000 C                     MOVELDNAME     CNAME                            001   10/13/87
   35100 C                     MOVE DLSTOR    DLSTR                            001   10/13/87
   35200 C                     MOVE DSLSTM    DSLSM                            001   10/13/87
   35300 C                     MOVE DSPM01    DSPM1                            001   10/13/87
   35400 C                     MOVE DSPM02    DSPM2                            001   10/13/87
   35500 C                     MOVE DSTTYD    DSTYD                            001   10/13/87
   35600 C                     MOVE IDEPT     DEPT                             001   10/13/87
   35700 C                     WRITEDTLSCR                    BLD CUS SCR         001  10/13/87
   35800 C                     GOTO READRQ                                        001  10/13/87
   35900 C*****************************************************************              10/13/87
   36000 C*                                                                             10/13/87
   36100 C*    THIS ROUTINE HANDLES THE USER'S REQUEST FOLLOWING THE                    03/20/89
   36200 C*    DISPLAY OF THE CUSTOMER INFORMATION.  CMD KEY 1 WILL END                 03/20/89
   36300 C*    THE JOB, CMD KEY 2 WILL DISPLAY THE MAIN MENU, AND "ENTER"               03/20/89
   36400 C*    WILL BRING UP THE CUSTOMER INQUIRY MENU.                                 03/20/89
   36500 C*                                                                             10/13/87
   36600 C*****************************************************************              10/13/87
   36700 C* 11                                                                         10/13/87
   36800 C           DTLRTN    TAG                                             001   10/13/87
   36900 C           *IN99     CABEQ'1'       END             JOB ENDS         001   10/13/87
   37000 C           *IN98     IFEQ '1'                                        B002  10/13/87
   37100 C                     WRITECIMENU                    MAIN MENU           002  10/13/87
   37200 C                     GOTO READRQ                                        002  10/13/87
   37300 C                     END                                             E002  10/13/87
   37400 C                     WRITEDTLMNU                    CUSTOMER INQ        001  10/13/87
   37500 C                     GOTO READRQ                                        001  10/13/87
   37600 C*****************************************************************              10/13/87
   37700 C*                                                                             10/13/87
   37800 C*    WHEN AN I/O OPERATION ERROR IS DETECTED, A MESSAGE IS                    10/13/87
   37900 C*    PRINTED AND THE TRANSACTION AND SESSION ARE ENDED FOR EACH               03/20/89
   38000 C*    OF THE REMOTE SYSTEMS.                                                   03/20/89
   38100 C*                                                                             10/13/87
5728RG1 R01M02  881028            IBM AS/400 RPG/400                    ICFLIB/RSDMUL       03/20/89  15:44:17      Page       11
SEQUENCE                                                                                IND   DO    LAST        PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE   NUM   UPDATE      LINE   ID
   38200 C*****************************************************************              10/13/87
   38300 C* 12                                                                         10/13/87
   38400 C           RECERR    TAG                                             001   10/13/87
   38500 C                     EXCPTRECER                     WRONG RECID         001  10/13/87
   38600 C                     GOTO END                       END PROGRAM         001  10/13/87
   38700 C           ERROR     TAG                                             001   10/13/87
   38800 C                     EXCPTMMERR                                      001   10/13/87
```

Figure 11-22 (Part 8 of 14). Source Program Example — RSDMUL (User-Defined Formats)

```
38900 C           END       TAG                                                  001   10/13/87
39000 C                     MOVEL'ICF00   'CMID                                  001   10/13/87
39100 C                     WRITEDETACH                    DET 1ST TRN           001   10/13/87
39200 C                     MOVEL'ICF01   'CMID                                  001   10/13/87
39300 C                     WRITEDETACH                    DET 2ND TRN           001   10/13/87
39400 C                     MOVEL'ICF02   'CMID                                  001   10/13/87
39500 C                     WRITEDETACH                    DET 3RD TRN           001   10/13/87
39600 C                     MOVEL'ICF03   'CMID                                  001   10/13/87
39700 C                     WRITEDETACH                    DET 4TH TRN           001   10/13/87
39800 C           ABORT     TAG                                                  001   10/13/87
39900 C           'ICF00    'REL  CMNFIL              86  REL 1ST SES      2     001   10/13/87
40000 C           'ICF01    'REL  CMNFIL              86  REL 2ND SES      2     001   10/13/87
40100 C           'ICF02    'REL  CMNFIL              86  REL 3RD SES      2     001   10/13/87
40200 C           'ICF03    'REL  CMNFIL              86  REL 4TH SES      2     001   10/13/87
40300 C           FORCE     TAG                                                  001   10/13/87
40400 C                     SETON                  LR                  1        001   10/13/87
40500 C                     RETRN                                               001   10/13/87
40600 C                     END                                                 E001   10/13/87
40700 C*****************************************************************          10/13/87
40800 C*                                                                         10/13/87
40900 C*    THIS SUBROUTINE IS CALLED TO EVOKE THE TARGET PROGRAM.  THE          03/20/89
41000 C*    SAME TARGET PROGRAM (ICFLIB/RTDMULCL) IS EVOKED AT FOUR              03/20/89
41100 C*    DIFFERENT REMOTE SYSTEMS.  THE PROGRAM DEVICE IDENTIFIES             03/20/89
41200 C*    WHICH SESSION SHOULD BE EVOKED.  THE PROGRAM DEVICE WAS              03/20/89
41300 C*    SPECIFIED IN CMID PRIOR TO CALLING THIS ROUTINE.                     03/20/89
41400 C*                                                                         10/13/87
41500 C*****************************************************************          10/13/87
41600 C* 13                                                                      10/13/87
41700 C           EVKSR     BEGSR                                                10/13/87
41800 C                     MOVE *BLANK    PGMID           BLANK OUT             10/13/87
41900 C                     MOVE *BLANK    LIB             BLANK OUT             10/13/87
42000 C                     MOVEL'RTDMULCL'PGMID           PROGR NAME            10/13/87
42100 C                     MOVEL'ICFLIB  'LIB             LIBRARY               10/13/87
42200 C                     WRITEEVKREQ                                          10/13/87
42300 C           MAJCOD    CABGE'04'      END             TO END PGM            10/13/87
42400 C                     ENDSR                                                10/13/87
42500 C*****************************************************************          10/13/87
42600 C*                                                                         10/13/87
42700 C*    THIS SUBROUTINE IS CALLED TO PERFORM FURTHER CHECKS ON FILE           03/20/89
42800 C*    ERRORS RESULTING FROM THE READ OPERATION ISSUED TO THE PRO-           03/20/89
42900 C*    GRAM DEVICE. THIS ROUTINE CHECKS FOR THE TIME QUT INDICATION.         03/20/89
43000 C*    IF THERE IS A TIME OUT, A MESSAGE IS SENT TO THE USER'S               03/20/89
43100 C*    DISPLAY SCREEN REQUESTING ACTION, OTHERWISE PROGRAM ENDS.             10/16/87
43200 C*                                                                         10/13/87
43300 C*****************************************************************          10/13/87
43400 C* 14                                                                      10/13/87
43500 C           ERRCHK    BEGSR                                                10/13/87
5728RG1 R01M02  881028            IBM AS/400 RPG/400              ICFLIB/RSDMUL      03/20/89  15:44:17      Page      12
SEQUENCE                                                                  IND    DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE     LINE   ID
43600 C           MAJMIN    IFEQ '0310'                    TIMER EXPD?           B001   10/13/87
43700 C           CHKAGN    TAG                                                  001   10/13/87
43800 C                     WRITETIMOUT                    DISPLAY MSG           001   10/13/87
43900 C                     READ DSPFIL                   86READ REPLY       3   001   10/13/87
44000 C   88      TIMRSP    CABEQ'1'       TRY88           CUST INQUIR           001   10/13/87
44100 C   89      TIMRSP    CABEQ'1'       TRY89           ITEM INQUIR           001   10/13/87
44200 C           TIMRSP    IFEQ '2'                       END PROGRAM           B002   10/13/87
44300 C                     WRITEEOS                       END SESSION           002   10/13/87
44400 C                     GOTO FORCE                     END PROGRAM           002   10/13/87
44500 C                     END                                                 E002   10/13/87
44600 C                     GOTO CHKAGN                    ASK AGAIN             001   10/13/87
44700 C                     END                                                 E001   10/13/87
44800 C                     GOTO ERROR                     ABEND                 10/13/87
44900 C                     ENDSR                                                10/13/87
45000 C*****************************************************************          10/14/87
45100 C*                                                                         10/14/87
45200 C*    THIS IS THE PROGRAM ERROR SUBROUTINE THAT RECEIVES CONTROL            03/20/89
45300 C*    WHEN AN ERROR OCCURS AFTER AN I/O OPERATION IS ISSUED TO              03/20/89
```

Figure 11-22 (Part 9 of 14). Source Program Example — RSDMUL (User-Defined Formats)

```
   45400 C*    THE PROGRAM DEVICE AND THERE IS A NON-ZERO VALUE IN THE RPG                03/20/89
   45500 C*    STATUS FIELD (ERR). THIS ROUTINE CHECKS FOR STATUS VALUES                  03/20/89
   45600 C*    THAT RELATE TO ICF OPERATIONS. IF THE PROGRAM DEVICE                       03/20/89
   45700 C*    IS ALREADY ACQUIRED, THE ERROR IS IGNORED, OTHERWISE, THE                  03/20/89
   45800 C*    PROGRAM IS ENDED.                                                          03/20/89
   45900 C*                                                                               10/14/87
   46000 C*****************************************************************            10/14/87
   46100 C* 15                                                                          10/14/87
   46200 C           *PSSR     BEGSR                                                    10/14/87
   46300 C                     MOVE '      '  RETURN  6        DEFAULT                  10/14/87
   46400 C           ERR       CABEQ01285     ENDPSR           ALREADY ACQ?             10/14/87
   46500 C                     MOVE '*CANCL'  RETURN           JOB ENDS                 10/14/87
   46600 C           ENDPSR    ENDSRRETURN                     BACK TO MAIN             10/14/87
   46700 C*****************************************************************            10/13/87
   46800 OQPRINT  E 1             MMERR                                                 02/24/89
   46900 O                                     21 'COMMUNICATION ERROR.'                02/24/89
   47000 O                                     34 'MAJOR/MINOR:'                        02/24/89
   47100 O                        MAJCOD       37                                       02/24/89
   47200 O                                     38 '/'                                   02/24/89
   47300 O                        MINCOD       40                                       02/24/89
   47400 O                                     49 'FORMAT:'                             02/24/89
   47500 O                        FMTNM        60                                       02/24/89
   47600 O                                     69 'PGMDEV:'                             02/24/89
   47700 O                        CMID         80                                       02/24/89
   47800 O        E 1             RECER                                                 02/24/89
   47900 O                                     20 'UNMATCH RECD FORMAT'                 02/24/89
   48000 O                                     31 '-JOB ENDS.'                          02/24/89
   48100 O                        MAJCOD       37                                       02/24/89
   48200 O                                     38 '/'                                   02/24/89
   48300 O                        MINCOD       40                                       02/24/89
   48400 O                                     49 'FORMAT:'                             02/24/89
   48500 O                        RECID2       60                                       02/24/89
   48600 O                                     69 'PGMDEV:'                             02/24/89
   48700 O                        CMID         80                                       02/24/89
5728RG1 R01M02  881028            IBM AS/400 RPG/400                ICFLIB/RSDMUL      03/20/89  15:44:17      Page      13
SEQUENCE                                                                    IND   DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE   NUM   UPDATE     LINE   ID
* 6103     48701   OVERFLOW INDICATOR OA ASSIGNED TO FILE QPRINT.
  O000000   OUTPUT FIELDS FOR RECORD DETACH FILE CMNFIL FORMAT DETACH.
  P000000   OUTPUT FIELDS FOR RECORD EOS FILE CMNFIL FORMAT EOS.
  Q000000   OUTPUT FIELDS FOR RECORD EVKREQ FILE CMNFIL FORMAT EVKREQ.
  Q000001                          PGMID     10  CHAR   10
  Q000002                          LIB       20  CHAR   10
  R000000   OUTPUT FIELDS FOR RECORD ITMREQ FILE CMNFIL FORMAT ITMREQ.
  R000001                          ITEMNO     6  ZONE  6,0
  S000000   OUTPUT FIELDS FOR RECORD DTLREQ FILE CMNFIL FORMAT DTLREQ.
  S000001                          CUSTNO     6  ZONE  6,0
  T000000   OUTPUT FIELDS FOR RECORD CIMENU FILE DSPFIL FORMAT CIMENU.
  T000000         MENU FOR INQUIRY
  U000000   OUTPUT FIELDS FOR RECORD DTLMNU FILE DSPFIL FORMAT DTLMNU.
  U000000         CUSTOMER INQUIRY SCREEN 1
  V000000   OUTPUT FIELDS FOR RECORD DTLSCR FILE DSPFIL FORMAT DTLSCR.
  V000000         CUSTOMER INQUIRY SCR. #2
  V000001                          CUSTN      6  CHAR    6
  V000002                          DEPT       9  ZONE  3,0
  V000003                          DLSTR     15  ZONE  6,0
  V000004                          DSLSM     24  ZONE  9,0
  V000005                          DSPM1     33  ZONE  9,0
  V000006                          DSPM2     42  ZONE  9,0
  V000007                          DSPM3     51  ZONE  9,0
  V000008                          DSTYD     62  ZONE 11,0
  V000009                          NAME      67  CHAR    5
  W000000   OUTPUT FIELDS FOR RECORD ITMMNU FILE DSPFIL FORMAT ITMMNU.
  W000000         ITEM INQUIRY SCREEN ONE
  X000000   OUTPUT FIELDS FOR RECORD ITMSC2 FILE DSPFIL FORMAT ITMSC2.
  X000000         ITEM INQUIRY SCREEN TWO
  X000001                          DSC       30  CHAR   30
  X000002                          QAVAIL    37  ZONE  7,0
```

Figure 11-22 (Part 10 of 14). Source Program Example — RSDMUL (User-Defined Formats)

```
  X000003                          QTYH      44  ZONE  7,0
  X000004                          QTYO      51  ZONE  7,0
  X000005                          QTYB      58  ZONE  7,0
  X000006                          UNT       60  CHAR    2
  X000007                          PR1       67  ZONE  7,2
  X000008                          PR5       74  ZONE  7,0
  X000009                          UFR       79  ZONE  5,2
  Y000000   OUTPUT FIELDS FOR RECORD ITMSC3 FILE DSPFIL FORMAT ITMSC3.
  Y000000         ITEM INQUIRY SCREEN 3
  Y000001                          SLSM       9  ZONE  9,2
  Y000002                          SLSY      20  ZONE 11,2
  Y000003                          CSTM      29  ZONE  9,2
  Y000004                          CSTY      40  ZONE 11,2
  Y000005                          PROFIT    45  ZONE  5,2
  Y000006                          LOSTS     54  ZONE  9,2
  Z000000   OUTPUT FIELDS FOR RECORD TIMOUT FILE DSPFIL FORMAT TIMOUT.
  Z000000         TIME OUT SCREEN
          *****  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 R01M02  881028            IBM AS/400 RPG/400                ICFLIB/RSDMUL       03/20/89  15:44:17      Page      14
* 7089      4400   RPG PROVIDES SEPARATE INDICATOR AREA FOR FILE CMNFIL.
 5728RG1 R01M02  881028            IBM AS/400 RPG/400                ICFLIB/RSDMUL       03/20/89  15:44:17      Page      15
                      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 CMNFIL      WORKSTN        4400D    9700     9800     9900    10000
                                 19800    32400    39900    40000    40100
                                 40200
         DETACH                   4400D C000000    39100    39300    39500
                                 39700  O000000
         DTLREQ                   4400D G000000    31900  S000000
         DTLRSP                   4400D B000000
         EOS                      4400D D000000    44300  P000000
         EVKREQ                   4400D E000000    42200  Q000000
         ITMREQ                   4400D F000000    19300  R000000
         ITMRSP                   4400D A000000
    02 DSPFIL      WORKSTN        4900D   12500    43900
         CIMENU                   4900D H000000    11000    13200    17800
                                 26400    31400    34600    37100  T000000
         DTLMNU                   4900D I000000    15100    37400  U000000
         DTLSCR                   4900D J000000    35700  V000000
         ITMMNU                   4900D K000000    14900    21800    26900
                                 27400  W000000
         ITMSC2                   4900D L000000    23300  X000000
         ITMSC3                   4900D M000000    28700  Y000000
         TIMOUT                   4900D N000000    43800  Z000000
    03 QPRINT      PRINTER        5100D   46800    47800    48701
 Field References:
       FIELD       ATTR     REFERENCES (M=MODIFIED D=DEFINED)
       *IN97       A(1)     H000001  I000001  J000001  K000001  L000001
                            M000001  N000001    26700
       *IN98       A(1)     H000002  I000002  J000002  K000002  L000002
                            M000002  N000002    17700    26300    31300
                              37000
       *IN99       A(1)     H000003  I000003  J000003  K000003  L000003
                            M000003  N000003    14700    17600    26200
                              31200    36900
       *PSSR       BEGSR      4400    46200D
* 7031 ABORT       TAG       39800D
       CHKAGN      TAG       43700D   44600
       CMID        A(10)      7300D   10100M   10300M   10500M   10700M
                            18500M   18800M   19100M   19700M   31800M
                            32300M   39000M   39200M   39400M   39600M
                            47700    48700
       CSTM        P(9,2)    28400M Y000003D
       CSTTM       P(9,2)   A000014D   27700    28400
       CSTTY       P(11,2)  A000015D   28600
```

Figure 11-22 (Part 11 of 14). Source Program Example — RSDMUL (User-Defined Formats)

```
5728RG1 R01M02  881028            IBM AS/400 RPG/400                ICFLIB/RSDMUL        03/20/89  15:44:17      Page      16
        CSTY        P(11,2)   28600M Y000004D
        CUSTN       A(6)      34900M V000001D
        CUSTNO      P(6,0)  B000002D G000001D I000004D   34400    34900
                            S000001D
        DEPT        P(3,0)    35600M V000002D
        DESC        A(30)   A000003D   22500
        DLSTOR      P(6,0)  B000004D   35100
        DLSTR       P(6,0)    35100M V000003D
        DNAME       A(30)   B000003D   35000
        DSC         A(30)     22500M X000001D
        DSLSM       P(9,0)    35200M V000004D
        DSLSTM      P(9,0)  B000005D   35200
        DSPM01      P(9,0)  B000006D   35300
        DSPM02      P(9,0)  B000007D   35400
* 7031  DSPM03      P(9,0)  B000008D
        DSPM1       P(9,0)    35300M V000005D
        DSPM2       P(9,0)    35400M V000006D
        DSPM3       P(9,0)  V000007D
        DSTTYD      P(11,0) B000009D   35500
        DSTYD       P(11,0)   35500M V000008D
        DTLIN       TAG       13000    31100D
        DTLRTN      TAG       13100    36800D
* 7031  DTOUT       TAG       34300D
        END         TAG       14700    17600    26200    31200    36900
                              38600    38900D   42300
        ENDPSR      ENDSR     46400    46600D
* 7031  ENTRY       TAG        9600D
        ERR         Z(5,0)     7000D   46400
        ERRCHK      BEGSR     19900    32500    43500D
        ERROR       TAG       19400    20100    32000    32700    38700D
                              44800
        EVDTL       TAG       31700D   32600
        EVKSR       BEGSR     10200    10400    10600    10800    41700D
* 7031  FILL01      A(240)     6500D
* 7031  FILL02      A(145)     6700D
* 7031  FILL03      A(240)     7100D
* 7031  FILL04      A(145)     7800D
* 7031  FILL1       A(56)   A000018D
* 7031  FILL2       A(57)   B000011D
        FMTNM       A(8)       7200D   47500
        FORCE       TAG       40300D   44400
        IDEPT       P(3,0)  B000010D   35600
        IODS        DS(415)    4900     6400D
        IOFB        DS(415)    4400     6800D
        ITEMNO      P(6,0)  A000002D F000001D K000004D   18100    18200
                              18300    21700  R000001D
        ITMIN       TAG       12700    17500D   20000
* 7031  ITMOUT      TAG       21600D
        ITMRTN      TAG       12800    12900    26100D
        LIB         A(10)     41900M   42100M Q000002D
* 7031  LOC         A(8)       6900D
* 7031  LOS         P(9,2)  A000017D
        LOSTS       P(9,2)    28100M Y000006D
* 7031  MAIN        TAG       10900D
        MAJCOD      A(2)       7500D   19400    20100    32000    32700
                              42300    47100    48100
        MAJMIN      A(4)       7400D   20000    32600    43600
5728RG1 R01M02  881028            IBM AS/400 RPG/400                ICFLIB/RSDMUL        03/20/89  15:44:17      Page      17
        MENU        TAG       12600    14600D
        MINCOD      A(2)       7600D   47300    48300
        MMERR       EXCPT     38800    46800
        NAME        A(5)      35000M V000009D
        OPTION      A(1)    H000004D   14800
        PGMID       A(10)     41800M   42000M Q000001D
* 7031  PRO         P(5,2)  A000016D
        PROFIT      P(5,2)    28500M Y000005D
        PROFM       P(9,2)    27700D   27800M   28000    28000M   28500
```

Figure 11-22 (Part 12 of 14). Source Program Example — RSDMUL (User-Defined Formats)

```
          PR01        P(7,2)  A000009D   23000    28100
          PR05        P(7,0)  A000010D   23100
          PR1         P(7,2)    23000M X000007D
          PR5         P(7,0)    23100M X000008D
          QAVAIL      P(7,0)    22100D   22200M   22300M   22400M X000002D
          QTYB        P(7,0)    22800M X000005D
          QTYBO       P(7,0)  A000007D   22400    22800
          QTYH        P(7,0)    22700M X000003D
          QTYLST      P(7,0)  A000004D   28100
          QTYO        P(7,0)    22600M X000004D
          QTYOH       P(7,0)  A000005D   22200    22700
          QTYOO       P(7,0)  A000006D   22300    22600
          READRQ      TAG       12300D   13300    15300    17900    21900
                                23400    26500    27000    27500    28800
                                31500    34700    35800    37200    37500
* 7031     RECCUS      A(1)    B000001D
          RECER       EXCPT     38500    47800
          RECERR      TAG       20200    32800    38400D
          RECID       A(8)       6600D   12600    12700    12800    12900
                                13000    13100    26800    27300
          RECID2      A(8)       7700D   20200    32800    48500
* 7031     RECITM      A(1)    A000001D
          RETURN      A(6)      46300D   46500M   46600
          SLSM        P(9,2)    28200M Y000001D
          SLSTM       P(9,2)  A000012D   27700    27900    28000    28200
          SLSTY       P(11,2) A000013D   28300
          SLSY        P(11,2)   28300M Y000002D
          TIMRSP      A(1)    N000004D   44000    44100    44200
          TRY88       TAG       32100D   44000
          TRY89       TAG       19500D   44100
          UFR         P(5,2)    23200M X000009D
          UFRT        P(5,2)  A000011D   23200
          UNITQ       A(2)    A000008D   22900
          UNT         A(2)      22900M X000006D
          XICF01      TAG       18100    18400D
          XICF02      TAG       18200    18700D
          XICF03      TAG       18300    19000D
          XITMIN      TAG       18600    18900    19200D
          *BLANK      LITERAL   41800    41900
          '       '   LITERAL   19700    32300
          '     '     LITERAL   46300
          '*CANCL'    LITERAL   46500
          'CIMENU  '  LITERAL   12600
          'DTLMNU  '  LITERAL   13000
          'DTLRSP'    LITERAL   32800
          'DTLSCR  '  LITERAL   13100
          'ICFLIB  '  LITERAL   42100
          'ICF00   '  LITERAL    9700    10100    31800    39000    39900
5728RG1 R01M02  881028            IBM AS/400 RPG/400                ICFLIB/RSDMUL        03/20/89  15:44:17      Page      18
          'ICF01   '  LITERAL    9800    10300    18500    39200    40000
          'ICF02   '  LITERAL    9900    10500    18800    39400    40100
          'ICF03   '  LITERAL   10000    10700    19100    39600    40200
          'ITMMNU  '  LITERAL   12700
          'ITMRSP'    LITERAL   20200
          'ITMSC2  '  LITERAL   12800    26800
          'ITMSC3  '  LITERAL   12900    27300
          'RTDMULCL'  LITERAL   42000
          '0300'      LITERAL   20000    32600
          '0310'      LITERAL   43600
          '04'        LITERAL   19400    20100    32000    32700    42300
          '1'         LITERAL   14700    14800    17600    17700    26200
                                26300    26700    31200    31300    36900
                                37000    44000    44100
          '2'         LITERAL   44200
          0           LITERAL   22100    27900
          000000      LITERAL   21700    34400
          01285       LITERAL   46400
          100         LITERAL   27800
```

Figure 11-22 (Part 13 of 14). Source Program Example — RSDMUL (User-Defined Formats)

```
       399999      LITERAL   18100
       699999      LITERAL   18200
       899999      LITERAL   18300
 Indicator References:
       INDICATOR  REFERENCES (M=MODIFIED D=DEFINED)
       *IN        H000001  H000002  H000003  I000001  I000002  I000003
                  J000001  J000002  J000003  K000001  K000002  K000003
                  L000001  L000002  L000003  M000001  M000002  M000003
                  N000001  N000002  N000003    14700    17600    17700
                    26200    26300    26700    31200    31300    36900
                    37000
       LR           40400M
       OA            5100D   48701
* 7031 10           19800M   32400M
       46           27900M   28000
* 7031 66           34500M
* 7031 86           39900M   40000M   40100M   40200M   43900M
* 7031 87           12500M
       88           12400M   32200M   32400M   32500    44000
       89           12400M   19600M   19800M   19900    44100
* 7031 90
       97         H000001  I000001  J000001  K000001  L000001  M000001
                  N000001    26700
       98         H000002  I000002  J000002  K000002  L000002  M000002
                  N000002    17700    26300    31300    37000
       99         H000003  I000003  J000003  K000003  L000003  M000003
                  N000003    14700    17600    26200    31200    36900
     * * * * *  E N D  O F  C R O S S  R E F E R E N C E  * * * * *
 5728RG1 R01M02  881028            IBM AS/400 RPG/400                 ICFLIB/RSDMUL       03/20/89  15:44:17      Page      19
                       M e s s a g e  S u m m a r y
* QRG6103 Severity:  00   Number:    1
          Message . . . . :   No Overflow Indicator is specified but an
            indicator is assigned to a file and automatic skip to 6 is
            generated.
* QRG7031 Severity:  00   Number:   22
          Message . . . . :   The Name or indicator is not referenced.
* 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 R01M02  881028            IBM AS/400 RPG/400                 ICFLIB/RSDMUL       03/20/89  15:44:17      Page      20
                        F i n a l  S u m m a r y
 Message Count:  (by Severity Number)
            TOTAL    00     10     20     30     40     50
              24     24      0      0      0      0      0
 Program Source Totals:
    Records . . . . . . . . . . :   487
    Specifications  . . . . . . :   245
    Table Records . . . . . . . :   0
    Comments  . . . . . . . . . :   242
 PRM has been called.
 Program RSDMUL is placed in library ICFLIB. 00 highest Error-Severity-Code.
          * * * * *  E N D  O F  C O M P I L A T I O N  * * * * *
```

Figure 11-22 (Part 14 of 14). Source Program Example — RSDMUL (User-Defined Formats)

```
5728RG1 R01M02  881028            IBM AS/400 RPG/400                 ICFLIB/RSFMUL        03/20/89  15:45:25      Page       1
Compiler . . . . . :   IBM AS/400 RPG/400
Command Options:
  Program  . . . . . :   ICFLIB/RSFMUL
  Source file  . . . . . :   ICFLIB/QICFPUB
  Source member  . . . . . :   *PGM
  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
  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 . . . . . :   RSFMUL
  File . . . . . :   QICFPUB
  Library  . . . . . :   ICFLIB
  Last Change  . . . . . :   03/20/89  15:42:07
  Description  . . . . . :   RPG Multi-Session example w/$$FORMAT (source)
5728RG1 R01M02  881028            IBM AS/400 RPG/400                 ICFLIB/RSFMUL        03/20/89  15:45:25      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  H*****************************************************************                10/13/87
     200  H*                                                                              03/20/89
     300  H*  THIS PROGRAM ASSIGNS FOUR SESSIONS AS FOLLOWS:                              10/13/87
     400  H*    'ICF00' TO INQUIRE ABOUT A CUSTOMER ACCOUNT BEFORE AN                     10/13/87
     500  H*            ORDER IS PROCESSED.                                               10/13/87
     600  H*    'ICF01' TO INQUIRE ABOUT THE INVENTORY STATUS OF AN ITEM                  10/13/87
     700  H*            BEING ORDERED (ITEM 000001 THRU 399999).                          10/13/87
     800  H*    'ICF02' TO INQUIRE ABOUT THE INVENTORY STATUS OF AN ITEM                  10/13/87
     900  H*            BEING ORDERED (ITEM 400000 THRU 699999).                          10/13/87
    1000  H*    'ICF03' TO INQUIRE ABOUT THE INVENTORY STATUS OF AN ITEM                  10/13/87
    1100  H*            BEING ORDERED (ITEM 700000 THRU 999999).                          10/13/87
    1200  H*  A DISPLAY DEVICE IS USED TO ENTER THE REQUEST ( USING A                    10/13/87
    1300  H*  CUSTOMER AND AN ITEM MENU) THAT IS SENT TO THE REMOTE                       10/13/87
    1400  H*  SYSTEM.                                                                    10/13/87
    1500  H*                                                                              03/20/89
    1600  F*****************************************************************             10/13/87
    1700  F*                                                                            10/13/87
    1800  F*                    F I L E   S P E C I F I C A T I O N S                   10/13/87
    1900  F*                                                                            10/13/87
    2000  F*    CMNFIL :   ICF FILE USED TO SEND A REQUEST TO ONE                       01/18/88
    2100  F*               OF FOUR DIFFERENT TARGET PROGRAMS.  MULTIPLE                10/13/87
    2200  F*               SESSIONS ARE ACTIVE AT THE SAME TIME.                       10/13/87
    2300  F*                                                                           10/13/87
    2400  F*    DSPFIL :   DISPLAY FILE USED TO ENTER A REQUEST TO BE                   10/13/87
    2500  F*               SENT TO A REMOTE SYSTEM.                                    10/13/87
    2600  F*                                                                           10/13/87
    2700  F*    THE FOLLOWING INFORMATION IS SPECIFIED AS PART OF THE                   10/13/87
    2800  F*    FILE SPECIFICATION:                                                     10/13/87
    2900  F*                 INFDS : I/O FEEDBACK AREA                                 10/13/87
    3000  F*                 NUM   : SPECIFIES THE MAXIMUM NUMBER OF PRO-              03/20/89
    3100  F*                         GRAM DEVICES THAT CAN BE ATTACHED TO              03/20/89
    3200  F*                         THIS FILE.  A VALUE OF 4 IS SPECIFIED             03/20/89
    3300  F*                         FOR THE ICF FILE.  THIS DEFINES                   03/20/89
    3400  F*                         THE FILE AS A MULTIPLE DEVICE FILE.               03/20/89
    3500  F*                 ID    : 10 CHARACTER PROGRAM DEVICE NAME FIELD            03/20/89
    3600  F*                         WHICH SPECIFIES WHICH PROGRAM DEVICE TO           03/20/89
    3700  F*                         DIRECT THE OPERATION.                             03/20/89
    3800  F*                         ID    : 10 CHARACTER PROGRAM DEVICE NAME          10/13/87
```

Figure 11-23 (Part 1 of 14). Source Program Example — RSFMUL (System-Supplied Formats)

```
    3900 F*                              FIELD THAT SPECIFIES WHICH PROGRAM              03/20/89
    4000 F*                              DEVICE TO DIRECT THE OPERATION.                 03/20/89
    4100 F*                                                                              10/13/87
    4200 F*                                                                              10/13/87
    4300 F*****************************************************************              10/13/87
    4400 H* 1                                                                            10/13/87
    4500 FCMNF1L  CF  F     150            WORKSTN                                       10/13/87
    4600 F                                            KINFDS IOFB                        10/13/87
    4700 F                                            KINFSR *PSSR                       10/14/87
    4800 F                                            KNUM       4                       10/13/87
5728RG1 R01M02  881028            IBM AS/400 RPG/400                 ICFLIB/RSFMUL       03/20/89  15:45:25      Page        3
SEQUENCE                                                                        IND    DO    LAST       PAGE    PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE     LINE    ID
    4900 F                                            KID    CMID                        10/13/87
    5000 FDSPFIL  CF  E                    WORKSTN                                       10/13/87
    5100 F                                            KINFDS IODS                        10/13/87
          RECORD FORMAT(S):  LIBRARY ICFLIB FILE DSPFIL.
                   EXTERNAL FORMAT CIMENU RPG NAME CIMENU
                   EXTERNAL FORMAT DTLMNU RPG NAME DTLMNU
                   EXTERNAL FORMAT DTLSCR RPG NAME DTLSCR
                   EXTERNAL FORMAT ITMMNU RPG NAME ITMMNU
                   EXTERNAL FORMAT ITMSC2 RPG NAME ITMSC2
                   EXTERNAL FORMAT ITMSC3 RPG NAME ITMSC3
                   EXTERNAL FORMAT TIMOUT RPG NAME TIMOUT
    5200 FQPRINT  O   F     132            PRINTER                                       10/13/87
    5300 I*****************************************************************              10/13/87
    5400 I*                                                                              10/13/87
    5500 I*                  I N P U T   S P E C I F I C A T I O N S                     10/13/87
    5600 I*                                                                              10/13/87
    5700 I*  IODS    :  REDEFINES THE I/O FEEDBACK AREA OF THE DISPLAY                   10/13/87
    5800 I*              FILE.  THIS AREA CONTAINS THE NAME OF THE LAST                  10/13/87
    5900 I*              RECORD PROCESSED.  THIS FIELD IS CALLED RECID.                  10/13/87
    6000 I*  IOFB    :  REDEFINES THE I/O FEEDBACK AREA FOR THE ICF                      01/18/88
    6100 I*              FILE.                                                           10/13/87
    6200 I*                                                                              03/20/89
    6300 I*****************************************************************              10/13/87
    6400  *                                                                              03/20/89
    6500 ICMNFIL  NS  80   1 CC                                                          03/20/89
    6600 I                                        1   1 RECCUS                           10/13/87
    6700 I                                        2   7 0CUSTNO                          10/13/87
    6800 I                                        8  37 DNAME                            10/13/87
    6900 I                                       38  43 DLSTOR                           10/13/87
    7000 I                                       44  52 DSLSTM                           10/13/87
    7100 I                                       53  61 DSPM01                           10/13/87
    7200 I                                       62  70 DSPM02                           10/13/87
    7300 I                                       71  79 DSPM03                           10/13/87
    7400 I                                       80  90 DSTTYD                           10/13/87
    7500 I                                       91  93 IDEPT                            10/13/87
    7600 I                                       94 150 FILL20                           10/13/87
    7700 I        NS  81   1 CI                                                          10/13/87
    7800 I                                        1   1 RECITM                           10/13/87
    7900 I                                        2   7 0ITEMNO                          10/13/87
    8000 I                                        8  37 DESC                             10/13/87
    8100 I                                       38  44 0QTYLST                         10/13/87
    8200 I                                       45  51 0QTYOH                          10/13/87
    8300 I                                       52  58 0QTYOO                          10/13/87
    8400 I                                       59  65 0QTYBO                          10/13/87
    8500 I                                       66  67 UNIT                            10/13/87
    8600 I                                       68  74 2PR01                           10/13/87
    8700 I                                       75  80 0PR05                           10/13/87
    8800 I                                       81  85 2UFRT                           10/13/87
    8900 I                                       86  94 2SLSTM                          10/13/87
    9000 I                                       95 105 2SLSTY                          10/13/87
    9100 I                                      106 114 2CSTTM                          10/13/87
```

Figure 11-23 (Part 2 of 14). Source Program Example — RSFMUL (System-Supplied Formats)

```
5728RG1 R01M02  881028            IBM AS/400 RPG/400                 ICFLIB/RSFMUL       03/20/89  15:45:25      Page       4
SEQUENCE                                                                                IND    DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE     LINE   ID
    9200  I                                    115 1252CSTTY                                  10/13/87
    9300  I                                    126 1302PRO                                    10/13/87
    9400  I                                    131 1392LOS                                    10/13/87
    9500  I                                    140 150 FILL10                                 10/13/87
    9600  I*****************************************************                              10/13/87
 A000000   INPUT  FIELDS FOR RECORD CIMENU FILE DSPFIL FORMAT CIMENU.
 A000000         MENU FOR INQUIRY
 A000001                                       3   3 *IN97
 A000002                                       2   2 *IN98
 A000003                                       1   1 *IN99
 A000004                                       4   4 OPTION
 B000000   INPUT  FIELDS FOR RECORD DTLMNU FILE DSPFIL FORMAT DTLMNU.
 B000000         CUSTOMER INQUIRY SCREEN 1
 B000001                                       3   3 *IN97
 B000002                                       2   2 *IN98
 B000003                                       1   1 *IN99
 B000004                                       4   90CUSTNO
 C000000   INPUT  FIELDS FOR RECORD DTLSCR FILE DSPFIL FORMAT DTLSCR.
 C000000         CUSTOMER INQUIRY SCR. #2
 C000001                                       3   3 *IN97
 C000002                                       2   2 *IN98
 C000003                                       1   1 *IN99
 D000000   INPUT  FIELDS FOR RECORD ITMMNU FILE DSPFIL FORMAT ITMMNU.
 D000000         ITEM INQUIRY SCREEN ONE
 D000001                                       3   3 *IN97
 D000002                                       2   2 *IN98
 D000003                                       1   1 *IN99
 D000004                                       4   90ITEMNO
 E000000   INPUT  FIELDS FOR RECORD ITMSC2 FILE DSPFIL FORMAT ITMSC2.
 E000000         ITEM INQUIRY SCREEN TWO
 E000001                                       3   3 *IN97
 E000002                                       2   2 *IN98
 E000003                                       1   1 *IN99
 F000000   INPUT  FIELDS FOR RECORD ITMSC3 FILE DSPFIL FORMAT ITMSC3.
 F000000         ITEM INQUIRY SCREEN 3
 F000001                                       3   3 *IN97
 F000002                                       2   2 *IN98
 F000003                                       1   1 *IN99
 G000000   INPUT  FIELDS FOR RECORD TIMOUT FILE DSPFIL FORMAT TIMOUT.
 G000000         TIME OUT SCREEN
 G000001                                       3   3 *IN97
 G000002                                       2   2 *IN98
 G000003                                       1   1 *IN99
 G000004                                       4   4 TIMRSP
    9700  IIODS 2         DS                                                          10/13/87
    9800  I                                      1 240 FILL01                       10/13/87
    9900  I                                    261 268 RECID                        10/13/87
   10000  I                                    271 415 FILL02                       10/13/87
   10100  IIOFB        DS                                                          10/13/87
   10200  I                                   *ROUTINE LOC                         10/14/87
   10300  I                                   *STATUS  ERR                         10/14/87
   10400  I                                      1 240 FILL03                       10/13/87
   10500  I                                     38  47 FMTNM                        10/13/87
   10600  I                                    273 282 CMID                         10/13/87
5728RG1 R01M02  881028            IBM AS/400 RPG/400                 ICFLIB/RSFMUL       03/20/89  15:45:25      Page       5
SEQUENCE                                                                                IND    DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE     LINE   ID
   10700  I                                    401 404 MAJMIN                         10/13/87
   10800  I                                    401 402 MAJCOD                         10/13/87
   10900  I                                    403 404 MINCOD                         10/13/87
   11000  I                                    261 268 RECID2                         10/13/87
   11100  I                                    271 415 FILL04                         10/13/87
   11200  C*****************************************************************         10/13/87
   11300  C*                                                                        10/13/87
```

Figure 11-23 (Part 3 of 14). Source Program Example — RSFMUL (System-Supplied Formats)

```
11400 C*          C A L C U L A T I O N   S P E C I F I C A T I O N S          10/13/87
11500 C*   THE DISPLAY DEVICE IS IMPLICITLY ACQUIRED WHEN THE                  10/16/87
11600 C*   FILE IS OPENED.                                                    10/13/87
11700 C*                                                                      10/13/87
11800 C*   ALL OF THE PROGRAM DEVICES ARE EXPLICITLY ACQUIRED.                10/16/87
11900 C*                                                                      10/13/87
12000 C*   EACH OF THE FOUR TARGET PROGRAMS ARE EVOKED TO ESTABLISH           10/13/87
12100 C*   TRANSACTIONS WITH THE REMOTE SYSTEMS.                              10/13/87
12200 C*                                                                      10/13/87
12300 C*   THE MAIN INQUIRY MENU (CIMENU) IS WRITTEN TO THE USER'S            10/13/87
12400 C*   DISPLAY.                                                           10/13/87
12500 C*                                                                      10/13/87
12600 C*************************************************************          10/13/87
12700 C* 3                                                                    03/20/89
12800 C           BEGIN     TAG                                               10/13/87
12900 C           'ICF00    'ACQ  CMNFIL                  1ST SESSION         10/13/87
13000 C           'ICF01    'ACQ  CMNFIL                  2ND SESSION         10/13/87
13100 C           'ICF02    'ACQ  CMNFIL                  3RD SESSION         10/13/87
13200 C           'ICF03    'ACQ  CMNFIL                  4TH SESSION         10/13/87
13300 C                     MOVEL'ICF00   'CMID           1ST PROGRAM         10/13/87
13400 C                     EXSR EVKSR                    CALL EVOKE          10/13/87
13500 C                     MOVEL'ICF01   'CMID           2ND PROGRAM         10/13/87
13600 C                     EXSR EVKSR                    CALL EVOKE          10/13/87
13700 C                     MOVEL'ICF02   'CMID           3RD PROGRAM         10/13/87
13800 C                     EXSR EVKSR                    CALL EVOKE          10/13/87
13900 C                     MOVEL'ICF03   'CMID           4TH PROGRAM         10/13/87
14000 C                     EXSR EVKSR                    CALL EVOKE          10/13/87
14100 C           MAIN      TAG                                               10/13/87
14200 C                     WRITECIMENU                                       10/13/87
14300 C*************************************************************          10/13/87
14400 C*                                                                      10/13/87
14500 C*                    DETERMINE USER'S REQUEST                          10/13/87
14600 C*                                                                      10/13/87
14700 C*   A READ OPERATION IS ISSUED TO THE DISPLAY PROGRAM DEVICE            10/13/87
14800 C*   TO RECEIVE THE USER'S REQUEST. THE TYPE OF REQUEST MADE IS         03/20/89
14900 C*   BASED ON THE DISPLAY FORMAT CURRENTLY ON THE SCREEN. THE           03/20/89
15000 C*   RECORD FORMAT NAME IS EXTRACTED FROM THE I/O FEEDBACK AREA         03/20/89
15100 C*   AND USED TO DETERMINE WHAT ACTION SHOULD BE TAKEN NEXT.            03/20/89
15200 C*                                                                      10/13/87
15300 C*************************************************************          10/13/87
15400 C* 4                                                                    10/13/87
15500 C           READRQ    TAG                                               10/13/87
15600 C                     READ DSPFIL                 88                3   10/13/87
15700 C           RECID     CABEQ'CIMENU  'MENU           MAIN MENU           10/13/87
15800 C           RECID     CABEQ'ITMMNU  'ITMIN          AT ITEM SCR?        10/13/87
5728RG1 R01M02  881028          IBM AS/400 RPG/400              ICFLIB/RSFMUL     03/20/89  15:45:25      Page       6
SEQUENCE                                                                 IND   DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE   NUM   UPDATE     LINE   ID
15900 C           RECID     CABEQ'ITMSC2  'ITMRTN         AT ITM SCR?         10/13/87
16000 C           RECID     CABEQ'ITMSC3  'ITMRTN         AT ITM SCR?         10/13/87
16100 C           RECID     CABEQ'DTLMNU  'DTLIN          FOR DETAIL?         10/13/87
16200 C           RECID     CABEQ'DTLSCR  'DTLRTN         CUST SCR?           10/13/87
16300 C                     WRITECIMENU                   MAIN MENU           10/13/87
16400 C                     GOTO READRQ                   THERE IS ERR        10/13/87
16500 C*************************************************************          10/13/87
16600 C*                                                                      10/13/87
16700 C*                           MAIN MENU                                  10/13/87
16800 C*                                                                      10/13/87
16900 C*    THE MAIN MENU IS READ TO DETERMINE THE REQUEST ENTERED BY         03/20/89
17000 C*    THE USER. IF CMD 1 (*IN99) IS PRESSED, THE PROGRAM IS ENDED.      03/20/89
17100 C*    IF OPTION = 1, AN ITEM INQUIRY MENU IS WRITTEN TO THE SCREEN.     03/20/89
17200 C*    IF OPTION = 2, A CUSTOMER INQUIRY MENU IS WRITTEN TO THE          03/20/89
17300 C*    SCREEN.                                                           03/20/89
17400 C*                                                                      10/13/87
17500 C*************************************************************          10/13/87
17600 C* 5                                                                    10/13/87
17700 C           MENU      TAG                                               10/13/87
17800 C           *IN99     CABEQ'1'      END             END PROGRAM         10/13/87
```

Figure 11-23 (Part 4 of 14). Source Program Example — RSFMUL (System-Supplied Formats)

```
17900 C           OPTION    IFEQ '1'                                          B001   10/13/87
18000 C                     WRITEITMMNU                 ITEM MENU              001   10/13/87
18100 C                     ELSE                                              X001   10/13/87
18200 C                     WRITEDTLMNU                 CUST MENU              001   10/13/87
18300 C                     END                                               E001   10/13/87
18400 C                     GOTO READRQ                                             10/13/87
18500 C*****************************************************************     10/13/87
18600 C*                                                                     10/13/87
18700 C*                        ITEM INQUIRY                                 10/13/87
18800 C*                                                                     10/13/87
18900 C*    THE ITEM NUMBER REQUESTED BY THE USER ON THE ITEM INQUIRY        10/13/87
19000 C*    SCREEN IS CHECKED.  THIS IS DETERMINED BY THE DISPLAY            03/20/89
19100 C*    RECORD FORMAT BEING PROCESSED - IN THIS CASE ITMMNU.             03/20/89
19200 C*                                                                     10/13/87
19300 C*    IF CMD 1 (*IN99) IS PRESSED, THE PROGRAM IS ENDED.  IF CMD 2     10/13/87
19400 C*    IS PRESSED, THE ITEM INQUIRY REQUEST IS CANCELED, AND THE        10/13/87
19500 C*    MAIN MENU (CIMENU) IS WRITTEN TO THE SCREEN.                     10/13/87
19600 C*                                                                     10/13/87
19700 C*    IF AN ITEM NUMBER IS ENTERED, A ITEM INQUIRY REQUEST IS          10/13/87
19800 C*    SENT TO THE APPROPRIATE REMOTE SYSTEM.  THE REMOTE SYSTEM        10/13/87
19900 C*    IS SELECTED BASED ON THE ITEM NUMBER REQUESTED.                  10/13/87
20000 C*                                                                     10/13/87
20100 C*    IF AN ERROR OCCURS, THE ERROR IS PRINTED AND THE JOB             10/13/87
20200 C*    IS ENDED.                                                        10/13/87
20300 C*                                                                     10/13/87
20400 C*****************************************************************     10/13/87
20500 C* 6                                                                   10/13/87
20600 C           ITMIN     TAG                                              10/13/87
20700 C           *IN99     CABEQ'1'       END          EXIT ON CMD3         10/13/87
20800 C           *IN98     IFEQ '1'                                          B001   10/13/87
20900 C                     WRITECIMENU                 MAIN MENU              001   10/13/87
21000 C                     GOTO READRQ                                        001   10/13/87
21100 C                     END                                               E001   10/13/87
21200 C           ITEMNO    CABLE399999    XICF01                                   10/13/87
5728RG1 R01M02  881028            IBM AS/400 RPG/400                ICFLIB/RSFMUL      03/20/89  15:45:25      Page      7
SEQUENCE                                                                    IND   DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE   NUM   UPDATE     LINE   ID
21300 C           ITEMNO    CABLE699999    XICF02                                   10/13/87
21400 C           ITEMNO    CABLE899999    XICF03                                   10/13/87
21500 C           XICF01    TAG                                                     10/13/87
21600 C                     MOVEL'ICF01    'CMID                                    10/13/87
21700 C                     GOTO XITMIN                                             10/13/87
21800 C           XICF02    TAG                                                     10/13/87
21900 C                     MOVEL'ICF02    'CMID                                    10/13/87
22000 C                     GOTO XITMIN                                             10/13/87
22100 C           XICF03    TAG                                                     10/13/87
22200 C                     MOVEL'ICF03    'CMID                                    10/13/87
22300 C           XITMIN    TAG                                                     10/13/87
22400 C                     EXCPTITEMRQ                 INQ W/INV                   10/13/87
22500 C           MAJCOD    CABGE'04'      ERROR        ERROR RTN                   10/13/87
22600 C           TRY89     TAG                                                     10/13/87
22700 C                     SETOF                  89                   1         10/13/87
22800 C                     MOVEL' '       RECITM       RESET RECID                 10/13/87
22900 C                     MOVEL'         'CMID                                    10/13/87
23000 C                     READ CMNFIL              8910                 2 3       10/13/87
23100 C   89                EXSR ERRCHK                 FILE ERROR?                 10/13/87
23200 C           MAJMIN    CABGE'0300'    ITMIN        NODATA?                     10/13/87
23300 C  N81                EXCPTNOTITM                 REC NOT FD?                 10/13/87
23400 C                     SETOF                  81                   1         10/13/87
23500 C           MAJCOD    CABGE'04'      ERROR        ERROR RTN                   10/13/87
23600 C           RECITM    CABNE'I'       RECERR       PRINT MSG                   10/13/87
23700 C*****************************************************************     10/13/87
23800 C*                                                                     10/13/87
23900 C*                     PROCESS ITEM INFORMATION                        10/13/87
24000 C*                                                                     10/13/87
24100 C*    THE ITEM RECORD RECEIVED FROM THE TARGET PROGRAM AND THE         03/20/89
24200 C*    INFORMATION ABOUT THE ITEM IS PROCESSED AND DISPLAYED. IF        03/20/89
24300 C*    ITEMNO IS 0 OR LESS, IT IS AN INVALID REQUEST AND A FRESH        03/20/89
```

Figure 11-23 (Part 5 of 14). Source Program Example — RSFMUL (System-Supplied Formats)

```
   24400 C*    ITEM MENU IS WRITTEN TO THE SCREEN. IF REQUEST IS VALID,                  03/20/89
   24500 C*    VALUES ARE CALCULATED BASED ON THE INFORMATION RECEIVED.                  03/20/89
   24600 C*                                                                              10/13/87
   24700 C*************************************************************                10/13/87
   24800 C* 7                                                                           03/20/89
   24900 C           ITMOUT    TAG                                                     10/13/87
   25000 C           ITEMNO    IFLE 000000                                    B001     10/13/87
   25100 C                     WRITEITMMNU                 ITEM MENU           001     10/13/87
   25200 C                     GOTO READRQ                 READ DISPLAY        001     10/13/87
   25300 C                     ELSE                                           X001     10/13/87
   25400 C                     Z-ADD0         QAVAIL  70   QTY AVAIL.          001     10/13/87
   25500 C                     ADD  QTYOH     QAVAIL                          001     10/13/87
   25600 C                     SUB  QTYOO     QAVAIL                          001     10/13/87
   25700 C                     ADD  QTYBO     QAVAIL                          001     10/13/87
   25800 C                     MOVELDESC      DSC                             001     10/13/87
   25900 C                     MOVE QTYOO     QTYO                            001     10/13/87
   26000 C                     MOVE QTYOH     QTYH                            001     10/13/87
   26100 C                     MOVE QTYBO     QTYB                            001     10/13/87
   26200 C                     MOVE UNIT      UNT                             001     10/13/87
   26300 C                     MOVE PR01      PR1                             001     10/13/87
   26400 C                     MOVE PR05      PR5                             001     10/13/87
   26500 C                     MOVE UFRT      UFR                             001     10/13/87
   26600 C                     WRITEITMSC2                 ITEM DETAIL         001     10/13/87
5728RG1 R01M02  881028            IBM AS/400 RPG/400              ICFLIB/RSFMUL      03/20/89  15:45:25      Page        8
SEQUENCE                                                                    IND   DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE   NUM   UPDATE     LINE   ID
   26700 C                     GOTO READRQ                                     001     10/13/87
   26800 C                     END                                            E001     10/13/87
   26900 C*************************************************************                10/13/87
   27000 C*                                                                              10/13/87
   27100 C*                    ADDITIONAL ITEM INFORMATION                               10/13/87
   27200 C*                                                                              10/13/87
   27300 C*    ADDITIONAL ITEM INFORMATION IS PROCESSED AND THE RESULT                   10/13/87
   27400 C*    IS DISPLAYED ON THE SCREEN WHEN A RESPONSE IS READ                        03/20/89
   27500 C*    FROM THE DISPLAY WITH AN ITEM SCREEN RECORD FORMAT.                       03/20/89
   27600 C*                                                                              10/13/87
   27700 C*    IF CMD 1 (*IN99) IS PRESSED, THE PROGRAM IS ENDED.                        10/13/87
   27800 C*                                                                              10/13/87
   27900 C*    IF CMD 2 (*IN98) IS PRESSED, THE ITEM INQUIRY IS ENDED,                   03/20/89
   28000 C*    AND THE MAIN MENU (CIMENU) IS WRITTEN TO THE SCREEN.                      03/20/89
   28100 C*                                                                              10/13/87
   28200 C*    IF CMD 3 (*IN97) IS PRESSED, THE ITEM INQUIRY MENU IS                     10/14/87
   28300 C*    WRITTEN ON THE SCREEN.                                                    10/14/87
   28400 C*                                                                              10/14/87
   28500 C*    IF 'ENTER' IS PRESSED WHILE SCREEN 2 FOR ITEM REQUESTED IS                10/16/87
   28600 C*    CURRENTLY DISPLAYED, MORE INFORMATION IS CALCULATED AND                   10/16/87
   28700 C*    DISPLAYED.                                                                10/16/87
   28800 C*                                                                              10/16/87
   28900 C*    IF 'ENTER' IS PRESSED WHILE SCREEN 3 FOR ITEM REQUESTED IS                10/16/87
   29000 C*    CURRENTLY DISPLAYED, THEN THE ITEM INQUIRY MENU IS WRITTEN                03/20/89
   29100 C*    TO THE SCREEN.                                                            03/20/89
   29200 C*                                                                              10/13/87
   29300 C*************************************************************                10/13/87
   29400 C* 8                                                                           03/20/89
   29500 C           ITMRTN    TAG                                                     10/13/87
   29600 C           *IN99     CABEQ'1'       END          JOB ENDS                    10/13/87
   29700 C           *IN98     IFEQ '1'                                       B001     10/13/87
   29800 C                     WRITECIMENU                 MAIN MENU           001     10/13/87
   29900 C                     GOTO READRQ                                     001     10/13/87
   30000 C                     END                                            E001     10/13/87
   30100 C           *IN97     IFEQ '1'                    CMD 3 ?             B001    10/13/87
   30200 C           RECID     IFEQ 'ITMSC2  '             ITM SCR 2 ?         B002    10/13/87
   30300 C                     WRITEITMMNU                 YES,THEN ITS        002     10/13/87
   30400 C                     GOTO READRQ                 ITEM MENU           002     10/13/87
   30500 C                     END                                            E002     10/13/87
   30600 C                     END                                            E001     10/13/87
   30700 C           RECID     IFEQ 'ITMSC3  '             ITM SCR 3 ?         B001    10/13/87
   30800 C                     WRITEITMMNU                 YES,THEN ITS        001     10/13/87
```

Figure 11-23 (Part 6 of 14). Source Program Example — RSFMUL (System-Supplied Formats)

```
   30900  C                   GOTO READRQ                         ITEM MENU              001   10/13/87
   31000  C                   END                                                        E001   10/13/87
   31100  C           SLSTM   SUB  CSTTM     PROFM   92           PROFIT MONTH                  10/13/87
   31200  C                   MULT 100       PROFM                                         10/13/87
   31300  C           SLSTM   COMP 0                           46                  3         10/13/87
   31400  C     N46   PROFM   DIV  SLSTM     PROFM               PROFIT PCT                 10/13/87
   31500  C           QTYLST  MULT PR01      LOSTS               LOST SALES                 10/13/87
   31600  C                   MOVE SLSTM     SLSM                                          10/13/87
   31700  C                   MOVE SLSTY     SLSY                                          10/13/87
   31800  C                   MOVE CSTTM     CSTM                                          10/13/87
   31900  C                   MOVE PROFM     PROFIT                                        10/13/87
   32000  C                   MOVE CSTTY     CSTY                                          10/13/87
5728RG1 R01M02  881028            IBM AS/400 RPG/400                ICFLIB/RSFMUL        03/20/89  15:45:25      Page        9
SEQUENCE                                                                      IND   DO    LAST        PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE   NUM   UPDATE      LINE   ID
   32100  C                   WRITEITMSC3                         ITEM DTL 2                 10/13/87
   32200  C                   GOTO READRQ                                                    10/13/87
   32300  C*****************************************************************               10/13/87
   32400  C*                                                                               10/13/87
   32500  C*                      CUSTOMER INQUIRY                                         10/13/87
   32600  C*                                                                               10/13/87
   32700  C*    THE REQUEST FROM THE CUSTOMER INQUIRY MENU IS PROCESSED.                   10/13/87
   32800  C*                                                                               10/13/87
   32900  C*    IF CMD 1 (*IN99) IS PRESSED, THE PROGRAM IS ENDED.                         10/13/87
   33000  C*                                                                               10/13/87
   33100  C*    IF CMD 2 (*IN98) IS PRESSED, THE CUSTOMER INQUIRY IS ENDED,                10/13/87
   33200  C*    AND THE MAIN MENU (CIMENU) IS WRITTEN TO THE SCREEN.                       10/13/87
   33300  C*                                                                               10/13/87
   33400  C*    IF A CUSTOMER NUMBER IS ENTERED, THE CUSTOMER INQUIRY                      10/13/87
   33500  C*    REQUEST IS SENT TO THE REMOTE SYSTEM.                                      10/13/87
   33600  C*                                                                               10/13/87
   33700  C*    A READ TO THE PROGRAM DEVICE IS ISSUED TO RECEIVE THE                      03/20/89
   33800  C*    INFORMATION FROM THE TARGET PROGRAM.                                       10/13/87
   33900  C*                                                                               10/13/87
   34000  C*    IF AN ERROR OCCURS, THE ERROR IS PRINTED AND THE JOB IS                    10/13/87
   34100  C*    ENDED.                                                                     10/13/87
   34200  C*                                                                               10/13/87
   34300  C*****************************************************************               10/13/87
   34400  C* 9                                                                            03/20/89
   34500  C           DTLIN   TAG                                                        10/13/87
   34600  C           *IN99   CABEQ'1'       END                 JOB ENDS                10/13/87
   34700  C           *IN98   IFEQ '1'                                                B001   10/13/87
   34800  C                   WRITECIMENU                         MAIN MENU           001   10/13/87
   34900  C                   GOTO READRQ                                             001   10/13/87
   35000  C                   END                                                     E001   10/13/87
   35100  C           EVDTL   TAG                                                        10/13/87
   35200  C                   MOVE CUSTNO    ITEMNO                                      10/13/87
   35300  C                   MOVEL'ICF00    'CMID                                       10/13/87
   35400  C                   EXCPTITEMRQ                         SEND CUST #            10/13/87
   35500  C           MAJCOD  CABGE'04'      ERROR               ERROR RTN              10/13/87
   35600  C           TRY88   TAG                                                        10/13/87
   35700  C                   MOVEL'         'CMID                                       10/13/87
   35800  C                   MOVEL' '       RECCUS              RESET RECID            10/13/87
   35900  C                   SETOF                         88                  1        10/13/87
   36000  C                   READ CMNFIL                     8810REC CUS INF       2 3  10/13/87
   36100  C   88              EXSR ERRCHK                         FILE ERR?              10/13/87
   36200  C           MAJMIN  CABGE'0300'    EVDTL               NODATATRYAGN           10/13/87
   36300  C  N80              EXCPTNOTCUS                         RECD NOT FD?           10/13/87
   36400  C           RECCUS  CABNE'C'       RECERR              PRINT MSG              10/13/87
   36500  C                   SETOF                         80                  1        10/13/87
   36600  C*****************************************************************               10/13/87
   36700  C*                                                                               10/13/87
   36800  C*                      PROCESS CUSTOMER INFORMATION                             10/13/87
   36900  C*                                                                               10/13/87
   37000  C*     THE CUSTOMER DATA RECEIVED FROM THE TARGET PROGRAM IS                     03/20/89
   37100  C*     PROCESSED. IF CUSTOMER NUMBER IS ZERO OR LESS, IT IS AN                   03/20/89
   37200  C*     INVALID REQUEST AND THE MAIN MENU IS WRITTEN TO THE SCREEN.               03/20/89
   37300  C*                                                                              10/14/87
```

Figure 11-23 (Part 7 of 14). Source Program Example — RSFMUL (System-Supplied Formats)

```
   37400 C*     WHEN THE RCVTRNRND INDICATOR(*IN90) IS RECEIVED, THE                    03/20/89
5728RG1 R01M02  881028            IBM AS/400 RPG/400                ICFLIB/RSFMUL      03/20/89  15:45:25      Page      10
SEQUENCE                                                                    IND   DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE   NUM   UPDATE     LINE   ID
   37500 C*     CUSTOMER INFORMATION IS WRITTEN TO THE SCREEN.                          03/20/89
   37600 C*                                                                            10/14/87
   37700 C*     IF DURING THE READ OPERATION AN ERROR IS RECEIVED,                      10/13/87
   37800 C*     CONTROL GOES TO THE ERROR ROUTINE TO END THE JOB.                       10/13/87
   37900 C*                                                                            10/13/87
   38000 C*****************************************************************          10/13/87
   38100 C* 10                                                                        10/13/87
   38200 C           DTOUT     TAG                                                     10/13/87
   38300 C           CUSTNO    IFEQ 000000                                    B001     10/13/87
   38400 C                     SETOF                       66                 3  001     10/13/87
   38500 C                     WRITECIMENU                   MAIN MENU           001     10/13/87
   38600 C                     GOTO READRQ                                       001     10/13/87
   38700 C                     END                                               E001    10/13/87
   38800 C                     MOVE CUSTNO    CUSTN                                      10/13/87
   38900 C                     MOVELDNAME     CNAME                                      10/13/87
   39000 C                     MOVE DLSTOR    DLSTR                                      10/13/87
   39100 C                     MOVE DSLSTM    DSLSM                                      10/13/87
   39200 C                     MOVE DSPM01    DSPM1                                      10/13/87
   39300 C                     MOVE DSPM02    DSPM2                                      10/13/87
   39400 C                     MOVE DSTTYD    DSTYD                                      10/13/87
   39500 C                     MOVE IDEPT     DEPT                                       10/13/87
   39600 C                     WRITEDTLSCR                   DETAIL INFO                10/13/87
   39700 C                     GOTO READRQ                                              10/13/87
   39800 C*****************************************************************          10/13/87
   39900 C*                                                                            10/13/87
   40000 C*    THIS ROUTINE HANDLES THE USER'S REQUEST FOLLOWING THE                   10/14/87
   40100 C*    DISPLAY OF THE CUSTOMER INFORMATION.  CMD KEY 1 WILL END                03/20/89
   40200 C*    THE JOB, CMD KEY 2 WILL DISPLAY THE MAIN MENU, AND "ENTER"              03/20/89
   40300 C*    WILL BRING UP THE CUSTOMER INQUIRY MENU.                                03/20/89
   40400 C*                                                                            10/13/87
   40500 C*****************************************************************          10/13/87
   40600 C* 11                                                                        10/13/87
   40700 C           DTLRTN    TAG                                                     10/13/87
   40800 C           *IN99     CABEQ'1'       END            JOB ENDS                 10/13/87
   40900 C           *IN98     IFEQ '1'                                       B001     10/13/87
   41000 C                     WRITECIMENU                   MAIN MENU           001     10/13/87
   41100 C                     GOTO READRQ                                       001     10/13/87
   41200 C                     END                                               E001    10/13/87
   41300 C                     WRITEDTLMNU                   CUSTOMER INQ                10/13/87
   41400 C                     GOTO READRQ                                              10/13/87
   41500 C*****************************************************************          10/13/87
   41600 C*                                                                            10/13/87
   41700 C*    WHEN AN I/O OPERATION ERROR IS DETECTED, A MESSAGE IS PRINTED           03/20/89
   41800 C*    AND THE TRANSACTION AND SESSION ARE ENDED FOR EACH OF THE               03/20/89
   41900 C*    REMOTE SYSTEMS.                                                         03/20/89
   42000 C*                                                                            10/13/87
   42100 C*****************************************************************          10/13/87
   42200 C* 12                                                                        10/13/87
   42300 C           RECERR    TAG                                                     10/13/87
   42400 C                     EXCPTRECER                    WRONG RECID               10/13/87
   42500 C                     GOTO END                      END PROGRAM               10/13/87
   42600 C           ERROR     TAG                                                     10/13/87
   42700 C                     EXCPTMMERR                                              10/13/87
   42800 C           END       TAG                                                     10/13/87
5728RG1 R01M02  881028            IBM AS/400 RPG/400                ICFLIB/RSFMUL      03/20/89  15:45:25      Page      11
SEQUENCE                                                                    IND   DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE   NUM   UPDATE     LINE   ID
   42900 C                     MOVEL'ICF00    'CMID                                      10/13/87
   43000 C                     EXCPTDETACH                   DET 1ST TRANS              10/13/87
   43100 C                     MOVEL'ICF01    'CMID                                      10/13/87
   43200 C                     EXCPTDETACH                   DET 2ND TRANS              10/13/87
   43300 C                     MOVEL'ICF02    'CMID                                      10/13/87
   43400 C                     EXCPTDETACH                   DET 3RD TRANS              10/13/87
   43500 C                     MOVEL'ICF03    'CMID                                      10/13/87
```

Figure 11-23 (Part 8 of 14). Source Program Example — RSFMUL (System-Supplied Formats)

```
43600  C                    EXCPTDETACH                        DET 4TH TRANS                10/13/87
43700  C           ABORT    TAG                                                             10/13/87
43800  C           'ICF00   'REL  CMNFIL                 86  REL 1ST SESS      2          10/13/87
43900  C           'ICF01   'REL  CMNFIL                 86  REL 2ND SESS      2          10/13/87
44000  C           'ICF02   'REL  CMNFIL                 86  REL 3RD SESS      2          10/13/87
44100  C           'ICF03   'REL  CMNFIL                 86  REL 4TH SESS      2          10/13/87
44200  C           FORCE    TAG                                                             10/13/87
44300  C                    SETON                     LR                    1               10/13/87
44400  C                    RETRN                                                           10/13/87
44500  C*****************************************************************                10/13/87
44600  C*                                                                                   10/13/87
44700  C*    THIS SUBROUTINE IS CALLED TO EVOKE THE TARGET PROGRAM. THE                     03/20/89
44800  C*    SAME TARGET PROGRAM (ICFLIB/RTDMULCL) IS EVOKED AT FOUR                        03/20/89
44900  C*    DIFFERENT REMOTE SYSTEMS.  THE PROGRAM DEVICE IDENTIFIES                       03/20/89
45000  C*    WHICH SESSION SHOULD BE EVOKED.  THE PROGRAM DEVICE WAS                        03/20/89
45100  C*    SPECIFIED IN CMID PRIOR TO CALLING THIS ROUTINE.                               03/20/89
45200  C*                                                                                   10/13/87
45300  C*****************************************************************                10/13/87
45400  C* 13                                                                               10/13/87
45500  C           EVKSR    BEGSR                                                          10/13/87
45600  C                    EXCPTEVOKE                         EVOKE TARGET                10/13/87
45700  C                    ENDSR                                                          10/13/87
45800  C*****************************************************************                10/13/87
45900  C*                                                                                   10/13/87
46000  C*    THIS SUBROUTINE IS CALLED TO PERFORM FURTHER CHECKS ON FILE                    03/20/89
46100  C*    ERRORS RESULTING FROM THE READ OPERATION ISSUED TO THE PRO-                    03/20/89
46200  C*    GRAM DEVICE. THIS ROUTINE CHECKS FOR THE TIME OUT INDICATION.                  03/20/89
46300  C*    IF THERE IS A TIME OUT, THEN A MESSAGE IS SENT TO THE USER'S                   03/20/89
46400  C*    DISPLAY SCREEN REQUESTING ACTION, OTHERWISE PROGRAM ENDS.                      10/16/87
46500  C*                                                                                   10/13/87
46600  C*****************************************************************                10/13/87
46700  C* 14                                                                               10/13/87
46800  C           ERRCHK   BEGSR                                                          10/13/87
46900  C           MAJMIN   IFEQ '0310'                        TIMER EXPD?          B001  10/13/87
47000  C           CHKAGN   TAG                                                      001   10/13/87
47100  C                    WRITETIMOUT                        DISPLAY MSG          001   10/13/87
47200  C                    READ DSPFIL                       87READ REPLY        3  001   10/13/87
47300  C   88      TIMRSP   CABEQ'1'       TRY88              CUST INQUIR          001   10/13/87
47400  C   89      TIMRSP   CABEQ'1'       TRY89              ITEM INQUIR          001   10/13/87
47500  C           TIMRSP   IFEQ '2'                          END PROGRAM          B002  10/13/87
47600  C                    EXCPTEOS                           END SESSION          002   10/13/87
47700  C                    GOTO FORCE                         END PROGRAM          002   10/13/87
47800  C                    END                                                     E002  10/13/87
47900  C                    GOTO CHKAGN                        ASK AGAIN            001   10/13/87
48000  C                    END                                                     E001  10/13/87
48100  C           MAJMIN   CABEQ'0300'    ERRESR             TRN/NODATA                 10/13/87
48200  C           MAJMIN   CABEQ'0000'    ERRESR             TRN W/DATA                 10/13/87
5728RG1 R01M02  881028          IBM AS/400 RPG/400                ICFLIB/RSFMUL        03/20/89  15:45:25      Page      12
SEQUENCE                                                                    IND   DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE   NUM   UPDATE     LINE   ID
48300  C                    GOTO ERROR                         ABEND                      10/13/87
48400  C           ERRESR   TAG                                                             10/13/87
48500  C                    ENDSR                                                          10/13/87
48600  C*****************************************************************                10/14/87
48700  C*                                                                                   10/14/87
48800  C*    THIS IS THE PROGRAM ERROR SUBROUTINE THAT RECEIVES CONTROL                     03/20/89
48900  C*    WHEN AN ERROR OCCURS AFTER AN I/O OPERATION IS ISSUED TO THE                   03/20/89
49000  C*    PROGRAM DEVICE AND THERE IS A NON-ZERO VALUE IN THE RPG                        03/20/89
49100  C*    STATUS FIELD (ERR). THIS ROUTINE CHECKS FOR STATUS VALUES                      03/20/89
49200  C*    THAT RELATE TO ICF OPERATIONS. IF THE PROGRAM DEVICE                           03/20/89
49300  C*    IS ALREADY ACQUIRED, THE ERROR IS IGNORED, OTHERWISE THE                       03/20/89
49400  C*    PROGRAM IS ENDED.                                                              03/20/89
49500  C*                                                                                   10/14/87
49600  C*****************************************************************                10/14/87
49700  C* 15                                                                               10/14/87
49800  C           *PSSR    BEGSR                                                          10/14/87
49900  C                    MOVE '      '  RETURN  6          DEFAULT                    10/14/87
50000  C           ERR      CABEQ01285     ENDPSR             ALREADY ACQ?               10/14/87
```

Figure 11-23 (Part 9 of 14). Source Program Example — RSFMUL (System-Supplied Formats)

```
    50100 C                   MOVE '*CANCL'  RETURN          JOB ENDS                  10/14/87
    50200 C           ENDPSR  ENDSRRETURN                   BACK TO MAIN              10/14/87
    50300 C*************************************************************              10/13/87
    50400 OQPRINT  E 1              MMERR                                               10/13/87
    50500 O                                    21 'COMMUNICATION ERROR.'              10/13/87
    50600 O                                    34 'MAJOR/MINOR:'                      10/13/87
    50700 O                         MAJCOD     37                                     10/13/87
    50800 O                                    38 '/'                                 10/13/87
    50900 O                         MINCOD     40                                     10/13/87
    51000 O                                    49 'FORMAT:'                           10/13/87
    51100 O                         FMTNM      60                                     10/13/87
    51200 O                                    69 'PGMDEV:'                           10/13/87
    51300 O                         CMID       80                                     10/13/87
    51400 O         E 1             RECER                                             10/13/87
    51500 O                                    22 'UNMATCH RECORD FORMAT'             10/13/87
    51600 O                                    34 '-JOB ENDED.'                       10/13/87
    51700 O         E 1             NOTITM                                            10/13/87
    51800 O                                    21 'NOT ITEM RECD-'                    10/13/87
    51900 O                         ITEMNO     28                                     10/13/87
    52000 O                                    29 '/'                                 10/13/87
    52100 O                         DESC       60                                     10/13/87
    52200 O         E 1             NOTCUS                                            10/13/87
    52300 O                                    21 'NOT CUST RECD-'                    10/13/87
    52400 O                         CUSTNO     28                                     10/13/87
    52500 O                                    29 '/'                                 10/13/87
    52600 O                         DNAME      60                                     10/13/87
    52700 OCMNFIL  E                EVOKE                                             10/13/87
    52800 O                                    K8 '$$EVOKNI'                          10/13/87
    52900 O                                     8 'RTFMULCL'                          10/13/87
    53000 O                                    32 'ICFLIB  '                          10/13/87
    53100 O         E               ITEMRQ                                            10/13/87
    53200 O                                    K6 '$$SEND'                            10/13/87
    53300 O                                     4 '0006'                              10/13/87
    53400 O                         ITEMNO     10                                     10/13/87
5728RG1 R01M02  881028            IBM AS/400 RPG/400                 ICFLIB/RSFMUL    03/20/89  15:45:25      Page      13
SEQUENCE                                                                    IND   DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE   NUM   UPDATE     LINE   ID
    53500 O         E               DETACH                                            10/13/87
    53600 O                                    K8 '$$SENDET'                          10/13/87
    53700 O                                     4 '0000'                              10/13/87
    53800 O         E               EOS                                               10/13/87
    53900 O                                    K5 '$$EOS'                             10/13/87
    54000 O                                     4 '0000'                              10/13/87
* 6103     54001   OVERFLOW INDICATOR OA ASSIGNED TO FILE QPRINT.
  H000000   OUTPUT FIELDS FOR RECORD CIMENU FILE DSPFIL FORMAT CIMENU.
  H000000         MENU FOR INQUIRY
  I000000   OUTPUT FIELDS FOR RECORD DTLMNU FILE DSPFIL FORMAT DTLMNU.
  I000000         CUSTOMER INQUIRY SCREEN 1
  J000000   OUTPUT FIELDS FOR RECORD DTLSCR FILE DSPFIL FORMAT DTLSCR.
  J000000         CUSTOMER INQUIRY SCR. #2
  J000001                           CUSTN      6  CHAR    6
  J000002                           DEPT       9  ZONE  3,0
  J000003                           DLSTR     15  ZONE  6,0
  J000004                           DSLSM     24  ZONE  9,0
  J000005                           DSPM1     33  ZONE  9,0
  J000006                           DSPM2     42  ZONE  9,0
  J000007                           DSPM3     51  ZONE  9,0
  J000008                           DSTYD     62  ZONE 11,0
  J000009                           NAME      67  CHAR    5
  K000000   OUTPUT FIELDS FOR RECORD ITMMNU FILE DSPFIL FORMAT ITMMNU.
  K000000         ITEM INQUIRY SCREEN ONE
  L000000   OUTPUT FIELDS FOR RECORD ITMSC2 FILE DSPFIL FORMAT ITMSC2.
  L000000         ITEM INQUIRY SCREEN TWO
  L000001                           DSC       30  CHAR   30
  L000002                           QAVAIL    37  ZONE  7,0
  L000003                           QTYH      44  ZONE  7,0
  L000004                           QTYO      51  ZONE  7,0
  L000005                           QTYB      58  ZONE  7,0
```

Figure 11-23 (Part 10 of 14). Source Program Example — RSFMUL (System-Supplied Formats)

```
L000006                          UNT       60  CHAR    2
L000007                          PR1       67  ZONE  7,2
L000008                          PR5       74  ZONE  7,0
L000009                          UFR       79  ZONE  5,2
M000000   OUTPUT FIELDS FOR RECORD ITMSC3 FILE DSPFIL FORMAT ITMSC3.
M000000         ITEM INQUIRY SCREEN 3
M000001                          SLSM       9  ZONE  9,2
M000002                          SLSY      20  ZONE 11,2
M000003                          CSTM      29  ZONE  9,2
M000004                          CSTY      40  ZONE 11,2
M000005                          PROFIT    45  ZONE  5,2
M000006                          LOSTS     54  ZONE  9,2
N000000   OUTPUT FIELDS FOR RECORD TIMOUT FILE DSPFIL FORMAT TIMOUT.
N000000         TIME OUT SCREEN
        * * * * *  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
* 7089      4500   RPG PROVIDES SEPARATE INDICATOR AREA FOR FILE CMNFIL.
5728RG1 R01M02  881028            IBM AS/400 RPG/400                ICFLIB/RSFMUL       03/20/89  15:45:25      Page      14
                      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  CMNFIL      WORKSTN       4500D    6500     7700    12900    13000
                                 13100    13200    23000    36000    43800
                                 43900    44000    44100    52700    53100
                                 53500    53800
         $$EOS                   53900
         $$EVOKNI                52800
         $$SEND                  53200
         $$SENDET                53600
   02  DSPFIL      WORKSTN        5000D   15600    47200
         CIMENU                   5000D A000000    14200    16300    20900
                                 29800    34800    38500    41000  H000000
         DTLMNU                   5000D B000000    18200    41300  I000000
         DTLSCR                   5000D C000000    39600  J000000
         ITMMNU                   5000D D000000    18000    25100    30300
                                 30800  K000000
         ITMSC2                   5000D E000000    26600  L000000
         ITMSC3                   5000D F000000    32100  M000000
         TIMOUT                   5000D G000000    47100  N000000
   03  QPRINT      PRINTER        5200D   50400    51400    51700    52200
                                 54001
Field References:
       FIELD       ATTR     REFERENCES (M=MODIFIED D=DEFINED)
       *IN97       A(1)     A000001  B000001  C000001  D000001  E000001
                            F000001  G000001    30100
       *IN98       A(1)     A000002  B000002  C000002  D000002  E000002
                            F000002  G000002    20800    29700    34700
                              40900
       *IN99       A(1)     A000003  B000003  C000003  D000003  E000003
                            F000003  G000003    17800    20700    29600
                              34600    40800
       *PSSR       BEGSR     4500    49800D
* 7031  ABORT       TAG      43700D
* 7031  BEGIN       TAG      12800D
       CHKAGN      TAG      47000D   47900
       CMID        A(10)    10600D   13300M   13500M   13700M   13900M
                            21600M   21900M   22200M   22900M   35300M
                            35700M   42900M   43100M   43300M   43500M
                            51300
       CSTM        P(9,2)   31800M M000003D
       CSTTM       P(9,2)    9100D   31100    31800
       CSTTY       P(11,2)   9200D   32000
       CSTY        P(11,2)  32000M M000004D
5728RG1 R01M02  881028            IBM AS/400 RPG/400                ICFLIB/RSFMUL       03/20/89  15:45:25      Page      15
       CUSTN       A(6)     38800M J000001D
       CUSTNO      P(6,0)    6700D B000004D   35200    38300    38800
                            52400
       DEPT        P(3,0)   39500M J000002D
```

Figure 11-23 (Part 11 of 14). Source Program Example — RSFMUL (System-Supplied Formats)

```
          DESC       A(30)      8000D   25800    52100
          DETACH     EXCPT     43000    43200    43400    43600    53500
          DLSTOR     A(6)       6900D   39000
          DLSTR      P(6,0)    39000M J000003D
          DNAME      A(30)      6800D   38900    52600
          DSC        A(30)     25800M L000001D
          DSLSM      P(9,0)    39100M J000004D
          DSLSTM     A(9)       7000D   39100
          DSPM01     A(9)       7100D   39200
          DSPM02     A(9)       7200D   39300
 * 7031   DSPM03     A(9)       7300D
          DSPM1      P(9,0)    39200M J000005D
          DSPM2      P(9,0)    39300M J000006D
          DSPM3      P(9,0)   J000007D
          DSTTYD     A(11)      7400D   39400
          DSTYD      P(11,0)   39400M J000008D
          DTLIN      TAG       16100    34500D
          DTLRTN     TAG       16200    40700D
 * 7031   DTOUT      TAG       38200D
          END        TAG       17800    20700    29600    34600    40800
                               42500    42800D
          ENDPSR     ENDSR     50000    50200D
          EOS        EXCPT     47600    53800
          ERR        Z(5,0)    10300D   50000
          ERRCHK     BEGSR     23100    36100    46800D
          ERRESR     TAG       48100    48200    48400D
          ERROR      TAG       22500    23500    35500    42600D   48300
          EVDTL      TAG       35100D   36200
          EVKSR      BEGSR     13400    13600    13800    14000    45500D
          EVOKE      EXCPT     45600    52700
 * 7031   FILL01     A(240)     9800D
 * 7031   FILL02     A(145)    10000D
 * 7031   FILL03     A(240)    10400D
 * 7031   FILL04     A(145)    11100D
 * 7031   FILL10     A(11)      9500D
 * 7031   FILL20     A(57)      7600D
          FMTNM      A(8)      10500D   51100
          FORCE      TAG       44200D   47700
          IDEPT      A(3)       7500D   39500
          IODS       DS(415)    5000     9700D
          IOFB       DS(415)    4500    10100D
          ITEMNO     P(6,0)     7900D D000004D   21200    21300    21400
                               25000    35200M   51900    53400
          ITEMRQ     EXCPT     22400    35400    53100
          ITMIN      TAG       15800    20600D   23200
 * 7031   ITMOUT     TAG       24900D
          ITMRTN     TAG       15900    16000    29500D
 * 7031   LOC        A(8)      10200D
 * 7031   LOS        P(9,2)     9400D
          LOSTS      P(9,2)    31500M M000006D
 * 7031   MAIN       TAG       14100D
          MAJCOD     A(2)      10800D   22500    23500    35500    50700
          MAJMIN     A(4)      10700D   23200    36200    46900    48100
 5728RG1 R01M02  881028          IBM AS/400 RPG/400                ICFLIB/RSFMUL        03/20/89  15:45:25      Page      16
                               48200
          MENU       TAG       15700    17700D
          MINCOD     A(2)      10900D   50900
          MMERR      EXCPT     42700    50400
          NAME       A(5)      38900M J000009D
          NOTCUS     EXCPT     36300    52200
          NOTITM     EXCPT     23300    51700
          OPTION     A(1)     A000004D  17900
 * 7031   PRO        P(5,2)     9300D
          PROFIT     P(5,2)    31900M M000005D
          PROFM      P(9,2)    31100D   31200M   31400    31400M   31900
          PR01       P(7,2)     8600D   26300    31500
          PR05       P(6,0)     8700D   26400
          PR1        P(7,2)    26300M L000007D
```

Figure 11-23 (Part 12 of 14). Source Program Example — RSFMUL (System-Supplied Formats)

```
        PR5         P(7,0)    26400M L000008D
        QAVAIL      P(7,0)    25400D   25500M   25600M   25700M L000002D
        QTYB        P(7,0)    26100M L000005D
        QTYBO       P(7,0)     8400D   25700    26100
        QTYH        P(7,0)    26000M L000003D
        QTYLST      P(7,0)     8100D   31500
        QTYO        P(7,0)    25900M L000004D
        QTYOH       P(7,0)     8200D   25500    26000
        QTYOO       P(7,0)     8300D   25600    25900
        READRQ      TAG       15500D   16400    18400    21000    25200
                              26700    29900    30400    30900    32200
                              34900    38600    39700    41100    41400
        RECCUS      A(1)       6600D   35800M   36400
        RECER       EXCPT     42400    51400
        RECERR      TAG       23600    36400    42300D
        RECID       A(8)       9900D   15700    15800    15900    16000
                              16100    16200    30200    30700
* 7031  RECID2      A(8)      11000D
        RECITM      A(1)       7800D   22800M   23600
        RETURN      A(6)      49900D   50100M   50200
        SLSM        P(9,2)    31600M M000001D
        SLSTM       P(9,2)     8900D   31100    31300    31400    31600
        SLSTY       P(11,2)    9000D   31700
        SLSY        P(11,2)   31700M M000002D
        TIMRSP      A(1)    G000004D   47300    47400    47500
        TRY88       TAG       35600D   47300
        TRY89       TAG       22600D   47400
        UFR         P(5,2)    26500M L000009D
        UFRT        P(5,2)     8800D   26500
        UNIT        A(2)       8500D   26200
        UNT         A(2)      26200M L000006D
        XICF01      TAG       21200    21500D
        XICF02      TAG       21300    21800D
        XICF03      TAG       21400    22100D
        XITMIN      TAG       21700    22000    22300D
        '       '   LITERAL   22900    35700
        '     '     LITERAL   49900
        ' '         LITERAL   22800    35800
        '*CANCL'    LITERAL   50100
        'C'         LITERAL   36400
        'CIMENU  '  LITERAL   15700
        'DTLMNU  '  LITERAL   16100
        'DTLSCR  '  LITERAL   16200
5728RG1 R01M02  881028          IBM AS/400 RPG/400                ICFLIB/RSFMUL        03/20/89  15:45:25      Page      17
        'I'         LITERAL   23600
        'ICF00   '  LITERAL   12900    13300    35300    42900    43800
        'ICF01   '  LITERAL   13000    13500    21600    43100    43900
        'ICF02   '  LITERAL   13100    13700    21900    43300    44000
        'ICF03   '  LITERAL   13200    13900    22200    43500    44100
        'ITMMNU  '  LITERAL   15800
        'ITMSC2  '  LITERAL   15900    30200
        'ITMSC3  '  LITERAL   16000    30700
        '0000'      LITERAL   48200
        '0300'      LITERAL   23200    36200    48100
        '0310'      LITERAL   46900
        '04'        LITERAL   22500    23500    35500
        '1'         LITERAL   17800    17900    20700    20800    29600
                              29700    30100    34600    34700    40800
                              40900    47300    47400
        '2'         LITERAL   47500
        0           LITERAL   25400    31300
        000000      LITERAL   25000    38300
        01285       LITERAL   50000
        100         LITERAL   31200
        399999      LITERAL   21200
        699999      LITERAL   21300
        899999      LITERAL   21400
```

Figure 11-23 (Part 13 of 14). Source Program Example — RSFMUL (System-Supplied Formats)

```
Indicator References:
       INDICATOR  REFERENCES (M=MODIFIED D=DEFINED)
       *IN        A000001  A000002  A000003  B000001  B000002  B000003
                  C000001  C000002  C000003  D000001  D000002  D000003
                  E000001  E000002  E000003  F000001  F000002  F000003
                  G000001  G000002  G000003    17800    20700    20800
                    29600    29700    30100    34600    34700    40800
                    40900
       LR           44300M
       OA            5200D   54001
* 7031 10           23000M   36000M
       46           31300M   31400
* 7031 66           38400M
       80            6500M   36300    36500M
       81            7700M   23300    23400M
* 7031 86           43800M   43900M   44000M   44100M
* 7031 87           47200M
       88           15600M   35900M   36000M   36100    47300
       89           22700M   23000M   23100    47400
       97         A000001  B000001  C000001  D000001  E000001  F000001
                  G000001    30100
       98         A000002  B000002  C000002  D000002  E000002  F000002
                  G000002    20800    29700    34700    40900
       99         A000003  B000003  C000003  D000003  E000003  F000003
                  G000003    17800    20700    29600    34600    40800
     * * * * *  E N D  O F  C R O S S  R E F E R E N C E  * * * * *
 5728RG1 R01M02  881028            IBM AS/400 RPG/400                ICFLIB/RSFMUL       03/20/89  15:45:25      Page      18
 5728RG1 R01M02  881028            IBM AS/400 RPG/400                ICFLIB/RSFMUL       03/20/89  15:45:25      Page      19
                       M e s s a g e  S u m m a r y
* QRG6103 Severity:  00   Number:    1
          Message . . . . :   No Overflow Indicator is specified but an
            indicator is assigned to a file and automatic skip to 6 is
            generated.
* QRG7031 Severity:  00   Number:   20
          Message . . . . :   The Name or indicator is not referenced.
* 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 R01M02  881028            IBM AS/400 RPG/400                ICFLIB/RSFMUL       03/20/89  15:45:25      Page      20
                         F i n a l  S u m m a r y
 Message Count:  (by Severity Number)
            TOTAL    00     10     20     30     40     50
             22      22      0      0      0      0      0
 Program Source Totals:
    Records . . . . . . . . . . :   540
    Specifications  . . . . . . :   296
    Table Records . . . . . . . :   0
    Comments  . . . . . . . . . :   244
 PRM has been called.
 Program RSFMUL is placed in library ICFLIB. 00 highest Error-Severity-Code.
         * * * * *  E N D  O F  C O M P I L A T I O N  * * * * *
```

Figure 11-23 (Part 14 of 14). Source Program Example — RSFMUL (System-Supplied Formats)

Target Program Multiple-Session Inquiry (Example II)

The following describes a target program for the multiple-session inquiry.

Program Files: The RPG/400 multiple-session example target program uses the following files:

CFILE An ICF file used to send records to and receive records from the source program.

PFILE A database file used to retrieve the requested information to send to the source program.

QPRINT A printer file used to print error messages resulting from communications errors.

DDS Source: The DDS source for the ICF file (CFILE) is illustrated in Figure 11-24.

```
5714PW1 R01M00  880301                SEU SOURCE LISTING                      10/14/87 17:20:35              PAGE 1
SOURCE FILE . . . . . .  QICFPUB/ICFLIB
MEMBER  . . . . . . . .  CFILE
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
  100      A*****************************************************************                                  10/14/87
  200      A*                                                               *                                  10/14/87
  300      A*                   ICF FILE                                    *                                  10/14/87
  400      A*            USED IN TARGET MULTIPLE SESSION PROGRAM            *                                  10/14/87
  500      A*                                                               *                                  10/14/87
  600      A*****************************************************************                                  10/14/87
  700      A                                    INDARA                                                         08/04/87
  800      A  05                                RQSWRT                                                         08/04/87
  900      A  10                                ALWWRT                                                         08/04/87
 1000      A                                    INDTXT(10 '10 END TRANS.')                                     08/04/87
 1100      A  15                                EOS                                                            08/04/87
 1200      A  20                                FAIL                                                           08/06/87
 1300      A                                    INDTXT(20 '20 F ABORT ST')                                     08/06/87
 1400      A                                    RCVFAIL(25 'RECEIVED FAIL')                                    08/04/87
 1500      A  30                                DETACH                                                         08/06/87
 1600      A                                    INDTXT(30 '30>DETACH TGT')                                     08/06/87
 1700      A                                    RCVDETACH(44 'RECV DETACH')                                    08/04/87
 1800      A          R SNDPART                                                                              08/04/87
 1900      A                                    INVITE                                                         08/14/87
 2000      A            RECTYP         1                                                                     10/01/87
 2100      A            ITEMNO         6                                                                     10/08/87
 2200      A            EDATA        130                                                                     08/04/87
 2300      A            FILL1         13                                                                     10/08/87
 2400      A          R RCVPART                                                                              08/04/87
 2500      A            RECID2         6                                                                     10/08/87
 2600      A            PARTDS        80                                                                     10/08/87
 2700      A            FILL4         64                                                                     08/04/87
 2800      A          R RCVTRND                                                                              08/07/87
 2900      A                                    RCVTRNRND(40 'END OF TRN')                                     08/07/87
                              * * * *  E N D  O F  S O U R C E  * * * *
```

Figure 11-24. DDS Source for ICF File Used by Target Program Multiple-Session Inquiry

The DDS for the database file (PFILE) is illustrated in Figure 11-25.

```
5714PW1 R01M00  880301                SEU SOURCE LISTING                    10/16/87 07:43:14          PAGE 1
SOURCE FILE . . . . . . . QICFPUB/ICFLIB
MEMBER  . . . . . . . . . PFILE
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
  100      A                                LIFO                                                    07/02/87
  200      A          R DBREC                                                                       05/06/87
  300      A            RECCUS          1                                                         10/01/87
  400      A            DBSEQ           6                                                         08/18/87
  500      A            DBDATA        130                                                         07/02/87
  600      A            DBFILL         13                                                         10/01/87
  700      A          K DBSEQ                                                                       07/04/87
                          * * * *  E N D  O F  S O U R C E  * * * *
```

Figure 11-25. DDS Source for Database File Used by Target Program Multiple Session Inquiry

ICF File Creation and Program Device Entry Definition: The command needed to create the ICF file is:

```
CRTICFF  FILE(ICFLIB/CFILE)  SRCFILE(ICFLIB/QICFPUB)  SRCMBR(CFILE)
   ACQPGMDEV(RQSDEV)  TEXT("TARGET ICF FILE FOR MULTIPLE SESSION PROGRAM")
```

The command needed to define the program device entry is:

```
OVRICFDEVE PGMDEV(RQSDEV) RMTLOCNAME(*REQUESTER)
```

Program Explanation: The following explains the structure of the program examples illustrated in Figure 11-26 on page 11-81 and Figure 11-27 on page 11-86. The ICF file used in the first example is defined by the user, and uses externally described data formats (DDS). The second example uses the same file, but uses program-described data and system-supplied formats. The reference numbers in the explanation below correspond to the numbers in the program examples.

Although the basic structure of the two examples provided is the same, there are differences because of the way the user-defined formats and the system-supplied formats are used. All output operations to the ICF file in the first example are done using the WRITE operation. All output operations in the ICF file in the second example using system-supplied formats are done using the EXCPT operation.

Differences between the first and second example are described as notes in each of the following descriptions where necessary.

1 The file specification defines the files used in the program.

CFILE is the ICF file used to send records to and receive records from the source program.

The files used in the program are implicitly opened at the beginning of the RPG/400 cycle when the program starts.

Note: In the program using system-supplied formats, the input records for CFILE are explicitly coded in the program since CFILE is now described as a program-described file. The system-supplied file QICDMF can be used instead of CFILE. To use QICDMF, specify QICDMF in the file specification, or use an OVRICFF command to change the file name from CFILE to QICDMF.

The continuation lines on the file specification for CFILE define the data structure name — FEEDBK for the feedback area (INFDS). FEEDBK contains the following information, which is used to monitor for error conditions after an I/O operation is issued to CFILE:

- Record format-name (FMTNM)
- Program device name (PGMDEV)
- Major/minor return code (MAJMIN)

2 A read operation is issued to the program device to receive an inquiry request from the source program. If an error occurs on the read operation (a major code greater than 03), control passes to **5**.

If a detach indication is received, control goes to **6**. Otherwise, the program goes to **3**. When a detach is received, indicator 44 is set on, as defined by the RCVDETACH keyword in the DDS for the ICF file.

Note: In the program using system-supplied formats, a minor return code of 08 is checked to determine if a detach indication was received. Also, the read operation is issued using file name CFILE in factor 2, whereas in the user-supplied format example, it is issued using a record format name.

3 If a turnaround indication was not received in **2**, the program continues to read the ICF file until the indication is received.

If an error occurs (a major return code greater than 03 is returned from the read operation), the program goes to **5**. Otherwise, the program goes to **4**.

The program also tests to see whether the receive detach indicator (indicator 44) is set. If it is, the program goes to **6**.

Note: In the program using system-supplied formats, a minor return code 00 is checked to determine if change direction occurred and a minor return code of 08 for a detach indication received.

4 The program uses the requested number received from the source program to access the record from the database. The information retrieved from the database file (PFILE) is moved into the work area for the ICF file. A write operation is issued to the ICF program device using record format SNDPART. The write operation sends the requested information back to the source program.

If the requested number is not found, zero is propagated into the field.

If an error occurs on the write operation (a major return code greater than 03), control passes to **5**.

If no error occurs on the write, the program goes back to **2**.

Note: In the program using the system-supplied format, the write operation uses the $$SEND format to send the data.

5 When an error in an I/O operation is detected, an EXCPT operation is issued to print an error message saying that an error has occurred on the ICF file. The major/minor return code is also printed.

The program then goes to **6**.

6 Control passes here whenever the program has detected a communication error, or received a detach indication from the source program. The last record indicator is set on, which ends the program. CFILE is implicitly closed.

7 This subroutine is called for I/O operation errors that are not handled by subroutine **6**. This subroutine checks whether the program device is already acquired when an acquire operation is requested, and, if so, the second acquire is ignored. Otherwise, the program ends.

```
5728RG1 R01M02  881028            IBM AS/400 RPG/400                 ICFLIB/RTDMUL        03/20/89  15:45:02      Page        1
Compiler . . . . . . . . . . . . :   IBM AS/400 RPG/400
Command Options:
  Program  . . . . . . . . . . . :   ICFLIB/RTDMUL
  Source file  . . . . . . . . . :   ICFLIB/QICFPUB
  Source member  . . . . . . . . :   *PGM
  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
  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 . . . . . . . . . . . . . :  RTDMUL
  File . . . . . . . . . . . . . . :  QICFPUB
  Library  . . . . . . . . . . . . :  ICFLIB
  Last Change  . . . . . . . . . . :  03/20/89  15:38:05
  Description  . . . . . . . . . . :  RPG Multi-Session example w/DDS (target)
5728RG1 R01M02  881028            IBM AS/400 RPG/400                 ICFLIB/RTDMUL        03/20/89  15:45:02      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  H*****************************************************************                  10/13/87
     200  H*                                                                                  03/20/89
     300  H*  THIS PROGRAM WILL HANDLE THE REQUEST FOR EITHER A CUSTOMER                       10/13/87
     400  H*  NUMBER OR AN ITEM NUMBER.  THIS IS ACCOMPLISHED BY MAKING                       10/13/87
     500  H*  THE DATABASE FILE STRUCTURE (KEY LENGTH, KEY POSITION, RE CORD                  10/13/87
     600  H*  LENGTH, RECORD SIZE, ETC.) THE SAME FOR BOTH FILES WITH ONLY                    10/13/87
     700  H*  THE RECORD CONTENTS DIFFERENT.                                                 10/13/87
     800  H*                                                                                  10/13/87
     900  H*  THIS PROGRAM ENDS WHEN A DETACH REQUEST IS RECEIVED FROM                       10/13/87
    1000  H*  THE SOURCE PROGRAM.                                                           10/13/87
    1100  H*                                                                                  03/20/89
    1200  H*****************************************************************                  10/13/87
          H 1
    1300  FCFILE   CF  E                    WORKSTN                                         10/13/87
    1400  F                                            KINFDS FEEDBK                       10/13/87
    1500  F                                            KINFSR *PSSR                        10/14/87
           RECORD FORMAT(S):  LIBRARY ICFLIB FILE CFILE.
                    EXTERNAL FORMAT SNDPART RPG NAME SNDPART
                    EXTERNAL FORMAT RCVPART RPG NAME RCVPART
                    EXTERNAL FORMAT RCVTRND RPG NAME RCVTRND
    1600  FPFILE   IF  E           K        DISK                                            10/13/87
           RECORD FORMAT(S):  LIBRARY ICFLIB FILE PFILE.
                    EXTERNAL FORMAT DBREC RPG NAME DBREC
    1700  FQPRINT  O   F     132            PRINTER                                         10/13/87
    1800   *                                                                                 03/20/89
  A000000   INPUT  FIELDS FOR RECORD SNDPART FILE CFILE FORMAT SNDPART.
  A000001                                      1   1 RECTYP
  A000002                                      2   7 ITEMNO
  A000003                                      8 137 EDATA
  A000004                                    138 150 FILL1
  B000000   INPUT  FIELDS FOR RECORD RCVPART FILE CFILE FORMAT RCVPART.
  B000001                                      1   6 RECID2
  B000002                                      7  86 PARTDS
  B000003                                     87 150 FILL4
  C000000   INPUT  FIELDS FOR RECORD RCVTRND FILE CFILE FORMAT RCVTRND.
  D000000   INPUT  FIELDS FOR RECORD DBREC FILE PFILE FORMAT DBREC.
  D000001                                      1   1 RECCUS
  D000002                                      2   7 DBSEQ
```

Figure 11-26 (Part 1 of 5). Target Program Example — RTDMUL (User-Defined Formats)

```
 D000003                                    8 137 DBDATA
 D000004                                  138 150 DBFILL
    1900  IFEEDBK      DS                                                    10/13/87
    2000  I                                *ROUTINE LOC                       10/14/87
    2100  I                                *STATUS  ERR                       10/14/87
    2200  I                                  38  47 FMTNM                     10/13/87
    2300  I                                 273 282 PGMDEV                    10/13/87
5728RG1 R01M02  881028            IBM AS/400 RPG/400                ICFLIB/RTDMUL      03/20/89  15:45:02      Page        3
SEQUENCE                                                                      IND    DO   LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM  UPDATE     LINE   ID
    2400  I                                401 404 MAJMIN                     10/13/87
    2500  I                                401 402 MAJCOD                     10/13/87
    2600  I                                403 404 MINCOD                     10/13/87
    2700  C*****************************************************************  10/13/87
    2800  C*                                                                 10/13/87
    2900  C*  READ THE REQUEST FROM THE SOURCE PROGRAM.  INDICATOR 40         10/13/87
    3000  C*  INDICATES RCVTRNRND OCCURRED.  INDICATOR 44 INDICATES THAT      10/13/87
    3100  C*  DETACH HAS BEEN RECEIVED.                                       10/13/87
    3200  C*                                                                 10/13/87
    3300  C*  INDICATOR 99 WILL BE TURNED ON FOR "I/O ERRORS" THEREBY         10/13/87
    3400  C*  PREVENTING THE RPG DEFAULT ERROR HANDLER FROM BEING CALLED.    10/13/87
    3500  C*  THIS IS NECESSARY TO ALLOW THE PROGRAM TO PROCESS THE ICF       01/18/88
    3600  C*  MAJOR/MINOR RETURN CODE.  THIS PROGRAM CHECKS FOR ERRORS ON     10/13/87
    3700  C*  EVERY ICF FILE OPERATION.  A MAJOR CODE GREATER THAN 03         01/18/88
    3800  C*  INDICATES AN ERROR.                                            10/13/87
    3900  C*                                                                 10/13/87
    4000  C*****************************************************************  10/13/87
    4100  C* 2                                                               03/20/89
    4200  C           READ      TAG                                          10/13/87
    4300  C                     READ RCVPART              9950                2 3    10/13/87
    4400  C           MAJCOD    CABGT'03'      ERROR                         10/13/87
    4500  C           *IN44     CABEQ'1'       END             DET RECV?     10/13/87
    4600  C                     MOVE RECID2    DBSEQ                         10/13/87
    4700  C           MAJMIN    CABEQ'0000'    XMIT            RCVTRNRND?    10/13/87
    4800  C           *IN40     CABEQ'1'       XMIT            RCVTRNRND?    10/13/87
    4900  C* 3                                                               03/20/89
    5000  C           *IN40     DOWEQ'0'                       RCVTRNRND?          B001   03/20/89
    5100  C                     READ RCVTRND              9950                2 3  001    10/13/87
    5200  C           MAJCOD    CABGT'03'      ERROR                           001    10/13/87
    5300  C           *IN44     CABEQ'1'       END             DETACH RECV?    001    10/13/87
    5400  C                     END                                            E001   10/13/87
    5500  C*****************************************************************  10/13/87
    5600  C*                                                                 10/13/87
    5700  C*  A REQUEST FROM THE SOURCE PROGRAM RESULTS IN READING A SINGLE    10/13/87
    5800  C*  RECORD CONTAINING THE REQUESTED CUSTOMER OR ORDER NUMBER.  THE  10/13/87
    5900  C*  RESPONSE WILL BE RETURNED IN A SINGLE RECORD CONTAINING EITHER  10/13/87
    6000  C*  THE ITEM OR CUSTOMER INFORMATION, DEPENDING ON THE DATABASE     10/13/87
    6100  C*  CONTENT.                                                        10/13/87
    6200  C*                                                                 10/13/87
    6300  C*  THE RESPONSE IS SENT TO THE SOURCE PROGRAM BY WRITING TO THE    10/13/87
    6400  C*  ICF FILE USING FORMAT SNDPART.                                  01/18/88
    6500  C*                                                                 10/13/87
    6600  C*****************************************************************  10/13/87
    6700  C* 4                                                               03/20/89
    6800  C           XMIT      TAG                                          10/13/87
    6900  C           DBSEQ     CHAINPFILE                98     98 IF NOT FD   1    03/20/89
    7000  C                     MOVE DBSEQ     ITEMNO                        10/13/87
    7100  C                     MOVE RECCUS    RECTYP          RECD FMT ID   10/13/87
    7200  C*****************************************************************  10/13/87
    7300  C*                                                                 03/20/89
    7400  C*  WHEN THE REQUESTED CUSTOMER OR ITEM NUMBER IS NOT FOUND,        10/13/87
    7500  C*  000000 IS PROPAGATED TO THE KEY FIELD BEFORE THE RESPONSE       10/13/87
5728RG1 R01M02  881028            IBM AS/400 RPG/400                ICFLIB/RTDMUL      03/20/89  15:45:02      Page        4
SEQUENCE                                                                      IND    DO   LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM  UPDATE     LINE   ID
    7600  C*  IS SENT BACK TO THE SOURCE PROGRAM.                            10/13/87
    7700  C*                                                                 03/20/89
```

Figure 11-26 (Part 2 of 5). Target Program Example — RTDMUL (User-Defined Formats)

```
    7800  C*****************************************************************              10/13/87
    7900  C   98               MOVE '000000'  ITEMNO                                      10/13/87
    8000  C                    MOVELDBDATA    EDATA           MOVE  DATA                  10/13/87
    8100  C                    WRITESNDPART                   DATA W/DET                  10/13/87
    8200  C           MAJCOD   CABGT'03'      ERROR                                       10/13/87
    8300  C                    GOTO READ                                                  10/13/87
    8400  C*****************************************************************              10/13/87
    8500  C*                                                                             10/13/87
    8600  C* IF ANY ICF FILE ERROR OCCURS, PRINT THE ERROR MESSAGE,                       01/18/88
    8700  C* AND THEN END THE JOB.                                                       10/13/87
    8800  C*                                                                             10/13/87
    8900  C*****************************************************************              10/13/87
    9000  C* 5                                                                          03/20/89
    9100  C           ERROR    TAG                                                       10/13/87
    9200  C                    EXCPTMMERR                                                10/13/87
    9300  C           END      TAG                                                       10/13/87
    9400  C* 6                                                                          03/20/89
    9500  C                    SETON                    LR                  1            10/13/87
    9600  C                    RETRN                                                     10/13/87
    9700  C*****************************************************************              10/14/87
    9800  C*                                                                             10/14/87
    9900  C*    THIS IS THE PROGRAM EXCEPTION/ERROR SUBROUTINE THAT RECEIVES             10/14/87
   10000  C*    CONTROL WHEN AN EXCEPTION OR ERROR OCCURS AFTER AN I/O IS                03/20/89
   10100  C*    ISSUED TO AN ICF PROGRAM DEVICE AND THERE IS A NON-ZERO                  03/20/89
   10200  C*    VALUE UPDATED IN THE RPG STATUS FIELD (ERR). THIS ROUTINE                03/20/89
   10300  C*    CHECKS FOR STATUS VALUES THAT RELATE TO AN ICF OPERATION.                03/20/89
   10400  C*    IF THE PROGRAM DEVICE IS ALREADY ACQUIRED, THE EXCEPTION IS              03/20/89
   10500  C*    IGNORED, OTHERWISE THE PROGRAM IS ENDED.                                 03/20/89
   10600  C*                                                                             10/14/87
   10700  C*****************************************************************              10/14/87
   10800  C* 7                                                                          03/20/89
   10900  C           *PSSR    BEGSR                                                     10/14/87
   11000  C                    MOVE '      '  RETURN  6        DEFAULT                  10/14/87
   11100  C           ERR      CABEQ01285     ENDPSR           ALREADY ACQ?             10/14/87
   11200  C                    MOVE '*CANCL'  RETURN           JOB ENDS                 10/14/87
   11300  C           ENDPSR   ENDSRRETURN                    BACK TO MAIN             10/14/87
   11400  C*************************************************************                10/13/87
   11500  OQPRINT  E 1             MMERR                                                 10/13/87
   11600  O                                 21 'ERROR ON ICF '                          01/18/88
   11700  O                                 34 'MAJOR/MINOR:'                           10/13/87
   11800  O                        MAJCOD   37                                          10/13/87
   11900  O                                 38 '/'                                      10/13/87
   12000  O                        MINCOD   40                                          10/13/87
   12100  O                                 49 'FORMAT:'                                10/13/87
   12200  O                        FMTNM    60                                          10/13/87
   12300  O                                 69 'PGMDEV:'                                10/13/87
   12400  O                        PGMDEV   80                                          10/13/87
* 6103     12401   OVERFLOW INDICATOR OA ASSIGNED TO FILE QPRINT.
  E000000   OUTPUT FIELDS FOR RECORD SNDPART FILE CFILE FORMAT SNDPART.
  E000001                           RECTYP     1  CHAR    1
 5728RG1 R01M02  881028            IBM AS/400 RPG/400                 ICFLIB/RTDMUL       03/20/89  15:45:02      Page       5
 SEQUENCE                                                                      IND    DO    LAST        PAGE   PROGRAM
 NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE      LINE   ID
  E000002                           ITEMNO     7  CHAR    6
  E000003                           EDATA    137  CHAR  130
  E000004                           FILL1    150  CHAR   13
          * * * * *  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
* 7089      1300   RPG PROVIDES SEPARATE INDICATOR AREA FOR FILE CFILE.
 5728RG1 R01M02  881028            IBM AS/400 RPG/400                 ICFLIB/RTDMUL       03/20/89  15:45:02      Page       6
               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  PFILE
           DBREC
                    DBSEQ                 CHAR    6
 5728RG1 R01M02  881028            IBM AS/400 RPG/400                 ICFLIB/RTDMUL       03/20/89  15:45:02      Page       7
```

Figure 11-26 (Part 3 of 5). Target Program Example — RTDMUL (User-Defined Formats)

```
                    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  CFILE       WORKSTN        1300D
         RCVPART                  1300D B000000     4300
         RCVTRND                  1300D C000000     5100
         SNDPART                  1300D A000000     8100  E000000
   02  PFILE       DISK           1600D    6900
         DBREC                    1600D D000000
   03  QPRINT      PRINTER        1700D   11500    12401
Field References:
       FIELD       ATTR    REFERENCES (M=MODIFIED D=DEFINED)
       *IN40       A(1)       4800     5000
       *IN44       A(1)       4500     5300
       *PSSR       BEGSR      1300    10900D
       DBDATA      A(130)  D000003D    8000
* 7031  DBFILL     A(13)   D000004D
       DBSEQ       A(6)    D000002D    4600M    6900     7000
       EDATA       A(130)  A000003D    8000M E000003D
       END         TAG        4500     5300     9300D
       ENDPSR      ENDSR     11100    11300D
       ERR         Z(5,0)     2100D   11100
       ERROR       TAG        4400     5200     8200     9100D
       FEEDBK      DS(404)    1300     1900D
       FILL1       A(13)   A000004D E000004D
* 7031  FILL4      A(64)   B000003D
       FMTNM       A(8)       2200D   12200
       ITEMNO      A(6)    A000002D    7000M    7900M E000002D
* 7031  LOC        A(8)       2000D
       MAJCOD      A(2)       2500D    4400     5200     8200    11800
       MAJMIN      A(4)       2400D    4700
       MINCOD      A(2)       2600D   12000
       MMERR       EXCPT      9200    11500
* 7031  PARTDS     A(80)   B000002D
       PGMDEV      A(10)      2300D   12400
       READ        TAG        4200D    8300
       RECCUS      A(1)    D000001D    7100
       RECID2      A(6)    B000001D    4600
       RECTYP      A(1)    A000001D    7100M E000001D
       RETURN      A(6)      11000D   11200M   11300
       XMIT        TAG        4700     4800     6800D
       '     '     LITERAL   11000
       '*CANCL'    LITERAL   11200
       '0'         LITERAL    5000
       '0000'      LITERAL    4700
5728RG1 R01M02  881028            IBM AS/400 RPG/400                ICFLIB/RTDMUL        03/20/89  15:45:02      Page       8
       '000000'    LITERAL    7900
       '03'        LITERAL    4400     5200     8200
       '1'         LITERAL    4500     4800     5300
       01285       LITERAL   11100
Indicator References:
       INDICATOR  REFERENCES (M=MODIFIED D=DEFINED)
       *IN           4500     4800     5000     5300
       LR            9500M
       OA            1700D   12401
* 7031  05
* 7031  10
* 7031  15
* 7031  20
* 7031  25
* 7031  30
       40            4800     5000
       44            4500     5300
* 7031  50            4300M    5100M
       98            6900M    7900
* 7031  99            4300M    5100M
     * * * * *  E N D   O F   C R O S S   R E F E R E N C E  * * * * *
```

Figure 11-26 (Part 4 of 5). Target Program Example — RTDMUL (User-Defined Formats)

```
5728RG1 R01M02  881028            IBM AS/400 RPG/400                ICFLIB/RTDMUL       03/20/89  15:45:02      Page       9
                        M e s s a g e   S u m m a r y
* QRG6103 Severity:  00   Number:    1
          Message . . . . :   No Overflow Indicator is specified but an
            indicator is assigned to a file and automatic skip to 6 is
            generated.
* QRG7031 Severity:  00   Number:   12
          Message . . . . :   The Name or indicator is not referenced.
* 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 R01M02  881028            IBM AS/400 RPG/400                ICFLIB/RTDMUL       03/20/89  15:45:02      Page      10
                          F i n a l   S u m m a r y
Message Count:  (by Severity Number)
           TOTAL    00     10     20     30     40     50
            14      14      0      0      0      0      0
Program Source Totals:
   Records . . . . . . . . . . :   124
   Specifications  . . . . . . :   54
   Table Records . . . . . . . :   0
   Comments  . . . . . . . . . :   70
PRM has been called.
Program RTDMUL is placed in library ICFLIB. 00 highest Error-Severity-Code.
         * * * * *  E N D   O F   C O M P I L A T I O N  * * * * *
```

Figure 11-26 (Part 5 of 5). Target Program Example — RTDMUL (User-Defined Formats)

```
5728RG1 R01M02  881028            IBM AS/400 RPG/400                 ICFLIB/RTFMUL       03/20/89  15:46:08      Page       1
Compiler . . . . . :   IBM AS/400 RPG/400
Command Options:
  Program . . . . . :   ICFLIB/RTFMUL
  Source file . . . . . :   ICFLIB/QICFPUB
  Source member . . . . . :   *PGM
  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
  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 . . . . . :   RTFMUL
  File . . . . . :   QICFPUB
  Library  . . . . . :   ICFLIB
  Last Change  . . . . . :   03/20/89  15:40:37
  Description  . . . . . :   RPG Multi-Session example w/$$FORMAT (target)
5728RG1 R01M02  881028            IBM AS/400 RPG/400                 ICFLIB/RTFMUL       03/20/89  15:46:08      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  H*****************************************************************                 10/13/87
     200  H*                                                                                 03/20/89
     300  H*  THIS PROGRAM WILL HANDLE THE REQUEST FOR EITHER A CUSTOMER                      10/13/87
     400  H*  NUMBER OR AN ITEM NUMBER. THIS IS ACCOMPLISHED BY MAKING                       10/13/87
     500  H*  THE DATABASE FILE STRUCTURE (KEY LENGTH, KEY POSITION,                        10/13/87
     600  H*  RECORD LENGTH, RECORD SIZE, ETC.) THE SAME FOR BOTH FILES                      03/20/89
     700  H*  WITH ONLY THE RECORD CONTENTS DIFFERENT.                                       03/20/89
     800  H*                                                                                 03/20/89
     900  H*  THIS PROGRAM ENDS WHEN A DETACH REQUEST IS RECEIVED FROM                       10/13/87
    1000  H*  THE REMOTE PROGRAM.                                                           10/13/87
    1100  H*                                                                                 03/20/89
    1200  H*****************************************************************                 10/13/87
          H
    1300  FCFILE   CF  F     256           WORKSTN                                         10/13/87
    1400  F                                            KINFDS FEEDBK                       10/13/87
    1500  F                                            KINFSR *PSSR                        10/14/87
    1600  FPFILE   IF  E           K       DISK                                            10/13/87
           RECORD FORMAT(S):  LIBRARY ICFLIB FILE PFILE.
                     EXTERNAL FORMAT DBREC RPG NAME DBREC
    1700  FQPRINT  O   F     132           PRINTER                                         10/13/87
    1800   * 1                                                                             03/20/89
    1900  ICFILE   NS  99                                                                  10/13/87
    2000  I                                       1   6 RECID2                            10/13/87
    2100  I                                       7 150 PARTDS                            10/13/87
 A000000   INPUT  FIELDS FOR RECORD DBREC FILE PFILE FORMAT DBREC.
 A000001                                         1   1 RECCUS
 A000002                                         2   7 DBSEQ
 A000003                                         8 137 DBDATA
 A000004                                       138 150 DBFILL
    2200  IFEEDBK      DS                                                                  10/13/87
    2300  I                                    *ROUTINE LOC                               10/14/87
    2400  I                                    *STATUS  ERR                               10/14/87
    2500  I                                      38  47 FMTNM                             10/13/87
    2600  I                                     273 282 PGMDEV                            10/13/87
    2700  I                                     401 404 MAJMIN                            10/13/87
    2800  I                                     401 402 MAJCOD                            10/13/87
    2900  I                                     403 404 MINCOD                            10/13/87
```

Figure 11-27 (Part 1 of 4). Target Program Example — RTFMUL (System-Supplied Formats)

```
    3000 C*****************************************************************      10/13/87
    3100 C*                                                                      03/20/89
    3200 C*  READ THE REQUEST FROM THE SOURCE PROGRAM.  INDICATOR 40             10/13/87
    3300 C*  INDICATES RCVTRNRND OCCURRED.  INDICATOR 44 INDICATES THAT           10/13/87
    3400 C*  DETACH HAS BEEN RECEIVED.                                           10/13/87
    3500 C*                                                                      10/13/87
    3600 C*  INDICATOR 99 WILL BE TURNED ON FOR "I/O ERRORS" THEREBY             10/13/87
5728RG1 R01M02  881028            IBM AS/400 RPG/400                ICFLIB/RTFMUL      03/20/89  15:46:08      Page        3
SEQUENCE                                                                  IND    DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE     LINE   ID
    3700 C*  PREVENTING THE RPG DEFAULT ERROR HANDLER FROM BEING CALLED.        10/13/87
    3800 C*  THIS IS NECESSARY TO ALLOW THE PROGRAM TO PROCESS THE ICF          01/18/88
    3900 C*  MAJOR/MINOR RETURN CODE.  THIS PROGRAM CHECKS FOR ERRORS ON        10/13/87
    4000 C*  EVERY ICF FILE OPERATION.  A MAJOR CODE GREATER THAN 03            01/18/88
    4100 C*  INDICATES AN ERROR.                                               10/13/87
    4200 C*                                                                    10/13/87
    4300 C*****************************************************************    10/13/87
    4400 C* 2                                                                  03/20/89
    4500 C           READ      TAG                                             10/13/87
    4600 C                     READ CFILE                9950                  2 3      10/13/87
    4700 C           MAJCOD    CABGT'03'      ERROR           SESSION ERR              10/13/87
    4800 C           MINCOD    CABEQ'08'      END             DETACH RECV?             10/13/87
    4900 C                     MOVE RECID2    DBSEQ           SAVE RECD #              10/13/87
    5000 C           MAJMIN    CABEQ'0000'    XMIT            RCVTRNRND?               10/13/87
    5100 C           MAJMIN    CABEQ'0300'    XMIT            RCVTRNRND?               10/13/87
    5200 C* 3                                                                  03/20/89
    5300 C           MINCOD    DOWNE'00'                      RCVTRNRND?          B001  10/13/87
    5400 C                     READ CFILE                9950                  2 3  001  10/13/87
    5500 C           MAJCOD    CABGT'03'      ERROR                               001  10/13/87
    5600 C           MINCOD    CABEQ'08'      END             DETACH RECV?        001  10/13/87
    5700 C                     END                                                E001  10/13/87
    5800 C*****************************************************************    10/13/87
    5900 C*                                                                    03/20/89
    6000 C*  A REQUEST FROM THE SOURCE PROGRAM RESULTS IN READING A SINGLE      10/13/87
    6100 C*  RECORD CONTAINING THE REQUESTED CUSTOMER OR ORDER NUMBER. THE      10/13/87
    6200 C*  RESPONSE WILL BE RETURNED IN A SINGLE RECORD CONTAINING EITHER     10/13/87
    6300 C*  THE ITEM OR CUSTOMER INFORMATION, DEPENDING ON THE DATABASE        10/13/87
    6400 C*  CONTENT.                                                          10/13/87
    6500 C*                                                                    10/13/87
    6600 C*  THE RESPONSE IS SENT TO THE SOURCE PROGRAM BY WRITING TO THE       10/13/87
    6700 C*  ICF FILE USING FORMAT SNDPART.                                    01/18/88
    6800 C*                                                                    03/20/89
    6900 C*****************************************************************    10/13/87
    7000 C* 4                                                                  03/20/89
    7100 C           XMIT      TAG                                             10/13/87
    7200 C           DBSEQ     CHAINPFILE              98     98 IF NOT FD   1         10/13/87
    7300 C                     MOVE DBSEQ     RECID2                           10/13/87
    7400 C*************************************************************       10/13/87
    7500 C*                                                                    03/20/89
    7600 C*  WHEN THE REQUESTED CUSTOMER OR ITEM NUMBER IS NOT FOUND,          10/13/87
    7700 C*  000000 IS PROPAGATED TO THE KEY FIELD BEFORE THE RESPONSE         10/13/87
    7800 C*  IS SENT BACK TO THE SOURCE PROGRAM.                               10/13/87
    7900 C*                                                                    03/20/89
    8000 C*************************************************************       10/13/87
    8100 C   98                MOVE '000000'  RECID2                          10/13/87
    8200 C                     EXCPTSNDITM                    DATA            10/13/87
    8300 C           MAJCOD    CABGT'03'      ERROR                           10/13/87
    8400 C                     GOTO READ                                      10/13/87
    8500 C*************************************************************       10/13/87
    8600 C*                                                                    10/13/87
    8700 C* IF ANY ICF FILE ERROR OCCURS, PRINT THE ERROR MESSAGE,             01/19/88
    8800 C* AND THEN END THE JOB.                                              10/13/87
    8900 C*                                                                    10/13/87
    9000 C*************************************************************       10/13/87
```

Figure 11-27 (Part 2 of 4). Target Program Example — RTFMUL (System-Supplied Formats)

```
5728RG1 R01M02  881028            IBM AS/400 RPG/400                 ICFLIB/RTFMUL       03/20/89  15:46:08      Page       4
SEQUENCE                                                                                IND   DO    LAST        PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE   NUM   UPDATE      LINE   ID
    9100  C* 5                                                                                    03/20/89
    9200  C           ERROR     TAG                                                               10/13/87
    9300  C                     EXCPTMMERR                                                        10/13/87
    9400  C           END       TAG                                                               10/13/87
    9500   * 6                                                                                    03/20/89
    9600  C                     SETON                     LR                  1                 10/13/87
    9700  C                     RETRN                                                             10/13/87
    9800  C*****************************************************************                     10/14/87
    9900  C*                                                                                      10/14/87
   10000  C*    THIS IS THE PROGRAM EXCEPTION/ERROR SUBROUTINE THAT RECEIVES                      10/14/87
   10100  C*    CONTROL WHEN AN EXCEPTION OR ERROR OCCURS AFTER AN I/O                            10/14/87
   10200  C*    IS ISSUED TO AN ICF PROGRAM DEVICE AND THERE IS A                                 01/19/88
   10300  C*    NON-ZERO VALUE UPDATED INTO THE RPG STATUS FIELD (ERR).                           01/19/88
   10400  C*    THIS ROUTINE CHECKS FOR STATUS VALUES THAT RELATE TO                              01/19/88
   10500  C*    ICF OPERATION.                                                                    01/19/88
   10600  C*    IF THE PROGRAM DEVICE IS ALREADY ACQUIRED, THE EXCEPTION IS                       03/20/89
   10700  C*    IGNORED, OTHERWISE THE PROGRAM IS ENDED.                                          03/20/89
   10800  C*                                                                                      10/14/87
   10900  C*****************************************************************                     10/14/87
   11000  C* 7                                                                                    03/20/89
   11100  C           *PSSR     BEGSR                                                             10/14/87
   11200  C                     MOVE '      '  RETURN  6        DEFAULT                          10/14/87
   11300  C           ERR       CABEQ01285     ENDPSR           ALREADY ACQ?                     10/14/87
   11400  C                     MOVE '*CANCL'  RETURN           JOB ENDS                         10/14/87
   11500  C           ENDPSR    ENDSRRETURN                     BACK TO MAIN                     03/20/89
   11600  OQPRINT  E 1              MMERR                                                         10/13/87
   11700  O                                    21 'ERROR ON ICF'                                 01/21/88
   11800  O                                    34 'MAJOR/MINOR:'                                 10/13/87
   11900  O                         MAJCOD     37                                                10/13/87
   12000  O                                    38 '/'                                            10/13/87
   12100  O                         MINCOD     40                                                10/13/87
   12200  O                                    49 'FORMAT:'                                      10/13/87
   12300  O                         FMTNM      60                                                10/13/87
   12400  O                                    69 'PGMDEV:'                                      10/13/87
   12500  O                         PGMDEV     80                                                10/13/87
   12600  OCFILE   E                SNDITM                                                       10/13/87
   12700  O                                    K6 '$$SEND'                                       10/13/87
   12800  O                                     4 '0150'                                         10/13/87
   12900  O                         RECCUS      5                                               10/13/87
   13000  O                         RECID2     11                                               10/13/87
   13100  O                         DBDATA    141                                               10/13/87
   13200  O                         DBFILL    154                                               10/13/87
* 6103     13201   OVERFLOW INDICATOR OA ASSIGNED TO FILE QPRINT.
          * * * * *  E N D  O F  S O U R C E  * * * * *
         A d d i t i o n a l  D i a g n o s t i c  M e s s a g e s
* 7089      1300   RPG PROVIDES SEPARATE INDICATOR AREA FOR FILE CFILE.
5728RG1 R01M02  881028            IBM AS/400 RPG/400                 ICFLIB/RTFMUL       03/20/89  15:46:08      Page       5
                 K e y  F i e l d  I n f o r m a t i o n
                     PHYSICAL    LOGICAL
       FILE/RCD      FIELD       FIELD       ATTRIBUTES
   02  PFILE
           DBREC
                     DBSEQ                   CHAR    6
5728RG1 R01M02  881028            IBM AS/400 RPG/400                 ICFLIB/RTFMUL       03/20/89  15:46:08      Page       6
                       C r o s s  R e f e r e n c e
File and Record References:
       FILE/RCD    DEV/RCD     REFERENCES (D=DEFINED)
   01  CFILE       WORKSTN       1300D    1900     4600     5400    12600
         $$SEND                 12700
   02  PFILE       DISK          1600D    7200
         DBREC                   1600D A000000
   03  QPRINT      PRINTER       1700D   11600    13201
Field References:
       FIELD       ATTR    REFERENCES (M=MODIFIED D=DEFINED)
       *PSSR       BEGSR      1300    11100D
```

Figure 11-27 (Part 3 of 4). Target Program Example — RTFMUL (System-Supplied Formats)

```
        DBDATA      A(130)  A000003D   13100
        DBFILL      A(13)   A000004D   13200
        DBSEQ       A(6)    A000002D    4900M    7200     7300
        END         TAG        4800     5600     9400D
        ENDPSR      ENDSR     11300    11500D
        ERR         Z(5,0)     2400D   11300
        ERROR       TAG        4700     5500     8300     9200D
        FEEDBK      DS(404)    1300     2200D
        FMTNM       A(8)       2500D   12300
* 7031  LOC         A(8)       2300D
        MAJCOD      A(2)       2800D    4700     5500     8300    11900
        MAJMIN      A(4)       2700D    5000     5100
        MINCOD      A(2)       2900D    4800     5300     5600    12100
        MMERR       EXCPT      9300    11600
* 7031  PARTDS      A(144)     2100D
        PGMDEV      A(10)      2600D   12500
        READ        TAG        4500D    8400
        RECCUS      A(1)    A000001D   12900
        RECID2      A(6)       2000D    4900     7300M    8100M   13000
        RETURN      A(6)      11200D   11400M   11500
        SNDITM      EXCPT      8200    12600
        XMIT        TAG        5000     5100     7100D
        '      '    LITERAL   11200
        '*CANCL'    LITERAL   11400
        '00'        LITERAL    5300
        '0000'      LITERAL    5000
        '000000'    LITERAL    8100
        '03'        LITERAL    4700     5500     8300
        '0300'      LITERAL    5100
        '08'        LITERAL    4800     5600
        01285       LITERAL   11300
5728RG1 R01M02  881028            IBM AS/400 RPG/400                 ICFLIB/RTFMUL       03/20/89  15:46:08      Page        7
Indicator References:
        INDICATOR  REFERENCES (M=MODIFIED D=DEFINED)
        LR            9600M
        OA            1700D   13201
* 7031  50            4600M    5400M
        98            7200M    8100
* 7031  99            1900M    4600M    5400M
     * * * * *  E N D  O F  C R O S S  R E F E R E N C E  * * * * *
5728RG1 R01M02  881028            IBM AS/400 RPG/400                 ICFLIB/RTFMUL       03/20/89  15:46:08      Page        8
                        M e s s a g e  S u m m a r y
* QRG6103 Severity:  00   Number:    1
          Message . . . . :   No Overflow Indicator is specified but an
            indicator is assigned to a file and automatic skip to 6 is
            generated.
* QRG7031 Severity:  00   Number:    4
          Message . . . . :   The Name or indicator is not referenced.
* 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 R01M02  881028            IBM AS/400 RPG/400                 ICFLIB/RTFMUL       03/20/89  15:46:08      Page        9
                          F i n a l  S u m m a r y
Message Count:  (by Severity Number)
           TOTAL    00     10     20     30     40     50
             6       6      0      0      0      0      0
Program Source Totals:
   Records . . . . . . . . . . :   132
   Specifications  . . . . . . :   62
   Table Records . . . . . . . :   0
   Comments  . . . . . . . . . :   70
PRM has been called.
Program RTFMUL is placed in library ICFLIB. 00 highest Error-Severity-Code.
         * * * * *  E N D  O F  C O M P I L A T I O N  * * * * *
```

Figure 11-27 (Part 4 of 4). Target Program Example — RTFMUL (System-Supplied Formats)

Chapter 12. Tracing Intersystem Communications Function Operations and Functions

You can use the Trace Intersystem Communications Function (TRCICF) command to save information about the language operations and communications functions directed to an ICF file. The trace information can be collected in the current job or in the job being serviced as a result of the Start Service Job (STRSRVJOB) command.

The Start Service Job (STRSRVJOB) command allows you to collect trace records for jobs started from other work stations or for batch jobs. After the STRSRVJOB command has been entered, the TRCICF command must be entered to start the trace.

The End Service Job (ENDSRVJOB) command is used to end the service job request. No parameters are used with this command. The trace must be stopped before this command can be entered. The *CL Reference* has more information about the STRSRVJOB and ENDSRVJOB commands.

Trace ICF traces all ICF I/O operations that occur in the job in which the command was entered. During the time that TRCICF is active, all programs that run in the job are monitored by TRCICF. TRCICF can be entered within different jobs, and the trace for one job runs independently of the trace for another job.

The Trace ICF function can be started, stopped, or ended. You can start the Trace ICF function from a system menu, by typing the TRCICF command on a command line, or from a control language (CL) program within a job. After the trace is started, trace records are collected and stored in an internal trace storage area. When the trace is stopped, the trace records can either be directed to the spooled printer file, QPIFTRCF, or sent to a database output file that you specify. When the trace is ended, all trace records are deleted. Details about starting, stopping, and ending the Trace ICF function are discussed in this chapter.

Starting the Trace

The Trace ICF (TRCICF) function can be started before running a job or after the job is active (as in a remote job). You can start TRCICF from a system menu, by typing TRCICF *ON on any command line, by adding the command to a CL program, or by typing TRCICF on a command line and pressing F4 (Prompt). If the latter method is used, an initial prompt is displayed for the *Trace option setting*. If *ON is specified and you press the Enter key, the following display is shown:

```
TRCICF                        Trace ICF

Type choices, press Enter

Trace option setting . . . . . *ON       *ON, *OFF, *END
Maximum storage to use . . . . . 200      1-16000 K
Trace full . . . . . *WRAP    *WRAP, *STOPTRC
User data length . . . . . 128      0-4096

                                                              Bottom
F3=Exit  F4=Prompt  F5=Refresh
F12=Cancel  F13=How to use this display  F24=More keys
```

Figure 12-1. Starting the ICF Trace

Trace option setting

Specify whether collecting trace information is to be started, stopped, or ended.

*ON — Trace ICF is to be started. This is the default value for this prompt.

*OFF — Trace ICF is stopped. No other trace information is recorded and the current information is written to the spooled printer file or a database file.

*END — Trace ICF is ended. All trace information is deleted.

Maximum storage to use

Specify the maximum amount of storage to use for the trace information collected. This prompt is only shown if you have selected *ON for the *Trace option setting* prompt.

200 K — The number of bytes (1 K equals 1024 bytes) of maximum storage. This is the default value.

1-16000 K — The valid range for the number of bytes of maximum storage.

Trace full

Specify whether new trace records are to replace the old trace records or to stop the trace function when the maximum storage specified has been reached. This prompt is only shown if you have selected *ON for the *Trace option setting* prompt. Valid values are:

*WRAP — When the trace storage area is full, new trace information is written over the oldest information, starting at the beginning of the storage area. This is the default value.

*STOPTRC — No new trace information is saved when the trace storage area is full. You must turn the trace off to get the output.

User data length

Specify the maximum length of user data to be saved for each trace record in the storage area. This prompt is only shown if you have selected *ON for the *Trace option setting prompt.*

128 The number of bytes for the user data length. This is the default value.

0-4096 The valid range of bytes for the user data length.

Stopping the Trace

Trace ICF continues to collect trace records until you stop the trace, or until the trace storage area is full. When you stop the trace, the trace records that have been created are either directed to the spooled printer file, QPIFTRCF, or to a database output file that you specify. If the output file specified already exists, it must have the same attributes as the system-supplied file QAIFTRCF.

You can stop the trace from a system menu, by typing TRCICF *OFF on any command line, by adding the command to a CL program, or by typing TRCICF and pressing F4 (prompt). If the latter method is used, and you specify *OFF for the *Trace option setting*, you are prompted for the OUTPUT parameter. If you specify the *OUTFILE option for the *Output* prompt, the following display is shown:

```
TRCICF                    Trace ICF


Type choices, press Enter

Trace option setting . . . . . *OFF       *ON, *OFF, *END
Output . . . . . *OUTFILE   *PRINT, *OUTFILE
Output file  . . . . . .          Name
  Library  . . . . . *LIBL      Name, *LIBL, *CURLIB
Output member options:
  Member to receive output . . . . . *FIRST     Name, *FIRST
  Replace or add records . . . . . *REPLACE   *REPLACE, *ADD









                                                                Bottom

F3=Exit  F4=Prompt  F5=Refresh
F12=Cancel  F13=How to use this display  F24=More keys
```

Figure 12-2. Stopping the ICF Trace

Output

Specifies whether the trace information is to be stored in a spooled file or saved in a database file. This display is only shown if *OFF is specified for the *Trace option setting* prompt. Valid values are:

*PRINT The trace information is sent to the spooled file QPIFTRCF in the output queue associated with the job being traced. The spooled file can be viewed or printed. Refer to Figure 12-3 on page 12-5 for an example of the spooled trace records. This is the default value.

*OUTFILE — The trace records are to be directed to a database file. Refer to Figure 12-4 on page 12-8 for a description of trace records directed to a database file. The *OUTFILE value for the *Output* prompt is only valid if a value is specified for the *Output file* prompt.

Output file

Specifies the name of the database file to which the trace records are to be sent. This prompt is only shown if you have selected *OFF for the *Trace option setting* prompt and *OUTFILE for the *Output* prompt. If the file does not exist, the system creates a new database file with the specified name in the library to which the file is to be added. The new file has the same attributes as the system-supplied file QAIFTRCF. If the file already exists, it must have the same attributes as the system-supplied file QAIFTRCF. Possible library values are:

Name — The name of the library where the file is located.

*LIBL — The file is located in the library list.

*CURLIB — The file is located in the current library for the job. If no current library entry exists in the library list, the library QGPL is used.

Output member options

Specifies the name of the file member that is to receive the trace information. This prompt is only shown if you have selected *OFF for the *Trace option setting* prompt and *OUTFILE for the *Output* prompt. If the output file is to be created by the system, an output member is also created and given the name specified in the *Member to receive output* prompt. If *FIRST is specified for the *Member to receive output* prompt, a member is created and given the name specified in the output file. If the output file exists, but the output member does not, a member with the specified name is created. The options for the *Output member options* prompt are:

Member to receive output

Type the name of the member to receive the output. Valid values are:

*FIRST — The first member in the output file receives the trace information. This is the default.

Name — The specified member receives the trace information.

Replace or add records

The trace information is either added to the file or replaces existing trace information. Valid values are:

*REPLACE — New trace information replaces what is already in the file member. This is the default.

*ADD — New trace information is appended to the end of data already in the file member.

Trace Records Sent to a Spooled File

When you select *OFF for the *Trace option setting* prompt and press F4, you are presented with the option on the *Output* prompt to send the trace records to a spooled file (*PRINT) or to a database file (*OUTFILE). The default value is *PRINT. If you choose the *PRINT value for the *Output* prompt, the trace information is sent to the spooled file QPIFTRCF. Figure 12-3 shows the format of the spooled trace records.

```
5728SS1 R02 M00 891006                    AS/400 Trace ICF Information          11/08/89  9:56:10       PAGE    1
-----------------Table of Function Codes----------------
1 Function Codes     Meaning
  ACQ                Acquire
  AWT                Allow-Write
  CFM                Confirm
  CLS                Close File Prior to REL or EOS
  CNI                Cancel-Invite
  CNL                Cancel
  DET                Detach
  EGP                End-of-Group
  EOA                End-of-Session-Abnormal
  EOS                End-of-Session
  ERR                Error: Function Not Valid
  EVK                Evoke
  FAL                Fail
  FMH                Function-Management-Header
  FMT                Format-Name
  FRC                Force-Data
  INV                Invite
  NRP                Negative-Response
  OPN                Open with Acquire-Program-Device
  RCF                Respond-to-Confirm
  RCV                Receive
  REL                Release
  RFI                Read-From-Invited-Program-Devices
  RST                Restore
  RWT                Request-to-Write
  SDV                Subdevice-Selection
  SND                Send
  SPD                Suspend
  TMR                Timer
  TRN                Turn-Around
5728SS1 R02 M00 891006                    AS/400 Trace ICF Information          11/08/89 15:40:47       PAGE    2
2 Job  . . . :   DSP10     3 User . . . :   QUSER      4 Number . . . . :   006267
5 Program . . . . :   ICFTEST   /ICFMAIN          6 Program File  . . . . :   ICFTEST   /ICFTSTCF
                                                  7 Opened File   . . . . :   ICFLIB    /ICFTSTITF

8 Program   9 Record    10 Return   11 Function   12 Response   13 Data   14 Remote   15 Time
  Device      Format       Code                     Indicator     Length    Location

DEV1                     0000        ACQ                          0        Chicago      15:37:00.857
DEV1          EVOKENOV   0000        EVK                          0        Chicago      15:37:06.264
DEV1          NOVARLEN   0001        SND,INV                     80        Chicago      15:37:38.798
DATA:
  THIS IS A PUT WITH INVITE.
DEV1          NOVARLEN   0308        RCV              DET         0        Chicago      15:37:50.097
DEV1          EVOKENOV   0000        EVK                          0        Chicago      15:38:08.887
DEV1          NOVARLEN   0001        SND,INV                     80        Chicago      15:38:38.912
DATA: 16
  THIS IS ANOTHER PUT WITH INVITE.
DEV1          NOVARLEN   0308        RCV              DET         0        Chicago      15:38:52.835
DEV1                     0000        EOS                          0        Chicago      15:40:43.560

                                 *****  E N D  O F  L I S T I N G  *****
                                                                                        R36B454-1
```

Figure 12-3. Spooled Trace Records

1 Table of Function Codes

The first page of the spooled trace records is a table of the function codes used for each ICF operation. The function code is printed in the Function and Response Indicator columns.

Notes:

1. The suspend (SPD) and restore (RST) function codes are used in the System/36 environment. These function codes are part of the read-under-format (RUF) support. If two programs using the RUF support do not run in the same job, then the Trace ICF function does not trace both programs unless Trace ICF is started for both jobs.
2. The function code OPN indicates that the program opened a file that automatically acquired a program device. CLS indicates that the program closed the file prior to releasing or ending the session.

2 Job

Name of the job in which your program is running.

3 User

The user identification (User ID) used to start the job (either the user ID used to sign on the work station or the user ID received on the program start request).

4 Number

The number assigned to the job step when your program started.

5 Program

Name of the library where the program resides, and the name of the program that issued the operation that is being traced.

6 Program File

Name of the library where the ICF file named in the program resides, and the name of the ICF file named in the program.

7 Opened File

Name of the library where the ICF file opened by your program resides, and the name of the ICF file opened by your program.

Note: If you used the OVRICFF command to temporarily override the file named in the program, the name specified for the Opened File will be different from the Program File name.

8 Program Device

Name assigned to the session to which the language operation or communications function was directed. This is the name specified in the Add ICF Program Device Entry (ADDICFDEVE) or Override ICF Program Device Entry (OVRICFDEVE) commands.

9 Record Format

Name of the record format used when the communications function is issued. The record format can either be a user-defined data description specification (DDS) or a system-supplied format.

10 Return Code

The major and minor return code issued to indicate the success or failure of each operation.

11 Function

The function code assigned to represent the language operation or communications function issued by the program. Only operations associated with your ICF sessions are traced. File open and close operations are not traced except when a program device is acquired or released as a result of an open or close operation.

12 Response Indicator

The function code assigned to represent the DDS response indicator that indicates status information about the input operation.

13 Data Length

Length of data sent or received by the program. If the function indicates a send and receive operation, then this field represents the length of data received by the program.

14 Remote Location

Name of the remote location with which a communication session is established.

15 Time

Time that the language operation or communication function was completed by the communications type. The time is displayed in hours, minutes, seconds, and milliseconds.

16 Data

The data sent or received by the program. The amount of data traced depends on the value specified for the *User data length* prompt (DTALEN parameter) of the TRCICF command. If the function indicates a send and receive operation, then the data received by your program is shown.

Trace Records Sent to a Database File

When you select *OFF for the *Trace option setting*, you are presented with the option either to send the trace records to a spooled file (*PRINT) or send the records to a database file (*OUTFILE). If you choose the *OUTFILE value for the *Output* prompt, the trace information is sent to the database file that you specify. If you specify a file that already exists, it must match the attributes of the system-supplied file QAIFTRCF.

Figure 12-4 on page 12-8 shows the layout of the trace records sent to a database file. The database file has a fixed record length of 4337 decimal bytes. The record format name is QIFTRC. Each record in the file contains all the information related to the language operation or communications function, as well as the length of data traced. The length of data traced is less than or equal to the value specified on the *User data length* prompt (the DTALEN parameter) of the TRCICF command.

```
A*
A* TRACE ICF OUTFILE RECORD FORMAT FOR TRCICF
A*
A          R QIFTRC                    TEXT('TRCICF Record')
A            IFJOB          10         COLHDG('JOB' 'NAME')
A                                       TEXT('Name of Job')
A            IFUSER         10         COLHDG('USER' 'NAME')
A                                       TEXT('Name of User')
A            IFNBR           6         COLHDG('JOB' 'NUMBER')
A                                       TEXT('Number of Job')
A            IFPGM          10         COLHDG('PROGRAM' 'NAME')
A                                       TEXT('Name of Program')
A            IFLIB          10         COLHDG('LIBRARY' 'NAME')
A                                       TEXT('Programs Library')
A            IFPGMF         10         COLHDG('PROGRAM' 'FILE''NAME')
A                                       TEXT('Program File')
A            IFPGML         10         COLHDG('PROGRAM' 'FILE' 'LIBRARY')
A                                       TEXT('Program Files Library')
A            IFOPNF         10         COLHDG('OPENED' 'FILE' 'NAME')
A                                       TEXT('Opened File')
A            IFOPNL         10         COLHDG('OPENED' 'FILE' 'LIBRARY')
A                                       TEXT('Opened Files Library')
A            IFPGDV         10         COLHDG('PROGRAM' 'DEVICE')
A                                       TEXT('Program Device')
A            IFRCFM         10         COLHDG('RECORD' 'FORMAT')
A                                       TEXT('Record Format')
A            IFMJMN          4         COLHDG('RETURN' 'CODE')
A                                       TEXT('Return Code')
A            IFOPCD         48         COLHDG('FUNCTION' 'CODE')
A                                       TEXT('Function Code')
A            IFRSPI         36         COLHDG('RESPONSE' 'INDICATOR')
A                                       TEXT('Response Indicator')
A            IFDTLN          2B        COLHDG('DATA' 'LENGTH')
A                                       TEXT('Data Length')
A            IFRLOC          8         COLHDG('REMOTE' 'LOCATION')
A                                       TEXT('Remote Location')
A            IFTIME          9S 0      COLHDG('TIME')
A                                       TEXT('Time of Entry')
A            IFDTTR          2B        COLHDG('TRACED' 'DATA' 'LENGTH')
A                                       TEXT('Traced Data Length')
A            IFRES          26         COLHDG('RESERVED')
A                                       TEXT('Reserved')
A            IFDATA       4096         COLHDG('DATA')
A                                       TEXT('Data')
```

Figure 12-4. Trace Records Sent to a Database File

Ending the Trace

You can end TRCICF from a system menu, by typing the TRCICF *END command on any command line, by adding the command to a CL program, or by typing TRCICF and pressing F4 to show the *Trace option setting* prompt, shown following. Type *END and press the Enter key. This causes Trace ICF to end and all trace records to be deleted.

```
TRCICF                      Trace ICF

Type choices, press Enter

Trace option setting . . . . . *END      *ON, *OFF, *END

F3=Exit  F4=Prompt  F5=Refresh
F12=Cancel  F13=How to use this display  F24=More keys
```

Figure 12-5. Ending the ICF Trace

Additional Considerations

Trace ICF traces only those operations that are associated with your ICF sessions. File open and close operations are not traced except when a program device is acquired or released as a result of an open or close operation. The following restrictions apply to the trace of open and close operations that occur in a given job:

- When an open of an ICF file is issued without implicit acquire of the program device, the explicit acquire of the program device (ACQ) will be traced and not the open (OPN) operation.
- An open of an ICF file with implicit acquire of the program device is traced as an open operation (OPN).
- When the close of an ICF file is preceded with an end of session (EOS), the end of session is traced but not the close (CLS).
- When the close of an ICF file is preceded by a release operation (REL), the release operation is traced but not the close operation (CLS).
- When a close of an ICF file is not preceded by an EOS or release operation, it is traced as a close (CLS) operation.

Appendix A. Language Operations, Data Description Specifications Keywords, and System-Supplied Formats

This appendix contains charts which show the following:

- All valid language operations supported by intersystem communications function (ICF)
- All valid operations for each programming language that supports ICF, C language (C/400), Common Business-Oriented Language (COBOL/400), and Report Program Generator (RPG/400)
- Data description specifications (DDS) processing keywords supported by communications types
- System-supplied formats supported by communications types

Language Operations

Table A-1 describes the language operations supported by ICF.

Table A-1. Language Operations

ICF Operations	Description
Open	Opens the ICF file.
Acquire	Establishes a session between the application and the remote location.
Get attributes	Used to determine the status of the session.
Read	Obtains data from a specific session.
Read-from-invited-program-devices	Obtains data from any session that has responded to an invite function.
Write	Passes data records from the issuing program to the other program in the transaction.
Write/Read	Allows a write operation followed by a read operation. Valid for RPG/400.
Release	Attempts to end a session.
Close	Closes the ICF file.

Table A-2 shows all the valid communications operations for each programming language that supports ICF (C/400, COBOL/400, RPG/400).

Note: C/400 statements are case sensitive.

Table A-2. Language Operations

ICF Operation	C/400 Operation Code	COBOL/400 Procedure Statement	RPG/400 Function
Open	fopen	OPEN	OPEN
Acquire	QXXACQUIRE	ACQ	ACQUIRE
Get Attributes	QXXDEVAT	POST	ACCEPT
Read	fread	READ	READ
Read-from-Invited-Program-Devices	QXXREADINVDEV, followed by an fread[1]	READ[1]	READ[1]
Write	fwrite	WRITE	WRITE
Write/Read	Not supported	EXFMT	Not supported
Release	QXXRELEASE	REL	DROP
Close	fclose	CLOSE	CLOSE

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

DDS Keyword Support

Table A-3 defines the communications processing control DDS keywords supported for the ICF file, and the communications type that supports these keywords.

Table A-3 (Page 1 of 3). Processing Control DDS Keywords

DDS Keyword	APPC	SNUF	BSCEL	Asyn-chronous	Intra-system	Finance	Retail
ALWWRT[2] The record currently being written ends a transmission. The program goes to receive state.	X	X	X		X		
CANCEL Cancels a group of records that has just been sent.		X			X	X[3]	X
CNLINVITE Cancels any valid invite issued by your program.		X	X	X	X	X	X
CONFIRM[2] Requests that the remote program confirm receiving data.	X				X		
DETACH Informs the remote program that the sending program is ending the transaction.	X	X	X	X[1]	X		X
ENDGRP Indicates the end of a user-defined group of records.		X	X		X	X	X
EOS Used to specify an end-of-session function.	X	X	X	X	X	X	X
EVOKE Starts a program on the remote system.	X	X	X	X	X		X
FAIL Sends a fail indication to the remote system.	X	X	X	X	X	X	
FMH Informs the remote program that a function-management-header (FMH) is being sent.		X		X[1]	X	X[4]	X
FMTNAME[2] Specifies that the format name should be sent on output operations.	X				X		
FRCDTA[2] Immediately sends communications data currently in the buffer, without waiting for the buffer to become full.	X				X	X	X

Table A-3 (Page 2 of 3). Processing Control DDS Keywords

DDS Keyword	APPC	SNUF	BSCEL	Asyn-chronous	Intra-system	Finance	Retail
INVITE Schedules an invite.	X	X	X	X	X	X	X
NEGRSP Informs the remote system that the data received is not valid.		X			X	X	X
RCVCANCEL[2] Indicates that the remote program sent a cancel request.		X			X	X[3]	X
RCVCONFIRM[2] Indicates that the remote program is requesting a confirmation of transaction activity.	X				X		
RCVDETACH[2] Indicates that the remote program has ended the transaction.	X	X	X		X		X
RCVENDGRP[2] Indicates the end of a user-defined group of records sent to the program.		X	X		X	X	X
RCVFAIL[2] Indicates that the remote program issued a fail.	X			X	X		
RCVFMH[2] Indicates to the program that FMH data has been received.		X			X	X[4]	X
RCVNEGRSP[2] Indicates that the remote program issued a negative-response request.		X			X	X	X
RCVTRNRND[2] Indicates that the program is now in send state.	X	X	X		X		
RECID[2] Used to allow the data content to identify the record format to use to receive the data.	X	X	X	X	X	X	X
RQSWRT Specifies that the program is requesting permission to write.	X	X	X		X		
RSPCONFIRM[2] Sends a positive response to a received confirm request.	X				X		

Table A-3 (Page 3 of 3). Processing Control DDS Keywords

DDS Keyword	APPC	SNUF	BSCEL	Asyn-chronous	Intra-system	Finance	Retail
SECURITY Includes security information needed to start a program on the remote system.	X	X	X	X	X	X	X
SUBDEV[2] Specifies the subdevice to which output should be directed (for example, printer, punch, and so on).			X		X		
SYNLVL[2] Indicates the synchronization level of the program.	X				X		
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.	X	X	X	X	X	X	X
VARLEN Specifies the length of the data record sent with each write operation.	X	X	X	X	X	X	X

1 Use of this keyword is restricted. Refer to the *Asynchronous Communications Programmer's Guide* for more details.

2 These DDS keywords do not have system-supplied format equivalents.

3 These keywords are not valid for the 3694 controller. Refer to the *Finance Communications Programmer's Guide* for details.

4 These keywords are valid for the 3694 controller only. Refer to the *Finance Communications Programmer's Guide* for details.

System-Supplied Format Support

The following table defines the system-supplied formats supported for ICF, and shows the communications types that supports these formats.

Table A-4. System-Supplied Format Support

Operation	APPC	SNUF	BSCEL	Asyn-chronous	Intra-system	Finance	Retail
$$CANL Cancel with invite		X			X	X[1]	X
$$CANLNI Cancel		X			X	X[1]	X
$$CNLINV Cancel invite		X	X	X	X	X	X
$$EOS End of session	X	X	X	X	X	X	X
$$EVOK Evoke with invite	X	X	X	X	X		X
$$EVOKET Evoke with detach	X	X	X	X	X		X
$$EVOKNI Evoke	X	X	X	X	X		X
$$FAIL Fail	X	X	X	X	X	X	
$$NRSP Negative response with invite		X			X	X	X
$$NRSPNI Negative response		X			X	X	X
$$RCD Request write with invite	X	X	X		X		
$$SEND Send with invite or invite	X	X	X	X	X	X	X
$$SENDE Send with end of group		X	X		X	X	X
$$SENDET Send with detach	X	X	X		X		X
$$SENDFM Send FMH with invite		X			X	X[2]	X
$$SENDNF Send FMH		X		X	X	X[2]	X
$$SENDNI Send	X	X	X	X	X	X	X
$$TIMER Timer	X	X	X	X	X	X	X

[1] These keywords are not valid for the 3694 controller. Refer to the *Finance Communications Programmer's Guide* for more information.

[2] These keywords are valid for the 3694 controller only. Refer to the *Finance Communications Programmer's Guide* for more information.

Appendix B. Communications Error Handling

This chapter describes programming considerations for ICF communications error recovery. It includes information on:

- System error classifications
- System messages sent on communications errors, and related file error handling in the affected job
- Major/minor return codes and descriptions
- Error reason codes for failed program start requests

System Error Classification

The system divides communications error conditions into several classifications and processes them according to those classifications. The system automatically tries recovery for many of these errors without notification to the using program. In some cases, messages indicating error recovery is in progress are issued to the system operator message queue (QSYSOPR), to the job log, and to other queues specified during device configuration. When the system retry limits specified in configuration objects are exceeded, a message is sent to these queues and any jobs currently using the failing line, controller, or device.

Some errors, such as an application program violation of a communications protocol, do not cause messages to be sent to system message queues, but are reported to the affected program.

It is recommended that all communications programs examine return codes after each operation to detect error conditions and other normal conditions, such as receipt of the detach indication from the remote system. Although many error conditions are reported to the affected job through messages, the primary method for a program to detect these conditions is through return codes and the open feedback and input/output (I/O) feedback areas.

For a complete description of error classifications and system-provided recovery support, see the *Communications User's Guide*.

System Messages

Some errors can occur that do not affect your program. For example, the varying on of a communications line may fail before starting any programs that use devices on the failed line.

Errors that affect your program can occur:

- When a file is opened
- During I/O operations to the file
- When a program device is acquired or released
- When the file is closed

When you encounter errors that can affect the running of a program, a system message is sent to the program message queue of the program using the file.

Error messages are divided into the following message types:

- Notify
- Status
- Diagnostic
- Escape

See the *CL Programmer's Guide* for more information about the message types.

Table B-1 is a summary of the messages, by operation, that can be issued.

Table B-1. File Error Message Identifier Groups

Operation	Message Type	Message Identifiers
Open	Diagnostic and status	CPF4001 through CPF4099
Open	Escapes that make the file unusable	CPF4101 through CPF4399
Close	Diagnostic and status	CPF4401 through CPF4499
Close	Escapes that make the file unusable	CPF4501 through CPF4699
Input/Output, Acquire, and Release	Notify with a default reply of cancel, status, and escapes that do not make the file or program device unusable	CPF4701 through CPF4899 and CPF5001 through CPF5099
Input/Output, Acquire, and Release	Notify with a default reply of ignore	CPF4901 through CPF4999
Input/Output, Acquire, and Release	Escapes that make the file or program device unusable	CPF5101 through CPF5399 and CPF5501 through CPF5699

These messages are logged in the job log, and the error is communicated to the program through language status codes and an ICF major/minor return code in the I/O feedback area of the file.

Some conditions are considered normal application exceptions and do not cause job messages. As a result, file error handling for high-level languages is not called. You may need to examine the I/O feedback area for major/minor return codes or other device-specific information. You can detect some conditions by using data description specifications (DDS) response keywords.

User Program Error Detection

All user programs should detect error conditions and determine appropriate error processing. Review all major/minor return codes described in this chapter, and in the programming manual for the communications type you are using, to determine what processing to do.

Permanent errors can cause the session, your program, or both to end. A program can try to recover from errors without ending. The operation you use and the major code you receive determine how your program recovers from the errors.

In general, you recover from an open operation that fails as follows:

- Close the file
- Correct the problem
- Issue the open operation again

An acquire failure is handled as follows:

- Correct the problem
- Issue the acquire operation again

You can resume communications for most I/O operations that encounter session errors and complete with a major return code of 81 by reacquiring the program device associated with the session. For input/output operations that encounter system or file errors completed with a major return code of 80, you may or may not need to close and reopen the file to resume communications. In some cases, depending on the cause of the error, the device must be varied off, then on again, to remedy the problem. Reacquiring the session that failed may also allow you to resume communications. To determine specific error recovery procedures, check each major/minor return code description in the programming manual for the communications type you are using.

If an I/O operation completes with an exception or a nonpermanent error (04, 08, 11, 34, and 83 majors), then the session is still intact and the program can recover, based on the action described for the major/minor return code.

A release failure can be handled in one of two ways. If you want to end the session gracefully, correct the problem as indicated by the return code, and issue the release operation again. For example, if the release operation completes with an 832F, issue a detach request, and then issue the release again. If you want to force the session to end, issue an end-of-session function.

If a close operation fails, issue the close again. A second close is always successful.

The ICF file is implicitly closed when the job ends, if the job ends without recovering from a failure.

Control Language (CL) Commands for Determining Configuration Status

The Work with Configuration Status (WRKCFGSTS) command provides line, controller, device, and mode status for communications on your AS/400 system. Information provided by this command can help you determine the status of your devices and sessions in order to determine error recovery options. The RTVCFGSTS command is also available to determine line, controller, and device status. See the *Communications User's Guide* for more information on configuration status.

Major/Minor Return Codes

This section contains:

- Return code tables that identify all communications types and the return codes that are valid for each. This summary table is useful when you want to make changes to a program so you can use it with a different communications type.
- Summary descriptions of all major and minor return codes for all communications types.

These return codes are set in the I/O feedback area of the ICF file, and report the results of each I/O operation issued by your application program. Your program should check the return code and act accordingly. Refer to your high-level language manual for more information on how to access these return codes.

Each return code is a 4-digit hexadecimal value. The first two digits contain the *major code*, and the last two digits contain the *minor code*.

Notes:

1. In the return code descriptions, *your program* refers to the AS/400 application program that issues the operation and receives a return code from ICF. The *target program* refers to the application program on the remote system with which your program is communicating through ICF.
2. Each communications programming manual provides detailed information about every return code for the communications type and the recovery actions that should be taken.
3. Certain return codes describe the turnaround indication, which is not applicable to asynchronous, retail and finance communications. Also, not all return codes have the same meaning for all communications types.

Major Code 00

A major return code 00 indicates that the 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.

Table B-2. Major Code 00

Code	APPC	Asyn-chronous	BSCEL	Finance	Intra-system	Retail	SNUF
0000	X	X	X	X	X	X	X
0001	X		X	X	X	X	X
0003				X	X	X	X
0004		X			X		X
0005					X	X	X
0007				X	X	X	X
0008	X		X		X		X
000C							X
0010	X		X		X		X
0014	X				X		
0015	X				X		
0016		X					
0017					X		
001C	X				X		
0020			X				X
0021			X				X
0023							X
0025							X
0027							X
0028			X				X
0030			X				X
0031			X				X
0033							X
0035							X
0037							X
0038			X				X
0042		X					
0044					X		
0045					X		
0047					X		

Code	Description
0000	Turnaround or end-of-transmission indication and data received on successful input operation, or an output operation was successful.
0001	Successful input operation. A turnaround or end-of-transmission indication was not received. Continue to receive.
0003	End-of-group indication received on successful input operation.
0004	Function-management-header and turnaround indications received on successful input operation, or a PAD message received from a remote PAD.
0005	Function-management-header indication received on successful input operation.
0007	Function-management-header and end-of-group indications received on successful input operation.
0008	Detach indication received on successful input operation.
000C	Function-management-header and detach indications received on successful input operation.
0010	Request-to-write, reverse-interrupt (RVI), or request-to-change-direction received on successful output operation.
0014	Turnaround indication received on successful input operation. In addition, the remote system requested confirmation.
0015	Remote system requested confirmation on successful input operation. The local application program continues to receive data.
0016	Parity error or stop bit error (framing) or both received on successful input operation.
0017	End-of-group indication received on successful input operation. In addition, the remote system requested confirmation.
001C	Detach indication received on successful input operation. In addition, the remote system requested confirmation.
0020	System message and turnaround or end-of-transmission indication received on a successful input operation.
0021	System message received on successful input operation. Continue to receive.
0023	System message with end-of-group indication received.
0025	Function-management-header indication received with system message.
0027	System message received with function-management-header and end-of-group indications.
0028	System message and detach indication received on successful input operation.
0030	Truncated system message and turnaround or end-of-transmission indication received on successful input operation.
0031	Truncated system message received on successful input operation. Continue to receive.
0033	Truncated system message received with end-of-group indication.

0035 Truncated system message received with function-management-header indication.

0037 Truncated system message received with function-management-header and end-of-group indications.

0038 Truncated system message and detach indication received on successful input operation.

0042 Some data was lost on successful input operation.

0044 Function-management-header and turnaround indications received on successful input operation. In addition, the remote system requested confirmation.

0045 Function-management-header indication received on successful input operation. In addition, the remote system requested confirmation.

0047 Function-management-header and end-of-group indications received on successful input operation. In addition, the remote system requested confirmation.

Major Code 02

The major return code 02 indicates that the input operation completed successfully, but your job is being ended (controlled).

Description

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

Table B-3 (Page 1 of 2). Major Code 02

Code	APPC	Asyn-chronous	BSCEL	Finance	Intra-system	Retail	SNUF
0200	X	X		X	X	X	X
0201	X		X		X		X
0203				X	X	X	X
0204		X			X		X
0205					X	X	X
0207				X	X	X	X
0208	X		X		X		X
020C							X
0214	X				X		
0215	X				X		
0216		X					
0217					X		
021C	X				X		
0220			X				X
0221			X				X
0223							X

Table B-3 (Page 2 of 2). Major Code 02

Code	APPC	Asyn-chronous	BSCEL	Finance	Intra-system	Retail	SNUF
0225							X
0227							X
0228			X				X
0230			X				X
0231			X				X
0233							X
0235							X
0237							X
0238			X				X
0242		X					
0244					X		
0245					X		
0247					X		

Code **Description**

0200 On a successful input operation, a turnaround indication or data that is the beginning or middle record of a group of records was received. In addition, a job ended (controlled) indication was received.

0201 Data with job ended (controlled) indication received on a successful input operation. A turnaround indication was not received. Continue to receive.

0203 End-of-group indication received with job ended (controlled) indication on successful input operation.

0204 Function-management-header and turnaround indications or a PAD message from a remote PAD was received with job ended (controlled) indication on a successful input operation.

0205 Function-management-header indication received with job ended (controlled) indication on successful input operation.

0207 Function-management-header and end-of-group indications received with job ended (controlled) indication on successful input operation.

0208 Detach indication received with job ended (controlled) indication on successful input operation.

020C Function-management-header and detach indications received with job ended (controlled) indication on successful input operation.

0214 Turnaround indication received with job ended (controlled) indication on successful input operation. In addition, the remote system requested confirmation.

0215 Remote system requested confirmation. The local application program continues to receive data. Job ended (controlled) indication received.

0216 Parity error or stop bit error (framing) or both received with job ended (controlled) indication on successful input operation.

0217 End-of-group indication received with job ended (controlled) indication on successful input operation. In addition, the remote system requested confirmation.

021C Detach indication received with job ended (controlled) indication on successful input operation. In addition, the remote system requested confirmation.

0220 Remote system message and turnaround or end-of-transmission indication received with job ended (controlled) indication on successful input operation.

0221 Remote system message received with job ended (controlled) indication on successful input operation. The session is still in receive state.

0223 Remote system message with end-of-group and job ended (controlled) indications received on successful input operation.

0225 Function-management-header indication received with system message and job ended (controlled) indication on a successful input operation.

0227 System message received with function-management-header, end-of-group, and job ended (controlled) indications on a successful input operation.

0228 System message and detach indication received with job ended (controlled) indication on a successful input operation.

0230 Truncated system message and turnaround or end-of-transmission indication received with job ended (controlled) indication on successful input operation.

0231 Truncated system message received with job ended (controlled) indication on a successful input operation. The session is still in receive state.

0233 Truncated system message received with end-of-group and job ended (controlled) indications on a successful input operation.

0235 Truncated system message received with function-management-header and job ended (controlled) indications on a successful input operation.

0237 Truncated system message received with function-management-header, end-of-group, and job ended (controlled) indications on a successful input operation.

0238 Truncated system message and detach indication received with job ended (controlled) indication on successful input operation.

0242 Data-loss indication received with job ended (controlled) indication on successful input operation.

0244 Function-management-header and turnaround indications with job ended (controlled) indication received on a successful input operation. In addition, the remote system requested confirmation.

0245 Function-management-header with job ended (controlled) indication received on a successful input operation. In addition, the remote system requested confirmation.

0247 Function-management-header and end-of-group indications with job ended (controlled) indication received on a successful input operation. In addition, the remote system requested confirmation.

Major Code 03

Major return code 03 indicates that the input operation completed successfully, but no data was received.

Description

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

Table B-4. Major Code 03

Code	APPC	Asyn-chronous	BSCEL	Finance	Intra-system	Retail	SNUF
0300	X	X	X		X	X	X
0301	X		X		X		X
0302		X			X		
0303				X	X	X	X
0308	X		X		X	X	X
0309	X	X	X	X	X	X	X
0310	X	X	X	X	X	X	X
0314	X				X		
0315	X				X		
0317					X		
031C	X				X		

Code **Description**

0300 On a successful input operation, a turnaround or end-of-transmission indication with no data, or a null record that was the beginning or middle record in a group of records was received.

0301 No data received on successful input operation. A turnaround indication was not received. Continue to receive.

0302 Fail indication received with no data on successful input operation.

0303 End-of-group indication received with no data on successful input operation.

0308 Detach indication received with no data on successful input operation.

0309 Job ended (controlled) indication received on read-from-invited-program-devices operation.

0310 Timer interval has ended.

0312 No data was received, but data from other sources is waiting to be received.

0314 Turnaround indication received with no data on a successful input operation. In addition, the remote system requested confirmation.

0315 Remote system requested confirmation. The local application program continues to receive data.

0317 End-of-group indication received with no data on successful input operation. In addition, the remote system requested confirmation.

031C A detach indication received with no data on a successful input operation. In addition, the remote system requested confirmation.

Major Code 04

Major return code 04 indicates that an output exception occurred.

Description

An output exception occurred because your program attempted to send data when it should be receiving data. The data from your output operation was not sent. You can attempt to send data later.

Table B-5. Major Code 04

Code	APPC	Asyn-chronous	BSCEL	Finance	Intra-system	Retail	SNUF
0402					X		
0411			X				
0412	X	X	X	X	X	X	X

Code **Description**

0402 Fail indication received. Your program must receive.

0411 Message for your program is waiting to be received.

0412 Data for your program is waiting to be received, or a negative response has been received from the remote system, and your program has not been informed.

Major Codes 08-11

Major return codes 08-11 indicate that 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.

Table B-6. Major Codes 08-11

Code	APPC	Asyn-chronous	BSCEL	Finance	Intra-system	Retail	SNUF
0800	X	X	X	X	X	X	X
1100	X	X	X	X	X	X	X

Code **Description**

0800 Acquire operation was not successful because your program tried to acquire a program device that has already been acquired.

1100 Read-from-invited-program-devices operation was not successful.

Major Code 34

Major return code 34 indicates that an input exception occurred.

Description

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

Table B-7. Major Code 34

Code	APPC	Asyn-chronous	BSCEL	Finance	Intra-system	Retail	SNUF
3401			X	X	X	X	X
3431	X						
3441	X	X	X	X	X	X	X
3451	X	X	X	X	X	X	X
3461	X						

Code **Description**

3401 Input operation rejected because data received was too long for your program's input buffer.

3431 An input exception occurred because the program received data that exceeded its maximum record length. The data has been truncated. The local application program continues to receive data.

3441 The record format selected by the format selection option does not match the record format specified on the read.

3451 The file record size specified is not large enough for the data and/or indicators received.

3461 Partial record received because remote system sent an error condition before completing the record.

Major Code 80

Major return code 80 indicates a permanent system or file error (nonrecoverable).

Description

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

Table B-8. Major Code 80

Code	APPC	Asyn-chronous	BSCEL	Finance	Intra-system	Retail	SNUF
8081	X	X	X	X	X	X	X
8082	X	X	X	X	X	X	X
80B3	X	X	X	X	X	X	X
80C0	X						
80D0	X						
80EB	X	X	X	X	X	X	X
80ED	X	X	X	X	X	X	X
80EF	X	X	X	X	X	X	X
80F8	X	X	X	X	X	X	X

Code **Description**

8081 System error abnormally ended the support provided by the communications type.

8082 Communications device not usable or error recovery canceled by operator.

80B3 ICF file not available.

80C0 Session failed.

80D0 Remote transaction program not available. No retry allowed.

80EB Open operation was tried but was not successful. Either an open option was specified that was not valid, or there is a mismatch between the file and the program.

80ED File level check error occurred on open operation.

80EF User not authorized to file.

80F8 Open operation not successful because the file is already open or it is in error.

Major Code 81

Major return code 81 indicates a permanent session error (nonrecoverable).

Description

A nonrecoverable session error occurred during an I/O operation. Your session cannot continue and has ended. Before communications can resume, the session must be established 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.

Table B-9. Major Code 84

Code	APPC	Asyn-chronous	BSCEL	Finance	Intra-system	Retail	SNUF
810A			X				
8140	X	X	X	X	X	X	X
8187			X				
8191	X	X	X	X		X	X
8192			X				
8193			X				
8194			X				
8196	X						X
8197	X		X	X		X	
8198			X				
8199			X				
819A			X				
819C			X				
819D			X				X
81A3				X		X	
81A4				X		X	
81AD				X		X	
81B9							X
81BA				X		X	
81C2	X						
81C5	X						
81C6	X						
81E9	X	X	X	X	X	X	X

Code **Description**

810A Combination of values detected on an input or output operation was not valid. Both CODE(ASCII) and TRNSPY(*YES) were specified.

8140 Cancel reply was received for a previous inquiry or notify message.

8187 Block length or record length is greater than buffer size on an input or output operation.

8191 Permanent line error occurred on an output operation, or station (controller) error occurred on an input or output operation.

8192 Permanent line error occurred on an input operation.

8193 Disconnect indication (for switched lines only) received on an output operation.

8194 Disconnect indication (for switched lines only) received on an input operation.

8196 Communications support has ended the session.

8197 Remote system abnormally ended the session on an output operation.

8198 Remote system abnormally ended the session on an input operation.

8199 Time between successive data blocks sent to, or received by, the remote system on output operations is larger than specified wait time.

819A Time between successive data blocks received from the remote system on input operations is larger than specified wait time.

819C On an input operation, the length of the data block sent by the remote system was greater than the buffer size.

819D Unexpected data or an unexpected program start request was received from the remote system during an active session.

81A3 SNA session ended abnormally.

81A4 SNA protocol violation occurred.

81AD Attempt to establish an SNA session was not successful. The SDLC frame size was not large enough to contain the response unit (RU) size.

81B9 A data record that exceeds the maximum user record length was received on an input operation.

81BA A data record that exceeds the maximum user record length was received on an input operation.

81C2 Operation failed because the local APPC could not establish a session.

81C5 The remote program or remote system abnormally ended the session (TYPE = SVC).

81C6 The remote program or remote system abnormally ended the session (TYPE = TIMER).

81E9 Data received does not match any record format in the file with the RECID keyword.

Major Code 82

Major return code 82 indicates that the open or acquire operation failed.

Description

Your attempt to establish a session was not successful. The error may be recoverable or permanent, and recovery from it is unlikely until the problem causing the error is detected and corrected.

Table B-10 (Page 1 of 2). Major Code 02

Code	APPC	Asyn-chronous	BSCEL	Finance	Intra-system	Retail	SNUF
8209	X	X	X	X	X	X	X
820A			X				
8221				X		X	
8233	X	X	X	X	X	X	X
8281	X	X	X	X	X	X	X
8282	X	X	X	X	X	X	X
8285		X					X
8287			X				
8289			X				
828B			X				
828C			X				
828D			X				
828E			X				
8290			X				
8291			X	X		X	X
8293			X				
8297			X	X		X	
82A0			X				
82A1							X
82A2				X			
82A4				X		X	
82A5							X
82A6	X			X		X	X
82A7			X	X		X	X
82A8	X	X	X	X	X	X	X
82A9	X	X	X	X	X	X	X
82AA	X	X	X	X	X	X	X
82AB	X	X	X	X	X	X	X
82AD				X		X	
82B3	X	X	X	X		X	X

Table B-10 (Page 2 of 2). Major Code 02							
Code	**APPC**	**Asyn-chronous**	**BSCEL**	**Finance**	**Intra-system**	**Retail**	**SNUF**
82B4							X
82B5							X
82BB							X
82C3	X						
82EA	X	X	X	X	X	X	X
82EC	X			X		X	
82EE	X	X	X	X	X	X	X
82EF	X	X	X	X	X	X	X
82F0	X	X					
82F2	X						
82F4	X			X	X	X	
82F5		X	X				X

Code **Description**

8209 An open or acquire operation was not successful because a prestart job is to be ended.

820A Combination of values detected was not valid. Both CODE(ASCII) and TRNSPY(*YES) were specified, or BLOCK(*USER) and RMTBSCEL(*YES) were specified.

8221 SNA command received for remote location or device description that was not supported or not valid.

8233 Program device name is either missing or not valid.

8281 System error abnormally ended the support provided by the communications type.

8282 Communications device not usable or error recovery canceled.

8285 Attempt to automatically call remote system failed.

8287 Block Length or record length greater than buffer size.

8289 Combination of values detected during an acquire operation was not valid. A record separator and text transparency were both specified.

828B Combination of values detected during an acquire operation was not valid. The maximum user record length specified was greater than the block length.

828C Combination of values detected during an acquire or open operation was not valid. GRPSEP(*DEV3740) and BLOCK(*ITB) were both specified.

828D Combination of values detected during an acquire or open operation was not valid. TRUNC(*YES) and BLOCK(*ITB) were both specified.

828E Combination of values detected during an acquire or open operation was not valid. TRUNC(*YES) and BLOCK(*ITB) were both specified, or TRUNC(*YES) and BLOCK(*NOSEP) were specified.

8290 Combination of values detected during an acquire or open operation was not valid. Blank compression and text transparency were both specified.

8291 Permanent line error or station (controller) error occurred on an unsuccessful open or acquire operation.

8293 Disconnect indication (for switched lines only) received from remote system during an acquire or open operation.

8297 Remote system ending line transmission.

82A0 A record separator character that was not valid was specified on the ADDICFDEVE or OVRICFDEVE command.

82A1 Logon portion of the acquire operation failed. Either the host subsystem was not active, or a remote program name that was not valid was specified in the APPID parameter.

82A2 User ID or password that was not valid was received on the INIT-SELF.

82A4 SNA protocol violation occurred.

82A5 Combination of parameter values detected during an acquire operation was not valid. *YES was specified for the MSGPTC and BATCH parameters.

82A6 SNA bind command failed.

82A7 The specified program device was already in use when the open or acquire operation was attempted.

82A8 The maximum number of program devices allowed for the ICF file was reached when the open or acquire operation was attempted.

82A9 Acquire to requesting program device rejected because *REQUESTER device was not available or was already acquired.

82AA Operation failed because remote location or device not found, or device was found but not usable.

82AB Operation failed because device not varied on.

82AD Attempt to establish an SNA session was not successful. The SDLC frame size was not large enough to contain the RU size.

82B3 Operation failed because device is being used by a different job or no sessions are currently available for specified remote location.

82B4 Operation failed because System/36 application program cannot open an ICF file to a program device for SNA 3270 API.

82B5 Operation not successful because SNA 3270 API session cannot use SNUF device from earlier release.

82BB Acquire operation failed because device specified was reserved for a program start request from host system.

82C3 Mode description was not found.

82EA *RECID format selection processing was requested to a file that contains no record formats with a *RECID keyword.

82EC The acquire operation was not successful because CNVTYPE(*USER) is not valid with FMTSLT(*RMTFMT); or this communications type does not support FMTSLT(*RMTFMT).

82EE An operation was attempted to a device that is not supported for an ICF file.

82EF An open or acquire operation was attempted to a device the user is not authorized to use, or to a device in service mode.

82F0 Error recovery not performed for file.

82F2 Conversation type specified for the requesting program device does not match value received from source program.

82F4 Open operation for input only not valid for a source program.

82F5 *RMTFMT format selection parameter not valid on acquire operation.

Major Code 83

Major return code 83 indicates that a 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.

Table B-11 (Page 1 of 2). Major Code 83

Code	APPC	Asyn-chronous	BSCEL	Finance	Intra-system	Retail	SNUF
830B	X	X	X	X	X	X	X
830C							X
830D							X
8311							X
8316	X						
8319				X	X	X	X
831A			X		X	X	
831B				X	X	X	X
831C		X	X	X	X	X	X
831E	X	X	X	X	X	X	X
831F	X	X	X	X	X	X	X
8322	X		X	X	X	X	X
8323				X	X		X
8324							X
8326				X	X	X	X
8327	X		X		X	X	X
8329	X	X	X		X	X	X
832A					X		
832B			X				
832C	X	X	X	X	X	X	X
832D	X	X	X	X	X	X	X
832F	X		X	X	X	X	X
8330					X	X	X

Table B-11 (Page 2 of 2). Major Code 83

Code	APPC	Asyn-chronous	BSCEL	Finance	Intra-system	Retail	SNUF
8331					X		X
8332							X
8334	X		X		X	X	
83B6				X		X	X
83B7							X
83B8							X
83C7	X						
83C8	X						
83C9	X						
83CA	X						
83CB	X						
83CC	X						
83CD	X				X		
83CE	X						
83CF	X						
83D0	X						
83D1	X						
83D2	X						
83D3	X						
83D5	X						
83D6	X				X		
83E0	X	X	X	X	X	X	X
83E8		X	X	X	X	X	X
83F1	X						
83F3	X						
83F6			X				
83F7			X				
83F8	X	X	X	X	X	X	X
83F9	X						

Code	Description
830B	An input or output operation was attempted to a program device that was not acquired.
830C	Length of function-management-header received from host system is greater than the maximum RU length.
830D	Shutdown indication received from host system.
8311	Output operation was attempted while a message containing sense data was waiting to be received.

8316 Evoke failed because target program was not found.

8319 Negative response with sense data was issued to your program's previous or current output request by the other program.

831A Evoke failed to complete successfully, or the target program ended abnormally.

831B Sense data that was not valid was specified on a negative-response function issued by your program.

831C Operation is not valid at this time. An indication was received that the return code from a previous operation was not properly handled by your program.

831E Operation or combination of operations not valid, or operation not supported by the communications type.

831F Length of data record or data specified on the operation not valid.

8322 Request-to-write, negative-response, or detach function not valid while your program is in send state.

8323 Cancel issued while in receive state, or fail indication received while in send state.

8324 Function-management-header indication issued by your program at wrong time. Function-management-header is valid only with the *first* record in the chain.

8326 Cancel or negative-response indication issued as a single record. These functions are only valid within a group of records.

8327 Input or output operation that was not valid was issued when no transaction existed. Your program may have expected more data when there was none.

8329 Evoke not valid in this session. Your program was started by a remote program start request.

832A Both programs were attempting to receive.

832B Output operation with zero record length detected when GRPSEP(*OFCSYS) was specified.

832C Release operation not valid after an invite function.

832D Attempted function is not valid following an invite function.

832F Evoke function or release operation that was not valid was issued before a transaction completed.

8330 Cancel indication or cancel and turnaround indications received on an input operation.

8331 Cancel indication received *without* turnaround indication on an input operation.

8332 Cancel and detach indications received on an input operation.

8334 Program name missing on an evoke sent by your program, or the length of the program name was not valid.

83B6 The remote program has quiesced the SNA session on which this transaction is running.

83B7 Value received in SNA header that was not valid.

83B8 The host system has sent a clear request to reset the session.

83C7 Fail indication (TYPE=PROG) received with no data on an input operation. No data truncated.

83C8 Fail indication (TYPE=SVC) received with no data on an input operation. No data truncated.

83C9 Fail indication (TYPE=PROG) received on an input or output operation with or without a confirm indication. Data may have been lost.

83CA Fail indication (TYPE=SVC) received on an input or output operation with or without a confirm indication. Data may have been lost.

83CB Fail indication (TYPE=PROG) received on an input operation. The last logical record truncated.

83CC Fail indication (TYPE=SVC) received on an input operation. The last logical record truncated.

83CD Confirm indication not allowed when SYNLVL(*NONE) is specified on the evoke function.

83CE Security information specified on the evoke function not valid. Request rejected by remote system.

83CF Remote location or remote program does not support the specified conversation type. Request rejected by remote system.

83D0 Program name specified on the evoke function is not currently available. Retry is allowed.

83D1 Program initialization parameters not allowed. Request rejected by remote system.

83D2 Program initialization parameters not specified correctly. Request rejected by remote system.

83D3 Synchronization level specified on the evoke function not supported by remote program.

83D5 Response-to-confirm request required.

83D6 Response-to-confirm request not valid in current state.

83E0 Record format not defined for the file.

83E8 Cancel-invite function not valid because an invite function was not previously issued.

83F1 The file was closed while the transaction was still active.

83F3 Length specification on a basic conversation not valid.

83F6 User-defined data not valid on an unsuccessful output operation.

83F7 Length of user-blocked data record not valid on an unsuccessful output operation.

83F8 Operation attempted to a device marked in error.

83F9 Your program issued an operation that did not complete the data record.

Failed Program Start Requests

Message CPF1269 is sent to the system operator message queue when the local system rejects an incoming program start request. You can use the message information to determine why the program start request was rejected.

The CPF1269 message contains two reason codes. One of the reason codes can be zero, which can be ignored. If only one nonzero reason code is received, that reason code represents the reason the program start request was rejected. If the System/36 environment is installed on your AS/400 system, there can be two nonzero reason codes. These two reason codes occur when the OS/400 cannot determine whether the program start request was to start a job in System/36 environment or in the OS/400 environment. One reason code explains why the program start request was rejected in the System/36 environment and the other explains why the program start request was rejected in the OS/400 environment. Whenever you receive two reason codes, you should determine which environment the job was to run in and correct the problem for that environment.

Table B-12 describes reason codes for failed program start requests.

Table B-12 (Page 1 of 5). Reason Codes for Rejected Program Start Requests

Reason Code	Reason Description	Communications Types
401	Program start request received to a device that is not allocated to an active subsystem.	All
402	Requested device is currently being held by a Hold Communications Device (HLDCMNDEV) command.	All except APPC
403	User profile is not accessible.	All
404	Job description is not accessible.	All
405	Output queue is not accessible.	All
406	Maximum number of jobs defined by subsystem description are already active.	All
407	Maximum number of jobs defined by communications entry are already active.	All
408	Maximum number of jobs defined by routing entry are already active.	All
409	Library on library list is exclusively in use by another job.	All
410	Group profile cannot be accessed.	All
411	Insufficient storage in machine pool to start job.	All
501	Job description was not found.	All
502	Output queue was not found.	All
503	Class was not found.	All
504	Library on initial library list was not found.	All
505	Job description or job description library is damaged.	All
506	Library on library list is destroyed.	All
507	Duplicate libraries were found on library list.	All

Table B-12 (Page 2 of 5). Reason Codes for Rejected Program Start Requests

Reason Code	Reason Description	Communications Types
508	Storage-pool defined size is zero.	All
602	Transaction program-name value is reserved but not supported.	All
604	Matching routing entry was not found.	All
605	Program was not found.	All
704	Password is not valid.	All except Retail and Finance
705	User is not authorized to device.	All except Finance
706	User is not authorized to subsystem description.	All
707	User is not authorized to job description.	All
708	User is not authorized to output queue.	All
709	User is not authorized to program.	All
710	User is not authorized to class.	All
711	User is not authorized to library on library list.	All
712	User is not authorized to group profile.	All
713	User ID is not valid.	All except Retail and Finance
714	Default user profile is not valid.	All except Finance
715	Neither password nor user ID was provided, and no default user profile was specified in the communications entry.	All except Finance
718	No user ID.	All except Retail
722	A user ID was provided, but no password was sent.	All except Retail and Finance
723	No password was associated with the user ID.	All except Finance
725	User ID does not follow naming convention.	All except Retail and Finance
801	Program initialization parameters are present but not allowed.	All except Retail
802	Program initialization parameter exceeds 2000 bytes for a prestart job.	All except Retail
803	Subsystem is ending.	All
804	Prestart job is inactive or is ending.	All
805	WAIT(*NO) was specified on the prestart job entry and no prestart job was available.	All
806	The maximum number of prestart jobs that can be active on a prestart job entry was exceeded.	All
807	Prestart job ended when a program start request was being received.	All
901	Program initialization parameters are not valid.	All except Retail
902	Invalid number of parameters for program.	All

Table B-12 (Page 3 of 5). Reason Codes for Rejected Program Start Requests

Reason Code	Reason Description	Communications Types
903	Program initialization parameters required but not present.	All
1001	System logic error. Function check or unexpected return code encountered.	All
1002	System logic error. Function check or unexpected return code encountered while receiving program initialization parameters.	All
1501	Character in procedure name is not valid.	All
1502	Procedure not found.	All
1503	S/36 environment library not found.	All
1504	Library QSSP not found.	All
1505	File QS36PRC not found in library QSSP.	All
1506	Procedure or library name is greater than 8 characters.	All
1507	Current library not found.	All
1508	Not authorized to current library.	All
1509	Not authorized to QS36PRC in current library.	All
1510	Not authorized to procedure in current library.	All
1511	Not authorized to S/36 environment library.	All
1512	Not authorized to file QS36PRC in S/36 environment library.	All
1513	Not authorized to procedure in S/36 environment library.	All
1514	Not authorized in library QSSP.	All
1515	Not authorized to file QS36PRC in QSSP.	All
1516	Not authorized to procedure in QS36PRC in QSSP.	All
1517	Unexpected return code from S/36 environment support.	All
1518	Problem phase program not found in QSSP.	All
1519	Not authorized to problem phase program in QSSP.	All
1520	Maximum number of target programs started (100 per S/36 environment).	All
1901	The record or block size exceeds maximum size.	BSCEL
1902	ASCII and transparency are mutually exclusive.	BSCEL
1903	Transparent mode and blank compression conflict.	BSCEL
1904	Block length is required with data format.	BSCEL
1905	Blank truncation and ITB mode conflict.	BSCEL
1906	Blank compression and ITB mode conflict.	BSCEL
1907	3740 multiple files and ITB mode conflict.	BSCEL

Table B-12 (Page 4 of 5). Reason Codes for Rejected Program Start Requests

Reason Code	Reason Description	Communications Types
1908	Record separator mode and transparent modes conflict.	BSCEL
1909	Record separator and ITB mode conflict.	BSCEL
1910	The record length exceeds the block length.	BSCEL
1911	Record separator character is not valid.	BSCEL
1912	BLOCK(*USER) and RMTBSCEL(*YES) conflict.	BSCEL
1913	BLOCK(*NOSEP) and TRUNC(*YES) conflict.	BSCEL
1914	Program name is not valid.	BSCEL
1915	Program start request record was too long.	BSCEL
2001	*FMH5* field length is not correct.	APPC
2002	Concatenation code is not valid.	APPC
2003	Function-management-header type is not 5.	APPC
2004	Command code field in function-management-header is not valid.	APPC
2005	Length for fixed-length fields is not valid.	APPC
2006	Conversation type is not supported.	APPC
2007	Synchronization level is not supported.	APPC
2008	Reconnection is not supported.	APPC
2009	Transaction program name field length is not valid.	APPC
2010	Access code subfield length is not valid.	APPC
2011	*UOW-ID* subfield length is not valid.	APPC
2012	*UOW-ID* contents are not valid.	APPC
2013	Requested device is currently being held by a Hold Communications Device (HLDCMNDEV) command.	APPC
2014	Transaction program name value is reserved but not supported.	APPC
2015	LU service request received but the LU service job is not active.	APPC
2016	Pre-verified user ID received, but device description specifies SECURELU(*NO).	APPC
2017	No user ID was provided, but a password was received.	APPC
2018	No user ID was provided, but a user profile was received.	APPC
2019	Remote system indicated it sent a pre-verified user ID, but no user ID was received.	APPC
2020	Remote system sent a pre-verified user ID, but also sent a password.	APPC
2021	Remote system sent a user ID (which it had not verified) and failed to send a password.	APPC

Table B-12 (Page 5 of 5). Reason Codes for Rejected Program Start Requests

Reason Code	Reason Description	Communications Types
2022	Password received, but this is a nonsecure system.	APPC
2111	Program name missing or not valid.	SNUF
2118	Function-management-header not valid.	SNUF
2123	End bracket or end chain missing.	SNUF
2501	System logic error. Function check or unexpected return code encountered while processing a program start request.	Intrasystem
2502	Temporarily unable to allocate needed resources for a program start request.	Intrasystem
2503	No subsystem accepting program start requests for this device.	Intrasystem
2601	Program name missing and device is not configured for display station pass-through.	Retail
2651	*EXEC not specified.	Finance
2652	Blank missing after *EXEC.	Finance
2653	Program name missing.	Finance
2654	Program name greater than 10 characters.	Finance
2655	Library name greater than 10 characters.	Finance

Appendix C. Open Feedback and I/O Feedback

This appendix contains information concerning the open feedback area and the input/output (I/O) feedback area.

Note: If you are using the RPG/400 support, you need to add the following RPG/400 offset values to the offset values listed in this appendix in order to access information from the feedback areas.

Table C-1. Offset Values for RPG/400

Open Feedback Area	Common I/O Feedback Area	File Dependent Feedback Area
81	241	367

Open Feedback Area

The open feedback area contains information about the ICF file. You can use this information, set during open processing, as long as the file is open. The support provided by the high-level language you are using determines how to access this information. See the appropriate language reference manual for more information.

The complete open feedback area is described in the *Data Management Guide*. Table C-2 shows the fields relevant to ICF support.

Table C-2. Open Feedback Area

Offset	Data Type	Length in Bytes	Contents
0	Character	2	Type of file being opened (DS = device file).
2	Character	10	Name of the file being opened.
12	Character	10	Name of the library containing the file.
44	Binary	2	Record length.
66	Binary	2	File type (11 = ICF).
116	Character	10	Program device name of the requester. This field contains the program device name, if valid for this job and file. If it is not valid, this field contains *N.
146	Binary	2	Number of program devices added to the file using the Add Intersystem Communications Function Device Entry (ADDICFDEVE) command, or number of program devices defined to the file using the Override ICF Device Entry (OVRICFDEVE) command and acquired by the program.
148	Character	Variable	Program device name definition list. Refer to Table C-3.

Program Device Definition List

Table C-3 shows the mapping for a single entry of the program device definition list, which is physically a part of the open feedback area. However, the fields in the definition list are not necessarily set when the file opens.

Table C-3 (Page 1 of 2). Program Device Definition List

Offset	Data Type	Length in Bytes	Contents
0	Character	10	Program device name.
60	Character	10	Device description name.
70	Character	1	Device class. Hex 0B for ICF.
71	Character	1	Communications type: Hex 0A Binary synchronous communications equivalence link (BSCEL) device Hex 0E Advanced Program-to-Program Communications (APPC) device Hex 1E Intrasystem device Hex 1F Asynchronous device Hex 20 System network architecture upline facility (SNUF) device Hex 42 Finance Hex 43 Retail
76	Character	2	Status flags, as follows: Bit 3, acquire status: 0 Program device not acquired. 1 Program device acquired. Bit 4, invite status: 0 Program device not invited. 1 Program device invited. Bit 5, data-available status: 0 Data not available. 1 Data available. Bit 6, transaction status: 0 Transaction not started. An evoke has not been sent, a detach has been sent or received, or the transaction has completed. 1 Transaction started. The session is active, an evoke has been sent or received, and the transaction has not ended. Bit 7, session type: 0 Session created with source program 1 Requesting program device

Table C-3 (Page 2 of 2). Program Device Definition List

Offset	Data Type	Length in Bytes	Contents	
78	Character	1	Synchronization level:	
			Hex 00	The transaction was started with SYNLVL(*NONE). Confirm processing is not allowed.
			Hex 01	The transaction was started with SYNLVL(*CONFIRM). Confirm processing is allowed.
79	Character	1	Conversation type:	
			Hex D0	Basic
			Hex D1	Mapped

Input/Output Feedback Area

The results of I/O operations are communicated to the program using ICF messages and I/O feedback information. The I/O feedback area is updated for every input/output operation. The support provided by the high-level language you are using determines how to access this information. See the appropriate communications language reference manual for more information.

The I/O feedback area consists of two parts:

- A common I/O feedback area
- A file-dependent I/O feedback area

Common I/O Feedback Area

The complete common I/O feedback area is described in the *Data Management Guide*. Table C-4 shows the fields relevant to ICF.

Table C-4 (Page 1 of 2). Common I/O Feedback Area

Offset	Data Type	Length in Bytes	Contents
0	Binary	2	Offset to device-dependent feedback area. Refer to Table C-5.
2	Binary	4	Output operation count. Updated only when a output operation completes successfully. This field is not updated for fail, request-to-write, cancel, or negative-response functions, or if the record length is 0.
6	Binary	4	Input operation count. Updated only when an input operation completes successfully and data is received.
10	Binary	4	Output then input operation count. Updated only when a combined output then input operation completes successfully.
14	Binary	4	Count of other operations. Number of successful acquire and release operations.
18	Character	1	Reserved.
19	Character	1	Current operation (the last requested): Hex 01 Input Hex 05 Output Hex 06 Output then Input Hex 11 Release Hex 12 Acquire
20	Character	10	Name of the just-processed record format, which is either specified on the I/O request or determined by default processing.

Table C-4 (Page 2 of 2). Common I/O Feedback Area

Offset	Data Type	Length in Bytes	Contents
30	Character	2	Communications class and type: Hex 0B0A BSCEL Hex 0B0E APPC Hex 0B1E Intrasystem Hex 0B1F Asynchronous Hex 0B20 SNUF Hex 0B42 Finance Hex 0B43 Retail
32	Character	10	Program device name (for operation just completed).
42	Binary	4	Record length specified by the record format processed by the last I/O operation.
46	Reserved	98	Not applicable to communications.

File-Dependent I/O Feedback Area

Table C-5 shows the communications type-dependent fields relevant to ICF.

Table C-5 (Page 1 of 2). File-dependent I/O Feedback Area

Offset	Data Type	Length in Bytes	Contents	ICF Type
5	Binary	4	Actual record length. This field is set as follows: Input: Contains the actual length of user data received from the remote system or device. When all the data cannot be contained in the record format used, the length of data is provided, if known. If the actual length cannot be determined, this field is set to hex FFFFFFFF. When a partial record is received, the length of the data received is provided. If the input operation completes with an error (other than partial record or buffer too small), the contents of the field are undetermined. Output: Contains the number of bytes moved from your program to the output buffer. If the output operation completes with an error, the contents of the field are undetermined.	All
34	Character	2	Major return code.	All
36	Character	2	Minor return code.	All

Table C-5 (Page 2 of 2). File-dependent I/O Feedback Area

Offset	Data Type	Length in Bytes	Contents	ICF Type
38	Character	8	Negative-response error data. For some return codes, this field can contain more detailed information about the reason for the error.	APPC Finance Retail
46	Character	1	Safe indicator: 0 Off. 1 On indicates that a block ending with ETX was received. The Safe indicator is not set for BLOCK(*USER).	BSCEL
47	Character	1	Reserved.	
48	Character	1	Request-to-write indication was received. 0 Request-to-write not received. 1 Request-to-write was received.	APPC SNUF BSCEL Intrasystem
49	Character	10	Remote format name received from the remote program on an input operation.	APPC Intrasystem
63	Character	8	Mode associated with the program device.	APPC

Appendix D. EBCDIC and ASCII Character Sets

The following charts show the EBCDIC and ASCII character sets. The charts are provided to show the data link control characters that are used in data communications.

EBCDIC Character Set

Figure D-1 shows a complete EBCDIC character set.

		Main Storage Bit Positions 0,1,2,3															
Main Storage Bit Positions		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
4,5,6,7	**Hex**	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	NUL	DLE	DS		SP	&	-						{	}	\	0
0001	1	SOH	DC1	SOS		RSP		/		a	j	~		A	J	NSP	1
0010	2	STX	DC2	FS	SYN					b	k	s		B	K	S	2
0011	3	ETX	DC3	WUS	IR					c	l	t		C	L	T	3
0100	4	SEL	ENP/RES	INP/BYP	PP					d	m	u		D	M	U	4
0101	5	HT	NL	LF	RS					e	n	v		E	N	V	5
0110	6		BS	ETB	NBS					f	o	w		F	O	W	6
0111	7	DEL	POC	ESC	EOT					g	p	x		G	P	X	7
1000	8	GE	CAN		SBS					h	q	y		H	Q	Y	8
1001	9	SPS	EM		IT				'	i	r	z		I	R	Z	9
1010	A	RPT	UBS	SM/SW	RFF	¢	!	|	:					SHY			
1011	B	VT	CU1	FMT	CU3	.	$	,	#								
1100	C	FF	IFS			<	•	%	@					⌐		⌐	
1101	D	CR	IGS	ENQ	NAK	(	)/DC4	—	'								
1110	E	SO	IRS	ACK		+	;	>	=					⌐			
1111	F	SI	ITB/IUS	BEL	SUB	|	¬	?	"								EQ

RSLS196-0

Figure D-1. EBCDIC Character Set

ASCII Character Set

Figure D-2 shows the ASCII character set.

		Main Storage Bit Positions 0,1,2,3															
Main Storage Bit Positions		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
4,5,6,7	**Hex**	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	NUL	DLE	SP	0	@	P	`	p								
0001	1	SOH	DC1	!	1	A	Q	a	q								
0010	2	STX	DC2	"	2	B	R	b	r								
0011	3	ETX	DC3	#	3	C	S	c	s								
0100	4	EOT	DC4	$	4	D	T	d	t								
0101	5	ENQ	NAK	%	5	E	U	e	u								
0110	6	ACK	SYN	&	6	F	V	f	v								
0111	7	BEL	ETB	'	7	G	W	g	w								
1000	8	BS	CAN	(	8	H	X	h	x								
1001	9	HT	EM	)	9	I	Y	i	y								
1010	A	LF	SUB	*	:	J	Z	j	z								
1011	B	VT	ESC	+	;	K	[	k	{								
1100	C	FF	FS	,	<	L	\	l	|							
1101	D	CR	GS	-	=	M	]	m	}								
1110	E	SO	RS	.	>	N	^	n	~								
1111	F	SI	US	/	?	O	_	o	DEL								

RSLS197-1

Figure D-2. ASCII Character Set

Appendix E. File Transfer Support

This appendix describes the application interface to the AS/400 system file transfer support (FTS).

File Transfer Support Overview

Using FTS, a user application program can send or retrieve database file members between one AS/400 system and another AS/400 system, send database file members to System/36, and retrieve files and library members from System/36. System/36 Release 5.1 with preventive PTF – DK3700 is required to communicate with System/36. FTS does not support the sending and retrieving of database file members between an AS/400 system and a System/38.

If an AS/400 database file has more than one member, you can send or retrieve only one member at a time. If a database file does not exist, it is created and the member is added. If a database file exists and the member does not exist, the member is added. If a member exists, you can specify that the member be replaced.

FTS running on the local system communicates with FTS on the remote system to complete the request. FTS defines the **target system** as the system that receives the object, which can be the local system or the remote system. For example, if system A in Figure E-1 sends a database file member to system B using FTS, system B is the target system. If system A retrieves a database file member from system B using FTS support, system A is the target system.

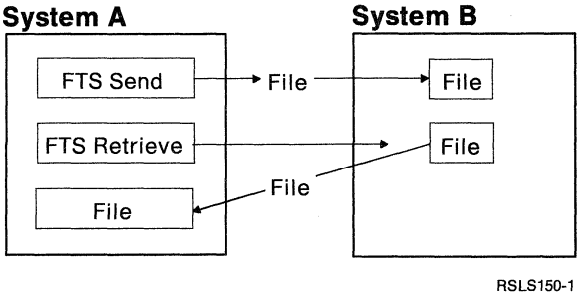


Figure E-1. Example of File Transfer Support

An application program can use FTS by calling the program QY2FTML.

Note: FTS uses the file QSYS/QY2ICFF. Do not change or delete this file. This file contains the DDS formats for the ICF file.

Either a high-level language program or a control language (CL) program can call FTS. FTS is supported by all high-level languages. Refer to the C/400, COBOL/400, RPG/400, and CL program examples in this appendix for more details.

FTS evokes a partner application on the remote system. You do not need to have a user application on the remote system to use FTS.

FTS assumes that the user ID used on the remote system is the same as the user ID for the job from which FTS was started. If you are creating a file on an AS/400 system, the sending user ID becomes the owner.

For line speeds greater than or equal to 56000 bits per second (bps), FTS does not compress data. However, for a token ring network, FTS always compresses data. Therefore, for line speeds greater than 56000 bps, FTS can transmit data between an AS/400 system and a System/36 or another AS/400 system that uses a token ring network.

File Transfer Considerations

The following sections describe in more detail the objects that you can send and retrieve using FTS.

To and From an AS/400 System

You can send physical file members to or retrieve them from an AS/400 system.

You cannot send or retrieve the following objects:

- Physical files that are part of an interactive data definition utility (IDDU) dictionary
- AS/400 program objects
- Logical files
- Device files

AS/400 System Retrieving from System/36

On an AS/400 system, you can retrieve the following objects from System/36:

- System/36 files
- System/36 library members

You cannot retrieve the following objects:

- System/36 data dictionary
- System/36 folder/document

When you retrieve a file from System/36, the file is stored on the AS/400 system as a physical file within a library. If you do not specify a member name, the member name becomes the creation date with an M added as the first character.

When you retrieve a library member from System/36, the file is stored on the AS/400 system as a physical file member in a file within a library. If you do not specify the file name, the type of member determines the file name. Source library members are stored in file QS36SRC, procedure library members are stored in file QS36PRC, subroutine library members are stored in file QS36SBR, and load library members are stored in file QS36LOD.

Note: A System/36 index file cannot be added as a member to a keyed file created on an AS/400 system.

AS/400 System Sending to System/36

You can send physical file members from an AS/400 system to System/36.

You cannot send the following objects:

- Physical files that are part of an interactive data definition utility (IDDU) dictionary
- AS/400 program objects
- Logical files
- Device files

The attributes of the physical file determine if physical file members are stored as library members or files on System/36.

If the physical file member has the attributes of a System/36 file, it is stored as a file on System/36. The file keeps its file organization (such as direct, sequential, or indexed). The file name and date field from the parameters are used to create the file on System/36.

If the physical file member being sent does not have the attributes of a System/36 file, the member is stored on System/36 as a library member. The library name, the member name, and the type field from the input parameters are used to create the library member on System/36.

Notes:

1. If a library name, file name, or member name is greater than 8 characters, System/36 uses the first 8 characters as the name.
2. The AS/400 system receives and keep source members with no records but the System/36 does not. If an empty source member is sent to a System/36, the source member is not kept.

Multiple Communication-Type Support

When using FTS, your system can set up communications with the following communication types and specific links:

- Advanced program-to-program communications (APPC)
  - Multiple sessions at the same time
  - Switched or nonswitched connections
  - Synchronous data link control (SDLC) links
  - X.25 links
  - Token ring local area network (LAN) links
  - Advanced peer-to-peer networking (APPN) capability

  When you configure a device description, one of the mode names you specify must be *NETATR.

- Binary synchronous communications equivalence link (BSCEL)
  - Single session only
  - Switched or nonswitched connections

  With BSCEL, you can configure the maximum user record length that you are going to send. This length cannot be greater than 4075. You must specify *YES for the TRNSPY parameter and *YES for the RMTBSCEL parameter when you

configure BSCEL. BSCEL supports a maximum length of 8 characters for the user ID and a maximum of 4 characters for the password. You must also specify *NONE for the BLOCK parameter (this is the default value).

- Asynchronous communications
  - Single session only
  - Switched or nonswitched connections

    Asynchronous communications performs a logical link protocol to ensure data integrity. Data is sent as 8-bit EBCDIC, not as ASCII. When you are using FTS on an asynchronous line description, your modem must be set to full duplex.

    If your system is connected to a network by a packet assembler/disassembler (PAD) and you use FTS on an X.25 packet-switched data network (PSDN), you must set the X.3 parameters and any network-specific parameters to allow data transparency. To achieve data transparency, set the X.3 parameters and any network-specific parameters to allow the following:

    - No PAD recall using a character
    - No echo
    - No selection of data forwarding characters
    - No use of XON/XOFF
    - PAD must send interrupt when a break signal from start-stop mode DTE is received
    - PAD must allow EBCDIC data
    - PAD must allow 8-bit transparency
    - Only forward on full packets or idle timer

Notes:

1. The network PAD must not perform any operations on the file transfer data stream.
2. If you use FTS with AS/400 integrated PAD, the X.3 parameter settings are ignored in order to achieve data transparency.

See the *Asynchronous Communications Programmer's Guide* for more information about the X.3 parameters.

You must create configurations for each communication type and vary on the configuration on both the local and remote systems. For further information on communications configurations, refer to the *Communications User's Guide*. FTS establishes a link to the varied on configuration based on the remote location name supplied in the input parameters.

FTS has a maximum record length of 4075. If you use BSCEL and configure your maximum user record length, the record must be at least 512 bytes long. You can, however, send files with less than 512-byte records.

File Transfer Parameters

This section describes the parameters passed to the file transfer program QY2FTML. FTS parameters are all positional. You must, therefore, reserve space in your program for all parameters. If you do not use a parameter, fill its space with blanks.

Refer to the *CL Reference* for general rules about naming libraries, database files, and database file members. Refer to the *System/36 System Reference* for System/36 naming conventions.

To and From an AS/400 System

Table E-1 describes the parameters for sending and retrieving files between one AS/400 system and another AS/400 system.

Table E-1 (Page 1 of 3). Transferring Files To and From an AS/400 System

Parameter	Value	Description
OPTION	Character	File transfer option to perform. Length: 1 character Type: Input, required Valid values are as follows: • S—Send • R—Retrieve
FROMLIB	Library name	Name of the library that contains the database file. Length: 10 characters Type: Input, required Valid values: Library name
FROMFILE	File name	Name of the database file that contains the member. Length: 10 characters Type: Input, required Valid values: Database file name
FROMMBR	Member name	Name of the member. Length: 10 characters Type: Input, required Valid values: Member name
TYPE	Blanks	Not needed for an AS/400 system to an AS/400 system. Length: 6 characters Type: Not applicable Valid values: Blanks

Table E-1 (Page 2 of 3). Transferring Files To and From an AS/400 System

Parameter	Value	Description
TOLIB	Target library name	Name of the receiving library. Length: 10 characters Type: Input, not required Valid values: Library name Default: FROMLIB
TOFILE	Target file name	Name of the receiving database file. Length: 10 characters Type: Input, not required Valid values: File name Default: FROMFILE
TOMBR	Target file member name	Name of the receiving member. Length: 10 characters Type: Input, not required Valid values: Member name Default: FROMMBR When you use TOMBR, consider the following: • If you are replacing a database member (REPLACE = Y), this is the name of the member to replace at the target system. • If you are adding a new database member, this is the name to assign.
TODATE	Blanks	Not needed for an AS/400 system to an AS/400 system. Length: 6 characters Type: Not applicable Valid values: Blanks
REPLACE	Character	This field tells whether you want to replace the member on the target system. Length: 1 character Type: Input, not required Valid values: • Y—Replace an existing member on the target system. • N—Do not replace an existing member. Default: N If you specify REPLACE = Y for a member, you cannot use the database file containing that member for any other operation during the replace operation.

Table E-1 (Page 3 of 3). Transferring Files To and From an AS/400 System

Parameter	Value	Description
RMTLOCNAME	Location Name	Name of the remote location with which you are communicating. Length: 8 characters Type: Input, required Valid values: Remote location name in a varied-on device description.
PASSWORD	PASSWORD	Password for signing on the remote system. Length: 10 characters Type: Input. This field is required only if the remote system has password security active. Valid values: Password
RTNCODE	Character	This field contains the return code. FTS returns this value to the application program to indicate the result of the transfer. Length: 1 character Type: Output Valid values: • 0—Normal completion. • 1—An error was detected at the local system. • 2—An error was detected at the remote system. For return codes 1 and 2, the specific error is sent to the job log of both systems, and the message-id is returned to the user in the *message-id* field. (See "File Transfer Support Messages" on page E-38 for more information.)
MESSAGE-ID	Character	This field contains the message-id for the specific error if the value returned in the RTNCODE field is 1 or 2 (indicating an error). Length: 8 characters Type: Output Valid values: Any message-id listed in the message section (See "File Transfer Support Messages" on page E-38 for more information.)

AS/400 System Sending to System/36

Table E-2 describes the parameters for sending from an AS/400 system to System/36.

Note: The TYPE parameter determines how data is stored on the System/36, therefore, it is important to specify the value of the TYPE parameter correctly. If this parameter is not specified correctly, results may occur that cannot be predicted.

Table E-2 (Page 1 of 4). AS/400 System Sending to System/36

Parameter	Value	Description
OPTION	Character	File transfer option to perform. Length: 1 character Type: Input, required Valid values are as follows: • S—Send • R—Retrieve. Refer to "AS/400 System Retrieving a File from System/36" on page E-11. This section describes only send.
FROMLIB	Library name	Name of the library that contains the database file. Length: 10 characters Type: Input, required Valid values: Valid library name
FROMFILE	File name	Name of the database file that contains the member. Length: 10 characters Type: Input, required Valid values: Valid database file name
FROMMBR	File member name	Name of the member. Length: 10 characters Type: Input, required Valid values: Valid member name
TYPE	File member type	This field tells System/36 how to store this member. Length: 6 characters Type: Input, required Valid values: SOURCE, LOAD, PROC (procedure), SUBR (subroutine), valid system date, or blanks. SOURCE, LOAD, PROC, or SUBR must be used if the member is to be stored as a library member. A date or blanks can be used if the member is to be stored as a file.

Table E-2 (Page 2 of 4). AS/400 System Sending to System/36

Parameter	Value	Description
TOLIB	Target library name	Name of the receiving library to which the member is sent. If the member's attributes indicate that this is a System/36 file, this parameter must be left blank and the TOFILE parameter must be specified. Length: 10 characters (see note at the end of this figure) Type: Input, not required Valid values: System/36 library name, System/36 file name, and blanks Default: FROMLIB
TOFILE	Target file name	Name of the file. If the attributes of the member being sent indicate that this is a System/36 file, this parameter is used as the file name on System/36. Length: 10 characters (see note at the end of this figure) Type: Input, not required Valid values: System/36 file name and blanks Default: FROMFILE
TOMBR	Target member name	Name of the library member. If the attributes of the member being sent indicate that this is a System/36 file, this parameter is not used. Length: 10 characters (see note at the end of this figure) Type: Input, not required Valid values: System/36 library member name Default: FROMMBR When using TOMBR, consider the following: • If you are replacing a library member (REPLACE=Y), this is the name of the library member to replace at the target system. • If you are adding a new library member, this is the name to assign.
TODATE	Numeric	This field is used to change the date of a file sent to a System/36. Make sure that the system date format on the target system is the same as the format on your system. This field is not used if the member being sent will be used as a library member. Length: 6 characters Type: Input, not required Valid values: Numeric date

Table E-2 (Page 3 of 4). AS/400 System Sending to System/36

Parameter	Value	Description
REPLACE	Character	This field tells whether you want to replace the file or library member on the target system. Length: 1 character Type: Input, not required Valid values: • Y—Replace an existing file or library member on the target system. • N—Do not replace an existing file or library member. Default: N If you specify REPLACE = Y for a library member, you cannot use the library containing that member for any other operation during the replace operation. If you specify REPLACE = Y for a file, the file cannot be used for any other operation during the replace operation.
RMTLOCNAME	Location Name	Name of the remote location with which you are communicating. Length: 8 characters Type: Input, required Valid values: Remote location name in a varied-on device description.
PASSWORD	Password	Password for signing on the remote system. Length: 10 characters. The largest password System/36 accepts is 4 characters. Type: Input. This field is required only if the remote system has password security active. Valid values: Password
RTNCODE	Character	This field contains the return code. FTS returns this value to the application program to indicate the result of the transfer. Length: 1 character Type: Output Valid values: • 0—Normal completion. • 1—An error was detected at the local system. • 2—An error was detected at the remote system. For return codes 1 and 2, the specific error is logged to the history file of the System/36 and to the job log file of the AS/400 system. The message-id is returned to the user in the *message-id* field. (See "File Transfer Support Messages" on page E-38 for more information.)

Table E-2 (Page 4 of 4). AS/400 System Sending to System/36

Parameter	Value	Description
MESSAGE-ID	Character	This field contains the message-id for the specific error if the value returned in the RTNCODE field is 1 or 2 (indicating an error). Length: 8 characters Type: Output Valid values: Any message-id listed in the message section. (See "File Transfer Support Messages" on page E-38 for more information.)

Note: If a library name, file name, or member name is greater than 8 characters, System/36 uses the first 8 characters as the name.

AS/400 System Retrieving a File from System/36

Table E-3 describes the parameters for retrieving a file from System/36.

Table E-3 (Page 1 of 4). AS/400 System Retrieving a File from System/36

Parameter	Value	Description
OPTION	Character	File transfer option to perform. Length: 1 character Type: Input, required Valid values are as follows: • S—Send. Refer to "AS/400 System Sending to System/36" on page E-7. This section describes only retrieve. • R—Retrieve a file from the remote System/36.
FROMLIB	Blanks	Not needed in retrieving a file from System/36. Length: 10 characters Type: Not applicable Valid values: Blanks
FROMFILE	File name	Name of the System/36 file. Length: 10 characters (see note at the end of this figure) Type: Input, required Valid values: System/36 file name
FROMMBR	Blanks	Not needed in retrieving a file from System/36. Length: 10 characters Type: Not applicable Valid values: Blanks

Table E-3 (Page 2 of 4). AS/400 System Retrieving a File from System/36

Parameter	Value	Description
TYPE	File type	This field contains the date of the file to retrieve. Make sure that the system date format on the target system is the same as the format on your system. Length: 6 characters Type: Input, not required Valid values: Numeric date Default: If no date is given, the most recent file is retrieved.
TOLIB	Target library name	The name of the receiving library. Length: 10 characters Type: Input, not required Valid values: Library name Default: The System/36 environment default library name.
TOFILE	Target file name	Name of the receiving database file. Length: 10 characters Type: Input, not required Valid values: File name Default: FROMFILE
TOMBR	Target file member name	Name of the receiving member. Length: 10 characters Type: Input, not required Valid values: Member name Default: If the name is not given, the member name is the date given in TODATE or the creation date with an *M* added as the first character. When using TOMBR, consider the following: • If you are replacing a member (REPLACE=Y), TOMBR is the name of the member to replace at the target system. • If you are adding a new member, TOMBR is the name to assign.
TODATE	Numeric	This field gives a different date to a file received from a System/36. Make sure that the system date format on the target system is the same as the format on your system. Length: 6 characters Type: Input, not required Valid values: Numeric date

Table E-3 (Page 3 of 4). AS/400 System Retrieving a File from System/36

Parameter	Value	Description
REPLACE	Character	This field tells whether you want to replace the member. Length: 1 character Type: Input, not required Valid values: • Y—Replace an existing member. • N—Do not replace an existing member. Default: N If you specify REPLACE=Y for a member, you cannot use the database file containing that member for any other operation during the replace operation.
RMTLOCNAME	Location Name	Name of the remote location with which you are communicating. Length: 8 characters Type: Input, required Valid values: Remote location name in a varied-on device description.
PASSWORD	Password	Password for signing on the remote system. Length: 10 characters. The largest password System/36 accepts is 4 characters. Type: Input. This field is required only if the remote system has password security active. Valid values: Password
RTNCODE	Character	This field contains the return code. FTS returns this value to the application program to indicate the result of the transfer. Length: 1 character Type: Output Valid values: • 0—Normal completion. • 1—An error was detected at the local system. • 2—An error was detected at the remote system. For return codes 1 and 2, the specific error is logged to the history file of the System/36 and to the job log file of the AS/400 system. The message-id is returned to the user in the *message-id* field. (See "File Transfer Support Messages" on page E-38 for more information.)

Table E-3 (Page 4 of 4). AS/400 System Retrieving a File from System/36

Parameter	Value	Description
MESSAGE-ID	Character	This field contains the message-id for the specific error if the value returned in the RTNCODE field is 1 or 2 (indicating an error). Length: 8 characters Type: Output Valid values: Any message-id listed in the message section. (See "File Transfer Support Messages" on page E-38 for more information.)

Note: If a library name, file name, or member name is greater than 8 characters, System/36 uses the first 8 characters as the name.

Retrieving a Library Member from System/36

Table E-4 describes the parameters for retrieving a library member from System/36.

Table E-4 (Page 1 of 4). Retrieving a Library Member from System/36

Parameter	Value	Description
OPTION	Character	File transfer option to perform. Length: 1 character Type: Input, required Valid values are as follows: • S—Send. Refer to "AS/400 System Sending to System/36" on page E-7. This section describes only retrieve. • R—Retrieve a library member.
FROMLIB	Library name	Name of the library in which library member resides. Length: 10 characters (see note) Type: Input, required Valid values: System/36 library name
FROMFILE	Blanks	Not needed in retrieving a library member from System/36. Length: 10 characters Type: Not applicable Valid values: Blanks
FROMMBR	Library member name	Name of the library member. Length: 10 characters (see note) Type: Input, required Valid values: System/36 member name

Table E-4 (Page 2 of 4). Retrieving a Library Member from System/36

Parameter	Value	Description
TYPE	Library member type	This field tells System/36 the type of member to retrieve. Length: 6 characters Type: Input, required Valid values: SOURCE, LOAD, PROC (procedure), SUBR (subroutine)
TOLIB	Library name	Name of the receiving library to which members are sent. Length: 10 characters Type: Input, not required Valid values: Library name Default: FROMLIB
TOFILE	Target file name	Name of the file. Length: 10 characters Type: Input, not required Valid values: Database file name Default: If no value is given, the default name is determined by the type of the member, as follows: • Source—QS36SRC • Load—QS36LOD • Procedure—QS36PRC • Subroutine—QS36SBR
TOMBR	Target member name	Name of the member at the target system. Length: 10 characters Type: Input, not required Valid values: Member name Default: FROMMBR When you use TOMBR, consider the following: • If you are replacing a member (REPLACE = Y), TOMBR is the name of the member to replace at the target system. • If you are adding a new member, TOMBR is the name to assign.
TODATE	Blanks	Not needed in retrieving a library member from System/36. Length: 6 characters Type: Input, not applicable Valid values: Blanks

Table E-4 (Page 3 of 4). Retrieving a Library Member from System/36

Parameter	Value	Description
REPLACE	Character	This field tells whether you want to replace the member on the target system. Length: 1 character Type: Input, not required Valid values: • Y—Replace an existing member. • N—Do not replace an existing member. Default: N If you specify REPLACE = Y for a member, you cannot use the file containing that member for any other operation during the replace operation.
RMTLOCNAME	Location Name	Name of the remote location with which you are communicating. Length: 8 characters Type: Input, required Valid values: Remote location name in a varied-on device description.
PASSWORD	PASSWORD	Password for signing on the remote system. Length: 10 characters. The largest password System/36 accepts is 4 characters. Type: Input. This field is required only if the remote system has password security active. Valid values: Password
RTNCODE	Character	This field contains the return code. FTS returns this value to the application program to indicate the result of the transfer. Length: 1 character Type: Output Valid values: • 0—Normal completion. • 1—An error was detected at the local system. • 2—An error was detected at the remote system. For return codes 1 and 2, the specific error is logged to the history log of the System/36 and to the job log file of the AS/400 system. The message-id is returned to the user in the *message-id* field. (See "File Transfer Support Messages" on page E-38 for more information.)

Table E-4 (Page 4 of 4). Retrieving a Library Member from System/36

Parameter	Value	Description
MESSAGE-ID	Character	This field contains the message-id for the specific error if the value returned in the RTNCODE field is 1 or 2 (indicating an error). Length: 8 characters Type: Output Valid values: Any message-id listed in the message section. (See "File Transfer Support Messages" on page E-38 for more information.)

Note: If a library name, file name, or member name is greater than 8 characters, System/36 uses the first 8 characters as the name.

Calling File Transfer Support for C/400

Figure E-2 on page E-19 is an example of a C/400 program that provides a data link between one AS/400 system and another AS/400 system. This program reads the file in which the parameters are stored, calls the file transfer support program (QY2FTML), and prints a listing of the parameters, return code, and message number.

The parameters passed to the file transfer program are described in "File Transfer Parameters" on page E-5.

```
 5728CX1 R00 M02 880101 IBM YY/XXX C/400      CEXAMPLES/QCSRC/CDRIVER                  05/19/89 10:58:03            Page   1
 Command options:
    Program name. . . . . . . . . : CEXAMPLES
    Source file name. . . . . . . : CEXAMPLES
    Compiler options. . . . . . . : *NOAGGR     *NODEBUG    *NOEXPMAC   *GEN        *NOALWBIND  *NOLIST     *NOPPONLY   *NOSECLVL
                                  : *NOSHOWINC  *SOURCE     *NOXREF
    Generation options. . . . . . : *NOLIST     *NOXREF     *GEN        *NOATR      *NODUMP     *NOSUBSTR   *SUBSCR     *OPTIMIZE
                                  : *NOALWBND
    User profile. . . . . . . . . : *USER
    Language level. . . . . . . . : *EXTENDED
Source margins. . . . . . . . : 1 - 32767
CCX0141
    Authority . . . . . . . . . . : *CHANGE
    Text description. . . . . . . : File transfer program
Printer file. . . . . . . . . : *LIBL
    Lines per page. . . . . . . . : 66
 Actual Program Source:
    Member. . . . . . . . . . . . : CDRIVER
    File. . . . . . . . . . . . . : QCSRC
    Library . . . . . . . . . . . : CEXAMPLES
    Last change . . . . . . . . . : 89/05/19 10:56:41
    Description . . . . . . . . . : File transfer program
 5728CX1 R00 M02 880101 IBM YY/XXX C/400      CEXAMPLES/QCSRC/CDRIVER                  05/19/89 10:58:03            Page   2
LINE  STMT                              * * * * *   S O U R C E * * * * *                                  SEQNO INCLUDE
          *---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9------*
    1       |#pragma langlvl(_EXTENDED)                /* IBM extensions */                                |     1
    2       |#pragma linkage(QY2FTML,OS)               /* Define program to be called */                   |     2
    3       |                                                                                              |     3
    4       |#define END 1                             /* Signals program end */                           |     4
    5       |#define NOEND 0                                                                              |     5
    6       |#define ERROR 1                           /* Signals an error during I/O */                   |     6
    7       |#define NOERROR 0                                                                            |     7
    8       |#include <stdio.h>                        /* Standard I/O header */                           |     8
    9       |#include <stdlib.h>                       /* General utilities */                             |     9
   10       |#include <stddef.h>                       /* Standard definitions */                          |    10
   11       |#include <string.h>                       /* String handling utilities */                     |    11
   12       |#include <xxasio.h>                       /* Indicator area structure */                      |    12
   13       |                                                                                              |    13
   14       |/*--------------------------------------------------------------------------*/                |    14
   15       |/* Define header structures to be written to the print file.               */                |    15
   16       |/*--------------------------------------------------------------------------*/                |    16
   17       |                                                                                              |    17
   18       |struct {                                                                                      |    18
   19       |    char filler??(124??);                                                                     |    19
   20       |} header_line_1 = {                                                                           |    20
   21       |             "     "                                                                          |    21
   22       |             "FRMLIB     "                                                                   |    22
   23       |             "FRMFIL     "                                                                   |    23
   24       |             "FRMMBR     "                                                                   |    24
   25       |             "TYPE     "                                                                     |    25
   26       |             "TOLIB      "                                                                   |    26
   27       |             "TOFIL      "                                                                   |    27
   28       |             "TOMBR      "                                                                   |    28
   29       |             "TODATE   "                                                                      |    29
   30       |             "OPTION "                                                                        |    30
   31       |             "REPL "                                                                          |    31
   32       |             "RMTLOC   "                                                                      |    32
   33       |             "RCODE "                                                                         |    33
   34       |             "MSGNUM  "                                                                       |    34
   35       |};                                                                                            |    35
   36       |                                                                                              |    36
   37       |struct {                                                                                      |    37
   38       |    char filler??(124??);                                                                     |    38
   39       |} header_line_2 = {                                                                           |    39
   40       |             "     "                                                                          |    40
   41       |             "__________ "                                                                    |    41
   42       |             "__________ "                                                                    |    42
   43       |             "_________ "                                                                     |    43
```

Figure E-2 (Part 1 of 7). C/400 Coding for File Transfer Support

```
   44       |             "_______ "                                                                  |     44
   45       |             "________ "                                                                 |     45
   46       |             "_________ "                                                                |     46
   47       |             "_________ "                                                                |     47
   48       |             "_________"                                                                 |     48
   49       |             "_______ "                                                                  |     49
   50       |             "______ "                                                                   |     50
   51       |             "____ "                                                                     |     51
   52       |             "_______ "                                                                  |     52
   53       |             "_____ "                                                                    |     53
   54       |             "_______ "                                                                  |     54
   55       |};                                                                                       |     55
   56       |                                                                                         |     56
   57       |/*-------------------------------------------------------------------------*/           |     57
   58       |/* Define data file structure that contains the values to be assigned   */           |     58
   59       |/* to parameters passed on call to program QY2FTML.                       */           |     59
   60       |/*-------------------------------------------------------------------------*/           |     60
   61       |                                                                                         |     61
 5728CX1 R00 M02 880101 IBM YY/XXX C/400      CEXAMPLES/QCSRC/CDRIVER                  05/19/89 10:58:03          Page   3
LINE  STMT                              * * * * *   S O U R C E * * * * *                        SEQNO INCLUDE
          *---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9------*
   62       |    char rec_num??(3??);                                                                 |     62
   63       |    char option;                                                                         |     63
   64       |    char repl;                                                                           |     64
   65       |    char filler1??(4??);                                                                 |     65
   66       |    char frmlib??(10??);                                                                 |     66
   67       |    char frmfil??(10??);                                                                 |     67
   68       |    char frmmbr??(10??);                                                                 |     68
   69       |    char typ??(6??);                                                                     |     69
   70       |    char filler2??(4??);                                                                 |     70
   71       |    char tolib??(10??);                                                                  |     71
   72       |    char tofil??(10??);                                                                  |     72
   73       |    char tombr??(10??);                                                                  |     73
   74       |    char todate??(6??);                                                                  |     74
   75       |    char filler3??(4??);                                                                 |     75
   76       |    char rmtloc??(8??);                                                                  |     76
   77       |    char passwd??(10??);                                                                 |     77
   78       |    char rcode;                                                                          |     78
   79       |    char msgnum??(8??);                                                                  |     79
   80       |} call_rec;                                                                              |     80
   81       |                                                                                         |     81
   82       |/*-------------------------------------------------------------------------*/           |     82
   83       |/* Define the structure types of the parameters on call to QY2FTML. The */           |     83
   84       |/* type definition is needed for prototyping.                             */           |     84
   85       |/*-------------------------------------------------------------------------*/           |     85
   86       |                                                                                         |     86
   87       |typedef struct {                                                                         |     87
   88       |    char option;                                                                         |     88
   89       |} option;                                                                                |     89
   90       |                                                                                         |     90
   91       |typedef struct {                                                                         |     91
   92       |    char frmlib??(10??);                                                                 |     92
   93       |} lib;                                                                                   |     93
   94       |                                                                                         |     94
   95       |typedef struct {                                                                         |     95
   96       |    char frmfil??(10??);                                                                 |     96
   97       |} file;                                                                                  |     97
   98       |                                                                                         |     98
   99       |typedef struct {                                                                         |     99
  100       |    char frmmbr??(10??);                                                                 |    100
  101       |} file_member;                                                                           |    101
  102       |                                                                                         |    102
  103       |typedef struct {                                                                         |    103
  104       |    char typ??(6??);                                                                     |    104
  105       |} type;                                                                                  |    105
  106       |                                                                                         |    106
  107       |typedef struct {                                                                         |    107
  108       |    char tolib??(10??);                                                                  |    108
```

Figure E-2 (Part 2 of 7). C/400 Coding for File Transfer Support

```
  109       |} tgt_lib;                                                                                                  |  109
  110       |                                                                                                            |  110
  111       |typedef struct {                                                                                            |  111
  112       |    char tofil??(10??);                                                                                     |  112
  113       |} tgt_file;                                                                                                 |  113
  114       |                                                                                                            |  114
  115       |typedef struct {                                                                                            |  115
  116       |    char tombr??(10??);                                                                                     |  116
  117       |} tgt_member;                                                                                               |  117
  118       |                                                                                                            |  118
  119       |typedef struct {                                                                                            |  119
  120       |    char todate??(6??);                                                                                     |  120
  121       |} tgt_file_date;                                                                                            |  121
  122       |                                                                                                            |  122
 5728CX1 R00 M02 880101 IBM YY/XXX C/400      CEXAMPLES/QCSRC/CDRIVER                    05/19/89 10:58:03              Page   4
LINE  STMT                                * * * * *   S O U R C E * * * * *                                  SEQNO INCLUDE
           *---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9------*
  123       |typedef struct {                                                                                            |  123
  124       |    char repl;                                                                                              |  124
  125       |} replace_member;                                                                                           |  125
  126       |                                                                                                            |  126
  127       |typedef struct {                                                                                            |  127
  128       |    char rmtloc??(8??);                                                                                     |  128
  129       |} rmt_loc;                                                                                                  |  129
  130       |                                                                                                            |  130
  131       |typedef struct {                                                                                            |  131
  132       |    char passwd??(10??);                                                                                    |  132
  133       |} pword;                                                                                                    |  133
  134       |                                                                                                            |  134
  135       |typedef struct {                                                                                            |  135
  136       |    char rcode;                                                                                             |  136
  137       |} ret_code;                                                                                                 |  137
  138       |                                                                                                            |  138
  139       |typedef struct {                                                                                            |  139
  140       |    char msgnum??(8??);                                                                                     |  140
  141       |} msg_num;                                                                                                  |  141
  142       |                                                                                                            |  142
  143       |/*------------------------------------------------------------------------*/                                |  143
  144       |/* Define structure for data to be written to the print file.           */                                |  144
  145       |/*------------------------------------------------------------------------*/                                |  145
  146       |                                                                                                            |  146
  147       |struct {                                                                                                    |  147
  148       |    char prec_num??(3??);                                                                                   |  148
  149       |    char filler1??(2??);                                                                                    |  149
  150       |    char pfrmlib??(10??);                                                                                   |  150
  151       |    char filler2;                                                                                           |  151
  152       |    char pfrmfil??(10??);                                                                                   |  152
  153       |    char filler3;                                                                                           |  153
  154       |    char pfrmmbr??(10??);                                                                                   |  154
  155       |    char filler4;                                                                                           |  155
  156       |    char ptyp??(6??);                                                                                       |  156
  157       |    char filler5??(3??);                                                                                    |  157
  158       |    char ptolib??(10??);                                                                                    |  158
  159       |    char filler6;                                                                                           |  159
  160       |    char ptofil??(10??);                                                                                    |  160
  161       |    char filler7;                                                                                           |  161
  162       |    char ptombr??(10??);                                                                                    |  162
  163       |    char filler8;                                                                                           |  163
  164       |    char ptodate??(6??);                                                                                    |  164
  165       |    char filler9??(5??);                                                                                    |  165
  166       |    char poption;                                                                                           |  166
  167       |    char filler10??(5??);                                                                                   |  167
  168       |    char prepl;                                                                                             |  168
  169       |    char filler11??(3??);                                                                                   |  169
  170       |    char prmtloc??(8??);                                                                                    |  170
  171       |    char filler12??(3??);                                                                                   |  171
  172       |    char prcode;                                                                                            |  172
  173       |    char filler13??(3??);                                                                                   |  173
```

Figure E-2 (Part 3 of 7). C/400 Coding for File Transfer Support

```
   174       |    char pmsgnum??(8??);                                                                                |   174
   175       |} print_rec;                                                                                            |   175
   176       |                                                                                                        |   176
   177       |/*--------------------------------------------------------------------------*/                          |   177
   178       |/* Declare structures to use as parameters on call to QY2FTML using the */                           |   178
   179       |/* type definitions already defined.                                       */                          |   179
   180       |/*--------------------------------------------------------------------------*/                          |   180
   181       |                                                                                                        |   181
   182       |option         option_parm;                                                                             |   182
   183       |lib            lib_parm;                                                                                |   183
 5728CX1 R00 M02 880101 IBM YY/XXX C/400      CEXAMPLES/QCSRC/CDRIVER                    05/19/89 10:58:03              Page   5
LINE  STMT                                * * * * *   S O U R C E * * * * *                                  SEQNO INCLUDE
          *---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9------*
   184       |file           file_parm;                                                                               |   184
   185       |file_member    file_member_parm;                                                                        |   185
   186       |type           type_parm;                                                                               |   186
   187       |tgt_lib        tgt_lib_parm;                                                                            |   187
   188       |tgt_file       tgt_file_parm;                                                                           |   188
   189       |tgt_member     tgt_member_parm;                                                                         |   189
   190       |tgt_file_date  tgt_file_date_parm;                                                                      |   190
   191       |replace_member replace_member_parm;                                                                     |   191
   192       |rmt_loc        rmt_loc_parm;                                                                            |   192
   193       |pword          pword_parm;                                                                              |   193
   194       |ret_code       ret_code_parm;                                                                           |   194
   195       |msg_num        msg_num_parm;                                                                            |   195
   196       |                                                                                                        |   196
   197       |char op_name??(5??);                        /* Current operation */                                 |   197
   198       |                                                                                                        |   198
   199       |extern void QY2FTML(option *, lib *, file *, file_member *, type *,                                     |   199
   200       |                    tgt_lib *, tgt_file *, tgt_member *, tgt_file_date *,                               |   200
   201       |                    replace_member *, rmt_loc *, pword *, ret_code *,                                   |   201
   202       |                    msg_num *);                                                                         |   202
   203       |int print_header(FILE *);                                                                               |   203
   204       |int file_trans(FILE *);                                                                                 |   204
   205       |void init_parms(void);                                                                                  |   205
   206       |int print_parms(FILE *);                                                                                |   206
   207       |int pos_ret_code_printf(void);                                                                          |   207
   208       |void print_file_error(void);                                                                            |   208
   209       |void close_files(FILE *, FILE *);                                                                       |   209
   210       |                                                                                                        |   210
   211       |main()                                                                                                  |   211
   212       |{                                                                                                       |   212
   213       |    FILE  *dtafptr;                         /* Pointer to the database file */                      |   213
   214       |    FILE  *prtfptr;                         /* Pointer to the printer file */                       |   214
   215       |                                                                                                        |   215
   216     1 |    if ((dtafptr = fopen("FTTEST", "rb type=record")) == NULL) {                                        |   216
   217     2 |        printf("\nUNEXPECTED ERROR WHILE OPENING DATA FILE.\n");                                       |   217
   218     3 |        exit(ERROR);                                                                                    |   218
   219       |    }                                                                                                   |   219
   220     4 |    if ((prtfptr = fopen("QSYSPRT", "wb type=record")) == NULL) {                                       |   220
   221     5 |        fclose(dtafptr);                                                                                |   221
   222     6 |        printf("\nUNEXPECTED ERROR WHILE OPENING PRINT FILE.\n");                                      |   222
   223     7 |        exit(ERROR);                                                                                    |   223
   224       |    }                                                                                                   |   224
   225     8 |    if (print_header(prtfptr) == NOERROR) {                                                             |   225
   226     9 |        while (1) {                                                                                     |   226
   227    10 |            if (file_trans(dtafptr) == END)                                                             |   227
   228    11 |                break;                                                                                  |   228
   229       |            else                                                                                        |   229
   230    12 |                if (print_parms(prtfptr) == ERROR) {                                                    |   230
   231    13 |                    print_file_error();                                                                 |   231
   232    14 |                    close_files(dtafptr, prtfptr);                                                      |   232
   233    15 |                    exit(ERROR);                                                                        |   233
   234       |                }                                                                                       |   234
   235       |        }                                                                                               |   235
   236       |    }                                                                                                   |   236
   237       |    else {                                                                                              |   237
   238    16 |       print_file_error();                                                                              |   238
```

Figure E-2 (Part 4 of 7). C/400 Coding for File Transfer Support

```
  239    17 |       close_files(dtafptr, prtfptr);                                                                  |  239
  240    18 |       exit(ERROR);                                                                                    |  240
  241       |   }                                                                                                   |  241
  242    19 |   close_files(dtafptr, prtfptr);                                                                      |  242
  243    20 |   exit(NOERROR);                                                                                      |  243
  244       |}                                                                                                      |  244
 5728CX1 R00 M02 880101 IBM YY/XXX C/400      CEXAMPLES/QCSRC/CDRIVER                    05/19/89 10:58:03              Page   6
LINE  STMT                                * * * * *   S O U R C E * * * * *                                  SEQNO INCLUDE
           *---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9------*
  245       |                                                                                                     |  245
  246       |                                                                                                     |  246
  247       |/*-------------------------------------------------------------------*/                               |  247
  248       |/* The routine prints a header to the print file.                    */                               |  248
  249       |/*-------------------------------------------------------------------*/                               |  249
  250       |                                                                                                     |  250
  251       |print_header(FILE *prtfptr)                                                                          |  251
  252       |{                                                                                                    |  252
  253       |    strcpy(op_name, "WRITE");                                                                        |  253
  254    2  |    fwrite(&header_line_1, sizeof(header_line_1), 1, prtfptr);                                       |  254
  255    3  |    if (pos_ret_code_printf() == NOERROR) {                                                          |  255
  256    4  |        fwrite(&header_line_2, sizeof(header_line_2), 1, prtfptr);                                   |  256
  257    5  |        return(pos_ret_code_printf());                                                               |  257
  258       |    }                                                                                                |  258
  259       |    else                                                                                             |  259
  260    6  |        return(ERROR);                                                                               |  260
  261       |}                                                                                                    |  261
  262       |                                                                                                     |  262
  263       |                                                                                                     |  263
  264       |/*-------------------------------------------------------------------*/                               |  264
  265       |/* The routine gets parameters from the data file and calls QY2FTML.  */                               |  265
  266       |/*-------------------------------------------------------------------*/                               |  266
  267       |                                                                                                     |  267
  268       |file_trans(FILE *dtafptr)                                                                            |  268
  269       |{                                                                                                    |  269
  270       |    int len;                                                                                         |  270
  271       |                                                                                                     |  271
  272       |    strcpy(op_name, "READ ");                                                                        |  272
  273    2  |    if ((len = fread(&call_rec, sizeof(call_rec), 1, dtafptr)) == 0)                                 |  273
  274    3  |        return(END);                                                                                 |  274
  275       |    else {                                                                                           |  275
  276    4  |        init_parms();                                                                                |  276
  277    5  |        QY2FTML(&option_parm, &lib_parm, &file_parm, &file_member_parm,                              |  277
  278       |                &type_parm, &tgt_lib_parm, &tgt_file_parm, &tgt_member_parm,                         |  278
  279       |                &tgt_file_date_parm, &replace_member_parm, &rmt_loc_parm,                            |  279
  280       |                &pword_parm, &ret_code_parm, &msg_num_parm);                                         |  280
  281    6  |        return(NOEND);                                                                               |  281
  282       |        }                                                                                            |  282
  283       |}                                                                                                    |  283
  284       |                                                                                                     |  284
  285       |                                                                                                     |  285
  286       |/*-------------------------------------------------------------------*/                               |  286
  287       |/* This routine initializes the parameters for the call to QY2FTML.  */                               |  287
  288       |/* Parameters passed to external programs in C/400 must be a structure */                              |  288
  289       |/* type, so the fields of the structure call_rec may not be sent indiv- */                            |  289
  290       |/* idually to QY2FTML.                                               */                               |  290
  291       |/*-------------------------------------------------------------------*/                               |  291
  292       |                                                                                                     |  292
  293       |void init_parms()                                                                                    |  293
  294       |{                                                                                                    |  294
  295    1  |    option_parm.option = call_rec.option;                                                            |  295
  296       |    strcpy(lib_parm.frmlib, call_rec.frmlib);                                                        |  296
  297       |    strcpy(file_parm.frmfil, call_rec.frmfil);                                                       |  297
  298       |    strcpy(file_member_parm.frmmbr, call_rec.frmmbr);                                                |  298
  299       |    strcpy(type_parm.typ, call_rec.typ);                                                             |  299
  300       |    strcpy(tgt_lib_parm.tolib, call_rec.tolib);                                                      |  300
  301       |    strcpy(tgt_file_parm.tofil, call_rec.tofil);                                                     |  301
  302       |    strcpy(tgt_member_parm.tombr, call_rec.tombr);                                                   |  302
  303       |    strcpy(tgt_file_date_parm.todate, call_rec.todate);                                              |  303
```

Figure E-2 (Part 5 of 7). C/400 Coding for File Transfer Support

```
   304    10 |    replace_member_parm.repl = call_rec.repl;                                                    |    304
   305       |    strcpy(rmt_loc_parm.rmtloc, call_rec.rmtloc);                                                |    305
 5728CX1 R00 M02 880101 IBM YY/XXX C/400      CEXAMPLES/QCSRC/CDRIVER                    05/19/89 10:58:03          Page   7
LINE  STMT                              * * * * *   S O U R C E * * * * *                                  SEQNO INCLUDE
          *---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9------*
   306       |    strcpy(pword_parm.passwd, call_rec.passwd);                                                  |    306
   307    13 |    ret_code_parm.rcode = ' ';                                                                   |    307
   308       |    strcpy(msg_num_parm.msgnum, "        ");                                                     |    308
   309       |}                                                                                                |    309
   310       |                                                                                                 |    310
   311       |                                                                                                 |    311
   312       |/*-----------------------------------------------------------------------*/                       |    312
   313       |/* This routine prints the parameters passed to QY2FTML after the call. */                       |    313
   314       |/*-----------------------------------------------------------------------*/                       |    314
   315       |                                                                                                 |    315
   316       |print_parms(FILE *prtfptr)                                                                       |    316
   317       |{                                                                                                |    317
   318       |    strncpy(print_rec.prec_num, call_rec.rec_num, 3);                                            |    318
   319     2 |    print_rec.poption = call_rec.option;                                                         |    319
   320     3 |    print_rec.prepl = call_rec.repl;                                                             |    320
   321       |    strncpy(print_rec.pfrmlib, lib_parm.frmlib, 10);                                             |    321
   322       |    strncpy(print_rec.pfrmfil, file_parm.frmfil, 10);                                            |    322
   323       |    strncpy(print_rec.pfrmmbr, file_member_parm.frmmbr, 10);                                     |    323
   324       |    strncpy(print_rec.ptyp, type_parm.typ, 6);                                                   |    324
   325       |    strncpy(print_rec.ptolib, tgt_lib_parm.tolib, 10);                                           |    325
   326       |    strncpy(print_rec.ptofil, tgt_file_parm.tofil, 10);                                          |    326
   327       |    strncpy(print_rec.ptombr, tgt_member_parm.tombr, 10);                                        |    327
   328       |    strncpy(print_rec.ptodate, tgt_file_date_parm.todate, 6);                                    |    328
   329       |    strncpy(print_rec.prmtloc, rmt_loc_parm.rmtloc, 8);                                          |    329
   330    13 |    print_rec.prcode = ret_code_parm.rcode;                                                      |    330
   331       |    strncpy(print_rec.pmsgnum, msg_num_parm.msgnum, 8);                                          |    331
   332       |    strncpy(print_rec.filler1, "  ", 2);                                                         |    332
   333       |    strncpy(print_rec.filler5, "   ", 3);                                                        |    333
   334       |    strncpy(print_rec.filler9, "    ", 5);                                                       |    334
   335       |    strncpy(print_rec.filler10, "    ", 5);                                                      |    335
   336       |    strncpy(print_rec.filler11, "   ", 3);                                                       |    336
   337       |    strncpy(print_rec.filler12, "   ", 3);                                                       |    337
   338       |    strncpy(print_rec.filler13, "   ", 3);                                                       |    338
   339    22 |    print_rec.filler2 = print_rec.filler3 = print_rec.filler4                                    |    339
   340       |    = print_rec.filler6 = print_rec.filler7 = print_rec.filler8 = ' ';                           |    340
   341       |    strcpy(op_name, "WRITE");                                                                    |    341
   342    24 |    fwrite(&print_rec, sizeof(print_rec), 1, prtfptr);                                           |    342
   343    25 |    return(pos_ret_code_printf());                                                               |    343
   344       |}                                                                                                |    344
   345       |                                                                                                 |    345
   346       |                                                                                                 |    346
   347       |/*-----------------------------------------------------------------------*/                       |    347
   348       |/* This routine checks to see if the last operation on the print file   */                       |    348
   349       |/* was successful.                                                       */                       |    349
   350       |/*-----------------------------------------------------------------------*/                       |    350
   351       |                                                                                                 |    351
   352       |pos_ret_code_printf()                                                                            |    352
   353       |{                                                                                                |    353
   354     1 |    if (strncmp(_Maj_Min_rc.major_rc, "00", 2) == NOERROR)                                       |    354
   355     2 |        return(NOERROR);                                                                         |    355
   356       |    else                                                                                         |    356
   357     3 |        return(ERROR);                                                                           |    357
   358       |}                                                                                                |    358
   359       |                                                                                                 |    359
   360       |                                                                                                 |    360
   361       |/*-----------------------------------------------------------------------*/                       |    361
   362       |/* This routine prints an error message to the display.                 */                       |    362
   363       |/*-----------------------------------------------------------------------*/                       |    363
   364       |                                                                                                 |    364
   365       |void print_file_error()                                                                          |    365
   366       |{                                                                                                |    366
```

Figure E-2 (Part 6 of 7). C/400 Coding for File Transfer Support

```
 5728CX1 R00 M02 880101 IBM YY/XXX C/400      CEXAMPLES/QCSRC/CDRIVER                    05/19/89 10:58:03            Page   8
LINE  STMT                                  * * * * *   S O U R C E * * * * *                                  SEQNO INCLUDE
           *---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9------*
  367     1 |        printf("\nUNEXPECTED ERROR ON %s FOR PRINT FILE\n", op_name);                                |    367
  368       |}                                                                                                    |    368
  369       |/*--------------------------------------------------------------------------*/                       |    369
  370       |/* Close the data and print files.                                          */                       |    370
  371       |/*--------------------------------------------------------------------------*/                       |    371
  372       |                                                                                                     |    372
  373       |void close_files(FILE *dtafptr, FILE *prtfptr)                                                       |    373
  374       |{                                                                                                    |    374
  375     1 |    fclose(dtafptr);                                                                                 |    375
  376     2 |    fclose(prtfptr);                                                                                 |    376
  377       |}                                                                                                    |    377
  378       |                                                                                                     |    378
 INCLUDE FILES  ---  FILE#   NAME            ACTUAL FILE USED
                       1   stdio.h          QCC/H/STDIO
                       2   stddef.h         QCC/H/STDDEF
                       3   errno.h          QCC/H/ERRNO
                       4   signal.h         QCC/H/SIGNAL
                       5   ctype.h          QCC/H/CTYPE
                       6   stdarg.h         QCC/H/STDARG
                       7   errno.h          QCC/H/ERRNO
                       8   stdlib.h         QCC/H/STDLIB
                       9   stddef.h         QCC/H/STDDEF
                      10   string.h         QCC/H/STRING
                      11   xxasio.h         QCC/H/XXASIO
 * * * * *   E N D   O F   S O U R C E   * * * * *  * * * * *
```

Figure E-2 (Part 7 of 7). C/400 Coding for File Transfer Support

Calling File Transfer Support for COBOL/400

Figure E-3 on page E-27 is an example of a COBOL/400 program that provides a data link between one AS/400 system and another AS/400 system. This program reads the file in which the parameters are stored, calls the file transfer support program (QY2FTML), and prints a listing of the parameters, return code, and message number.

The parameters passed to the file transfer program are described in "File Transfer Parameters" on page E-5.

```
5728CB1 R02 M00 891006              IBM AS/400  COBOL/400           KPSLIB/COBDRIVER       05/04/89 14:16:21       Page     1
Program name . . . . . . . . . . . . : COBDRIVER in KPSLIB
Source file  . . . . . . . . . . . . : QCBLSRC in KPSLIB     Member - COBDRIVER     05/04/89 14:16:12
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 . . . . . . . . . . . . . . . . : *SAME
Compiler . . . . . . . . . . . . . . : IBM AS/400  COBOL/400
5728CB1 R02 M00 891006               COBOL SOURCE LISTING            KPSLIB/COBDRIVER       05/04/89 14:16:21       Page     2
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
    1  000100 IDENTIFICATION DIVISION.                                                                 11/04/87
    2  000200 PROGRAM-ID. QY2FTML.                                                                     11/04/87
    3  000300 AUTHOR. EAPOE.                                                                           11/04/87
       000400*                                                                                         11/04/87
    4  000500 ENVIRONMENT DIVISION.                                                                    11/04/87
    5  000600 CONFIGURATION SECTION.                                                                   11/04/87
    6  000700 SOURCE-COMPUTER. IBM-AS400.                                                              05/04/89
    7  000800 OBJECT-COMPUTER. IBM-AS400.                                                              05/04/89
    8  000900 INPUT-OUTPUT SECTION.                                                                    11/04/87
    9  001000 FILE-CONTROL.                                                                            11/04/87
   10  001100     SELECT SEQ-FILE ASSIGN TO DISK-FTTEST                                                01/29/88
   11  001200     ORGANIZATION IS SEQUENTIAL                                                           11/04/87
   12  001300     FILE STATUS IS SEQ-FILE-STATUS.                                                      05/04/89
       001400*                                                                                         11/04/87
   13  001500     SELECT SYSPRT ASSIGN TO PRINTER-QSYSPRT,                                             01/29/88
   14  001600     ORGANIZATION IS SEQUENTIAL                                                           01/29/88
   15  001700     ACCESS IS SEQUENTIAL                                                                 05/04/89
   16  001800     FILE STATUS IS PRINT-FILE-STATUS.                                                    05/04/89
       001900*                                                                                         01/29/88
   17  002000 DATA DIVISION.                                                                           11/04/87
   18  002100 FILE SECTION.                                                                            11/04/87
   19  002200 FD  SEQ-FILE  LABEL RECORDS ARE STANDARD.                                                11/04/87
   20  002300 01  SEQ-FILE-REC.                                                                       11/04/87
   21  002400     02  FILLER   PIC X(101).                                                            11/04/87
       002500*                                                                                         01/29/88
   22  002600 FD  SYSPRT                                                                               01/29/88
   23  002700     LABEL RECORDS ARE OMITTED                                                            02/01/88
   24  002800     LINAGE IS 80 LINES.                                                                  02/01/88
       002900*                                                                                         02/02/88
   25  003000 01  PRINT-FILE-REC.                                                                     11/04/87
   26  003100     02  FILLER PIC X(132).                                                              11/04/87
       003200*                                                                                         02/02/88
   27  003300 WORKING-STORAGE SECTION.                                                                 11/04/87
   28  003400 77  SEQ-FILE-STATUS        PIC X(2).                                                    01/29/88
   29  003500 77  PRINT-FILE-STATUS      PIC X(2).                                                    01/29/88
   30  003600 77  OP-NAME                PIC X(7).                                                    11/04/87
   31  003700 77  ERRORFLAG              PIC X VALUE SPACES.                                          01/29/88
   32  003800 77  EOF                    PIC X VALUE SPACES.                                          01/29/88
   33  003900 77  EOF-FLAG               PIC X VALUE "1".                                             01/29/88
       004000*                                                                                         02/02/88
   34  004100 01  ERRORFLAG              PIC X VALUE SPACES.                                          01/29/88
   35  004200     88  ERROR-OCCURRED      VALUE "1".                                                   01/29/88
       004300*                                                                                         02/02/88
   36  004400 01  HEADER-LINE-1.                                                                      02/01/88
   37  004500     03  FILLER             PIC X(5)  VALUE SPACES.                                      02/01/88
   38  004600     03  FROM-LIBRARY       PIC X(11) VALUE "FRMLIB     ".                               02/01/89
   39  004700     03  FROM-FILE          PIC X(11) VALUE "FRMFIL     ".                               02/01/89
   40  004800     03  FROM-MEMBER        PIC X(11) VALUE "FRMMBR     ".                               02/01/89
   41  004900     03  OBJ-TYPE           PIC X(9)  VALUE "TYPE     ".                                 02/01/89
   42  005000     03  TO-LIBRARY         PIC X(11) VALUE "TOLIB      ".                               02/01/89
```

Figure E-3 (Part 1 of 5). COBOL/400 Coding for File Transfer Support

```
    43  005100     03  TO-FILE            PIC X(11) VALUE "TOFIL      ".                    02/01/89
    44  005200     03  TO-MEMBER          PIC X(11) VALUE "TOMBR      ".                    02/01/89
    45  005300     03  TO-DATE            PIC X(9)  VALUE "TODATE   ".                      02/01/89
    46  005400     03  OPTN               PIC X(7)  VALUE "OPTION ".                        02/01/89
    47  005500     03  REPLCE             PIC X(5)  VALUE "REPL ".                          02/01/89
5728CB1 R02 M00 891006               COBOL SOURCE LISTING            KPSLIB/COBDRIVER      05/04/89 14:16:21      Page     3
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
    48  005600     03  REMOTE-LOCATION    PIC X(9)  VALUE "RMTLOC   ".                      02/01/89
    49  005700     03  RETURN-CODE        PIC X(6)  VALUE "RCODE ".                         02/01/89
    50  005800     03  MESSAGE-NUMBER     PIC X(7)  VALUE "MSGNUM ".                        02/01/89
        005900*                                                                              02/02/88
    51  006000 01  HEADER-LINE-2.                                                          02/01/88
    52  006100     03  FILLER             PIC X(5)  VALUE SPACES.                           02/01/88
    53  006200     03  FILLER             PIC X(11) VALUE "_________ ".                     02/01/89
    54  006300     03  FILLER             PIC X(11) VALUE "_________ ".                     02/01/89
    55  006400     03  FILLER             PIC X(11) VALUE "_________ ".                     02/01/89
    56  006500     03  FILLER             PIC X(9)  VALUE "________ ".                      02/01/89
    57  006600     03  FILLER             PIC X(11) VALUE "_________ ".                     02/01/89
    58  006700     03  FILLER             PIC X(11) VALUE "_________ ".                     02/01/89
    59  006800     03  FILLER             PIC X(11) VALUE "_________ ".                     02/01/89
    60  006900     03  FILLER             PIC X(9)  VALUE "________ ".                      02/01/89
    61  007000     03  FILLER             PIC X(7)  VALUE "______ ".                        02/01/89
    62  007100     03  FILLER             PIC X(5)  VALUE "____ ".                          02/01/89
    63  007200     03  FILLER             PIC X(9)  VALUE "________ ".                      02/01/89
    64  007300     03  FILLER             PIC X(6)  VALUE "_____ ".                         02/01/89
    65  007400     03  FILLER             PIC X(8)  VALUE "_______ ".                       02/01/89
        007500*                                                                              02/02/88
    66  007600 01  CALL-REC.                                                               11/04/87
        007700*  RECORD NUMBER                                                             02/01/88
    67  007800       02  REC-NUM   PIC X(3).                                               02/01/88
        007900*  OPTION                                                                    02/01/88
    68  008000       02  OPTION    PIC X(1).                                               02/01/88
        008100*  REPLACE MEMBER                                                            02/01/88
    69  008200       02  REPL      PIC X(1).                                               02/01/88
        008300*  BLANKS                                                                    02/01/88
    70  008400       02  FILLER    PIC X(4).                                               02/01/88
        008500*  LIBRARY NAME                                                              02/01/88
    71  008600       02  FRMLIB    PIC X(10).                                              02/01/88
        008700*  FILE NAME                                                                 02/01/88
    72  008800       02  FRMFIL    PIC X(10).                                              02/01/88
        008900*  FILE MEMBER                                                               02/01/88
    73  009000       02  FRMMBR    PIC X(10).                                              02/01/88
        009100*  TYPE                                                                      02/01/88
    74  009200       02  TYP       PIC X(6).                                               02/01/88
        009300*  BLANKS                                                                    02/01/88
    75  009400       02  FILLER    PIC X(4).                                               02/01/88
        009500*  TARGET LIBRARY NAME                                                       11/04/87
    76  009600       02  TOLIB     PIC X(10).                                              02/01/88
        009700*  TARGET FILE/LIBRARY NAME                                                  11/04/87
    77  009800       02  TOFIL     PIC X(10).                                              02/01/88
        009900*  TARGET FILE/LIBRARY MEMBER NAME                                           11/04/87
    78  010000       02  TOMBR     PIC X(10).                                              02/01/88
        010100*  TARGET FILE DATE                                                          11/04/87
    79  010200       02  TODATE    PIC X(6).                                               02/01/88
        010300*  BLANKS                                                                    02/01/88
    80  010400       02  FILLER    PIC X(4).                                               02/01/88
        010500*  REMOTE LOCATION                                                           02/01/88
    81  010600       02  RMTLOC    PIC X(8).                                               02/01/88
        010700*  PASSWORD                                                                  11/04/87
    82  010800       02  PASSWD    PIC X(10).                                              02/24/89
        010900*  RETURN CODE                                                               11/04/87
    83  011000       02  RCODE     PIC X(1).                                               02/24/89
5728CB1 R02 M00 891006               COBOL SOURCE LISTING            KPSLIB/COBDRIVER      05/04/89 14:16:21      Page     4
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
        011100*  MESSAGE NUMBER                                                           02/01/88
    84  011200       02  MSGNUM    PIC X(8).                                               11/04/87
        011300*                                                                           02/02/88
    85  011400 01  PRINT-REC.                                                            02/01/88
```

Figure E-3 (Part 2 of 5). COBOL/400 Coding for File Transfer Support

```
     011500*  PRINT RECORD NUMBER                                                  02/01/88
  86 011600     02  PREC-NUM  PIC X(3).                                          02/01/88
     011700*  BLANKS                                                             02/01/88
  87 011800     02  FILLER    PIC X(2).                                          02/01/88
     011900*  PRINT LIBRARY NAME                                                 02/01/88
  88 012000     02  PFRMLIB   PIC X(10).                                         02/01/88
     012100*  BLANKS                                                             02/02/88
  89 012200     02  FILLER    PIC X(1).                                          02/02/88
     012300*  FILE NAME                                                          02/02/88
  90 012400     02  PFRMFIL   PIC X(10).                                         02/02/88
     012500*  BLANKS                                                             02/02/88
  91 012600     02  FILLER    PIC X(1).                                          02/02/88
     012700*  FILE MEMBER                                                        02/02/88
  92 012800     02  PFRMMBR   PIC X(10).                                         02/02/88
     012900*  BLANKS                                                             02/02/88
  93 013000     02  FILLER    PIC X(1).                                          02/02/88
     013100*  TYPE                                                               02/02/88
  94 013200     02  PTYP      PIC X(6).                                          02/02/88
     013300*  BLANKS                                                             02/02/88
  95 013400     02  FILLER    PIC X(3).                                          02/02/88
     013500*  TARGET LIBRARY NAME                                                02/02/88
  96 013600     02  PTOLIB    PIC X(10).                                         02/02/88
     013700*  BLANKS                                                             02/02/88
  97 013800     02  FILLER    PIC X(1).                                          02/02/88
     013900*  TARGET FILE/LIBRARY NAME                                           02/02/88
  98 014000     02  PTOFIL    PIC X(10).                                         02/02/88
     014100*  BLANKS                                                             02/02/88
  99 014200     02  FILLER    PIC X(1).                                          02/02/88
     014300*  TARGET FILE/LIBRARY MEMBER NAME                                    02/02/88
 100 014400     02  PTOMBR    PIC X(10).                                         02/02/88
     014500*  BLANKS                                                             02/02/88
 101 014600     02  FILLER    PIC X(1).                                          02/02/88
     014700*  TARGET FILE DATE                                                   02/02/88
 102 014800     02  PTODATE   PIC X(6).                                          02/02/88
     014900*  BLANKS                                                             02/02/88
 103 015000     02  FILLER    PIC X(5).                                          02/02/88
     015100*  PRINT OPTION                                                       02/01/88
 104 015200     02  POPTION   PIC X(1).                                          02/01/88
     015300*  BLANKS                                                             02/02/88
 105 015400     02  FILLER    PIC X(5).                                          02/02/88
     015500*  PRINT REPLACE MEMBER                                               02/01/88
 106 015600     02  PREPL     PIC X(1).                                          02/01/88
     015700*  BLANKS                                                             02/02/88
 107 015800     02  FILLER    PIC X(3).                                          02/02/88
     015900*  REMOTE LOCATION                                                    02/01/88
 108 016000     02  PRMTLOC   PIC X(8).                                          02/02/88
     016100*  BLANKS                                                             02/02/88
 109 016200     02  FILLER    PIC X(3).                                          02/02/88
     016300*  RETURN CODE                                                        02/01/88
 110 016400     02  PRCODE    PIC X(2).                                          02/02/88
     016500*  BLANKS                                                             02/02/88
5728CB1 R02 M00 891006              COBOL SOURCE LISTING          KPSLIB/COBDRIVER      05/04/89 14:16:21      Page     5
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
 111 016600     02  FILLER    PIC X(2).                                          02/02/88
     016700*  MESSAGE NUMBER                                                     02/01/88
 112 016800     02  PMSGNUM   PIC X(8).                                          02/02/88
     016900*                                                                     02/02/88
 113 017000 PROCEDURE DIVISION.                                                 11/04/87
     017100 DECLARATIVES.                                                       11/04/87
     017200*************************************************************        05/04/89
     017300* SEQUENTIAL DECLARATIVE SECTION                                      05/04/89
     017400*                                                                     05/04/89
     017500*************************************************************        05/04/89
     017600 I-O-ERROR-SEQ SECTION.                                              11/04/87
     017700     USE AFTER STANDARD ERROR PROCEDURE ON SEQ-FILE.                  11/04/87
     017800 I-O-ERROR-PARA-SEQ.                                                 11/04/87
 114 017900     DISPLAY "UNEXPECTED ERROR ON ", OP-NAME, "FOR SEQ FILE".        05/04/89
 115 018000     DISPLAY "FILE STATUS IS " SEQ-FILE-STATUS.                      05/04/89
```

Figure E-3 (Part 3 of 5). COBOL/400 Coding for File Transfer Support

```
    116  018100     SET ERROR-OCCURRED TO TRUE.                                    11/04/87
         018200*************************************************************      05/04/89
         018300* PRINTER FILE DECLARATIVE SECTION                                  05/04/89
         018400*                                                                   05/04/89
         018500*************************************************************      05/04/89
         018600 I-O-ERROR-PRINT SECTION.                                          11/04/87
         018700     USE AFTER STANDARD ERROR PROCEDURE ON SYSPRT.                  01/29/88
         018800 I-O-ERROR-PARA-PRINT.                                             11/04/87
    117  018900     DISPLAY "UNEXPECTED ERROR ON ", OP-NAME, " FOR PRINT FILE".    11/09/87
    118  019000     DISPLAY "FILE STATUS IS ", PRINT-FILE-STATUS.                  11/09/87
    119  019100     SET ERROR-OCCURRED TO TRUE.                                    11/04/87
         019200 END DECLARATIVES.                                                 11/04/87
         019300*                                                                  02/02/88
         019400 MAIN-PROGRAM SECTION.                                             11/04/87
         019500 MAIN-PROCEDURE.                                                   11/04/87
    120  019600     PERFORM OPEN-FILES.                                            02/01/88
    121  019700     PERFORM HDR-PRT.                                               02/01/88
    122  019800     PERFORM READ-REC.                                              05/04/89
    123  019900     PERFORM FILE-TRANS UNTIL EOF = EOF-FLAG.                       01/29/88
    124  020000     PERFORM CLOSE-FILES.                                           11/04/87
    125  020100     STOP RUN.                                                      11/04/87
         020200*                                                                  02/02/88
         020300 FILE-TRANS.                                                       11/04/87
    126  020400     MOVE SPACES TO MSGNUM.                                         11/09/87
    127  020500     MOVE SPACES TO RCODE.                                          11/09/87
    128  020600     CALL "QY2FTML" USING OPTION FRMLIB FRMFIL FRMMBR TYP TOLIB     02/01/88
         020700          TOFIL TOMBR TODATE REPL RMTLOC PASSWD RCODE MSGNUM.       02/01/88
    129  020800     PERFORM PRINT-PARAMETERS.                                      11/09/87
    130  020900     PERFORM READ-REC.                                              05/04/89
         021000*                                                                  05/04/89
         021100 READ-REC.                                                         05/04/89
    131  021200     MOVE "READ" TO OP-NAME.                                        05/04/89
    132  021300     READ SEQ-FILE INTO CALL-REC                                    05/04/89
    133  021400       AT END MOVE EOF-FLAG TO EOF.                                 05/04/89
    134  021500     IF ERROR-OCCURRED GO TO ERROR-TERMINATION.                     05/04/89
         021600*                                                                  02/02/88
         021700 PRINT-PARAMETERS.                                                 11/09/87
    136  021800     MOVE REC-NUM TO PREC-NUM.                                      02/02/88
    137  021900     MOVE OPTION  TO POPTION.                                       02/02/88
    138  022000     MOVE REPL    TO PREPL.                                         02/02/88
5728CB1 R02 M00 891006              COBOL SOURCE LISTING          KPSLIB/COBDRIVER      05/04/89 14:16:21      Page     6
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN S  COPYNAME   CHG/DATE
    139  022100     MOVE FRMLIB  TO PFRMLIB.                                       02/02/88
    140  022200     MOVE FRMFIL  TO PFRMFIL.                                       02/02/88
    141  022300     MOVE FRMMBR  TO PFRMMBR.                                       02/02/88
    142  022400     MOVE TYP     TO PTYP.                                          02/02/88
    143  022500     MOVE TOLIB   TO PTOLIB.                                        02/02/88
    144  022600     MOVE TOFIL   TO PTOFIL.                                        02/02/88
    145  022700     MOVE TOMBR   TO PTOMBR.                                        02/02/88
    146  022800     MOVE TODATE  TO PTODATE.                                       02/02/88
    147  022900     MOVE RMTLOC  TO PRMTLOC.                                       02/02/88
    148  023000     MOVE RCODE   TO PRCODE.                                        02/02/88
    149  023100     MOVE MSGNUM  TO PMSGNUM.                                       02/02/88
    150  023200     MOVE "WRITE" TO OP-NAME.                                       02/01/88
    151  023300     WRITE PRINT-FILE-REC FROM PRINT-REC.                           02/02/88
    152  023400     IF ERROR-OCCURRED GO TO ERROR-TERMINATION.                     11/09/87
         023500*                                                                  02/02/88
         023600 HDR-PRT.                                                          02/01/88
    154  023700     MOVE "WRITE" TO OP-NAME.                                       02/01/88
    155  023800     WRITE PRINT-FILE-REC FROM HEADER-LINE-1.                       02/01/88
    156  023900     WRITE PRINT-FILE-REC FROM HEADER-LINE-2.                       02/01/88
    157  024000     IF ERROR-OCCURRED GO TO ERROR-TERMINATION.                     02/01/88
         024100*                                                                  02/02/88
         024200 OPEN-FILES.                                                       11/09/87
    159  024300     MOVE "OPEN" TO OP-NAME.                                        11/09/87
    160  024400     OPEN I-O SEQ-FILE,                                             11/09/87
         024500              OUTPUT SYSPRT.                                      01/29/88
    161  024600     IF ERROR-OCCURRED GO TO ERROR-TERMINATION.                     11/09/87
```

Figure E-3 (Part 4 of 5). COBOL/400 Coding for File Transfer Support

```
      024700*                                                                   02/02/88
      024800 CLOSE-FILES.                                                       11/09/87
  163 024900     MOVE "CLOSE" TO OP-NAME.                                       11/09/87
  164 025000     CLOSE SEQ-FILE SYSPRT.                                         01/29/88
  165 025100     IF ERROR-OCCURRED GO TO ERROR-TERMINATION.                     11/09/87
      025200*                                                                   02/02/88
      025300 ERROR-TERMINATION.                                                 11/09/87
  167 025400     DISPLAY "I-O ERROR OCCURRED - PROCESS TERMINATION".            11/09/87
  168 025500     STOP RUN.                                                      11/09/87
                          * * * * *  E N D  O F  S O U R C E  * * * * *
5728CB1 R02 M00 891006                 COBOL MESSAGES              KPSLIB/COBDRIVER       05/04/89 14:16:21       Page     7
 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  * * * * *
255 source records read
0 copy records read
0 copy members processed
0 sequence errors
0 was the highest severity message issued
 LBL0901 00  Program COBDRIVER created in library KPSLIB.
                        * * * * *  E N D  O F  C O M P I L A T I O N  * * * * *
```

Figure E-3 (Part 5 of 5). COBOL/400 Coding for File Transfer Support

Calling File Transfer Support for RPG/400

Figure E-4 on page E-33 is an example of an RPG/400 program that provides a data link between one AS/400 system and another AS/400 system. The program shown reads the file in which the parameters are stored, calls the file transfer support program (QY2FTML), and prints a listing of the parameters, return code, and message number.

The parameters passed to the file transfer are described in "File Transfer Parameters" on page E-5.

```
5728RG1 R01M02  881028            IBM AS/400 RPG/400                KPSLIB/RPGDRIVER     02/23/89  17:13:20      Page        1
Compiler . . . . . :   IBM AS/400 RPG/400
Command Options:
  Program  . . . . . :   KPSLIB/RPGDRIVER
  Source file  . . . . . :   *LIBL/QRPGSRC
  Source member  . . . . . :   *PGM
  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
  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 . . . . . :   RPGDRIVER
  File . . . . . :   QRPGSRC
  Library  . . . . . :   KPSLIB
  Last Change  . . . . . :   02/23/89  17:13:06
  Description  . . . . . :   *SAME
5728RG1 R01M02  881028            IBM AS/400 RPG/400                KPSLIB/RPGDRIVER     02/23/89  17:13:20      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  H                                                                                          11/03/87
     200   *   SAMPLE PROGRAM TO READ A RECORD, THEN CALL PROGRAM                                    02/23/89
     300   *   'QY2FTML' TO TRANSFER MEMBER, REPEAT UNTIL LAST                                       02/23/89
     400   *   RECORD, THEN PRINT LISTING.                                                           02/23/89
     500  FFTTEST  IP  F     120           DISK                                                     11/10/87
     600  FQSYSPRT O   F    0132     OF     PRINTER                                                  02/23/89
     700  I*                                                                                         11/03/87
     800  IFTTEST  NS                                                                               11/11/87
* 4137            4137-**
     900  I                                     1   3 INDEX                                        11/12/87
    1000  I                                     4   4 OPTION                                       02/23/89
    1100  I                                     5   5 REPL                                         11/12/87
    1200  I                                    10  19 FRMLIB                                       02/23/89
    1300  I                                    20  29 FRMFIL                                       02/23/89
    1400  I                                    30  39 FRMMBR                                       02/23/89
    1500  I                                    40  45 TYPE                                         02/23/89
    1600  I                                    50  59 TOLIB                                        02/23/89
    1700  I                                    60  69 TOFIL                                        02/23/89
    1800  I                                    70  79 TOMBR                                        02/23/89
    1900  I                                    80  85 TODATE                                       02/23/89
    2000  I                                    90  97 RMTLOC                                       02/23/89
    2100  I                                    98 107 PASSWD                                       02/23/89
    2200  C*                                                                                         11/03/87
    2300  C           QYLIST    PLIST                                                              11/11/87
    2400  C                     PARM           OPTION  1                                           02/23/89
    2500  C                     PARM           FRMLIB 10                                           02/23/89
    2600  C                     PARM           FRMFIL 10                                           02/23/89
    2700  C                     PARM           FRMMBR 10                                           02/23/89
    2800  C                     PARM           TYPE    6                                           02/23/89
    2900  C                     PARM           TOLIB  10                                           02/23/89
    3000  C                     PARM           TOFIL  10                                           02/23/89
    3100  C                     PARM           TOMBR  10                                           02/23/89
    3200  C                     PARM           TODATE  6                                           02/23/89
    3300  C                     PARM           REPL    1                                           11/03/87
    3400  C                     PARM           RMTLOC  8                                           02/23/89
    3500  C                     PARM           PASSWD 10                                           02/23/89
    3600  C                     PARM           RCODE   1                                           02/23/89
    3700  C                     PARM           MSGNUM  8                                           02/23/89
    3800  C*                                                                                         11/03/87
```

Figure E-4 (Part 1 of 3). RPG/400 Coding for File Transfer Support

```
    3900 C                   CALL 'QY2FTML' QYLIST                                  11/11/87
    4000 C*                                                                         11/03/87
    4100 OQSYSPRT H  1     1P                                                       02/23/89
5728RG1 R01M02  881028            IBM AS/400 RPG/400              KPSLIB/RPGDRIVER     02/23/89  17:13:20      Page        3
SEQUENCE                                                                    IND    DO    LAST       PAGE    PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE     LINE    ID
    4200 O       OR        OF                                                     02/23/89
    4300 O                                  11 'FRMLIB'                          02/23/89
    4400 O                                  22 'FRMFIL'                          02/23/89
    4500 O                                  33 'FRMMBR'                          02/23/89
    4600 O                                  42 'TYPE'                            02/23/89
    4700 O                                  52 'TOLIB'                           02/23/89
    4800 O                                  63 'TOFIL'                           02/23/89
    4900 O                                  74 'TOMBR'                           02/23/89
    5000 O                                  86 'TODATE'                          02/23/89
    5100 O                                  95 'OPTION'                          02/23/89
    5200 O                                 101 'REPL '                           02/23/89
    5300 O                                 107 'RMTLOC'                          02/23/89
    5400 O                                 116 'RCODE '                          02/23/89
    5500 O                                 122 'MSGNUM'                          02/23/89
    5600 O       H  1     1P                                                     02/23/89
    5700 O       OR        OF                                                     02/23/89
    5800 O                                  15 '----------'                      11/12/87
    5900 O                                  26 '----------'                      11/12/87
    6000 O                                  37 '----------'                      11/12/87
    6100 O                                  46 '--------'                        11/12/87
    6200 O                                  57 '----------'                      11/12/87
    6300 O                                  68 '----------'                      11/12/87
    6400 O                                  79 '----------'                      11/12/87
    6500 O                                  88 '--------'                        11/12/87
    6600 O                                  95 '------'                          02/23/89
    6700 O                                 100 '----'                            02/23/89
    6800 O                                 109 '--------'                        02/23/89
    6900 O                                 115 '-----'                           02/23/89
    7000 O                                 124 '--------'                        02/23/89
    7100 O       D  2     N1P                                                    02/23/89
    7200 O                         INDEX     3                                   02/23/89
    7300 O                         FRMLIB   15                                   02/23/89
    7400 O                         FRMFIL   26                                   02/23/89
    7500 O                         FRMMBR   37                                   02/23/89
    7600 O                         TYPE     44                                   02/23/89
    7700 O                         TOLIB    57                                   02/23/89
    7800 O                         TOFIL    68                                   02/23/89
    7900 O                         TOMBR    79                                   02/23/89
    8000 O                         TODATE   86                                   02/23/89
    8100 O                         OPTION   92                                   02/23/89
    8200 O                         REPL     98                                   02/23/89
    8300 O                         RMTLOC  109                                   02/23/89
    8400 O                         RCODE   114                                   02/23/89
    8500 O                         MSGNUM  124                                   02/23/89
         * * * * *  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      500  RPG PROVIDES BLOCK OR UNBLOCK SUPPORT FOR FILE FTTEST.
5728RG1 R01M02  881028            IBM AS/400 RPG/400              KPSLIB/RPGDRIVER     02/23/89  17:13:20      Page        4
                    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 FTTEST      DISK         500D  800
   02 QSYSPRT     PRINTER      600D 4100  5600  7100
Field References:
      FIELD       ATTR    REFERENCES (M=MODIFIED D=DEFINED)
      FRMFIL      A(10)   1300D 2600D 7400
      FRMLIB      A(10)   1200D 2500D 7300
      FRMMBR      A(10)   1400D 2700D 7500
      INDEX       A(3)     900D 7200
      MSGNUM      A(8)    3700D 8500
      OPTION      A(1)    1000D 2400D 8100
      PASSWD      A(10)   2100D 3500D
      QYLIST      PLIST   2300D 3900M
      RCODE       A(1)    3600D 8400
```

Figure E-4 (Part 2 of 3). RPG/400 Coding for File Transfer Support

```
       REPL         A(1)    1100D 3300D 8200
       RMTLOC       A(8)    2000D 3400D 8300
       TODATE       A(6)    1900D 3200D 8000
       TOFIL        A(10)   1700D 3000D 7800
       TOLIB        A(10)   1600D 2900D 7700
       TOMBR        A(10)   1800D 3100D 7900
       TYPE         A(6)    1500D 2800D 7600
       'QY2FTML'    LITERAL 3900
Indicator References:
       INDICATOR  REFERENCES (M=MODIFIED D=DEFINED)
       LR          500D
       OF          600D 4200  5700
       1P         4100  5600  7100
     * * * * *  E N D   O F   C R O S S   R E F E R E N C E  * * * * *
5728RG1 R01M02  881028            IBM AS/400 RPG/400                KPSLIB/RPGDRIVER     02/23/89  17:13:20      Page       5
                        M e s s a g e   S u m m a r y
* QRG4137 Severity:  00   Number:    1
          Message . . . . :   Record-Identifying-Indicator entry is blank.
* QRG7086 Severity:  00   Number:    1
          Message . . . . :   The RPG handles blocking function for file.
            INFDS contents updated only when blocks of data transferred.
     * * * * *  E N D   O F   M E S S A G E   S U M M A R Y  * * * * *
5728RG1 R01M02  881028            IBM AS/400 RPG/400                KPSLIB/RPGDRIVER     02/23/89  17:13:20      Page       6
                          F i n a l   S u m m a r y
Message Count:  (by Severity Number)
           TOTAL    00     10     20     30     40     50
             2       2      0      0      0      0      0
Program Source Totals:
   Records . . . . . . . . . . :  85
   Specifications  . . . . . . :  78
   Table Records . . . . . . . :  0
   Comments  . . . . . . . . . :  7
PRM has been called.
Program RPGDRIVER is placed in library KPSLIB. 00 highest Error-Severity-Code.
         * * * * *  E N D   O F   C O M P I L A T I O N  * * * * *
```

Figure E-4 (Part 3 of 3). RPG/400 Coding for File Transfer Support

Calling File Transfer Support for a CL Program

Figure E-5 on page E-36 is an example of a CL program that uses file transfer support to retrieve a database member from a remote system and store it in the library QTEMP. The program uses the Display Physical File Member (DSPPFM) command which allows the user to view the member. The user then has the option to submit the database member as a batch job or end the program.

```
5728SS1 R02 M00 881028                Control Language                KPSLIB/SBMRMTJOB      02/24/89 12:53:35        Page
Program . . . . . :   SBMRMTJOB
  Library . . . . . :     KPSLIB
Source file . . . . . :   QCLSRC
  Library . . . . . :     KPSLIB
Source member name  . . . . . :   SBMRMTJOB   02/24/89 12:53:29
Source printing options . . . . . :   *SOURCE  *XREF  *GEN  *NOSECLVL
Program generation options  . . . . . :   *NOLIST  *NOXREF  *NOPATCH
User profile  . . . . . :   *USER
Program logging . . . . . :   *JOB
Allow RTVCLSRC command  . . . . . :   *YES
Authority . . . . . :   *CHANGE
Text  . . . . . :   Retrieves a member from a remote sys and SBMDBJOB
Compiler  . . . . . :   IBM AS/400 Control Language Compiler
                              Control Language Source
SEQNBR  *...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+.  DATE
   100-
   200-  /*****************************************************************/
   300-  /* This program uses File Transfer Support to retrieve a database  */                          01/24/89
   400-  /* member from a remote system and store it in library QTEMP. The   */
   500-  /* Display Physical File Member(DSPPFM) command is then used to    */
   600-  /* allow the user to view the member.  When the user is finished   */
   700-  /* viewing the member "enter" is pressed to continue.  An inquiry  */
   800-  /* message is then sent to the user asking them if they want       */
   900-  /* to submit this member as a batch job.  If a "Y"                 */                          01/24/89
  1000-  /* response is given, the Submit Database Jobs (SBMDBJOB) command is*/
  1100-  /* used to submit the job to batch.  If a "N" response is given,   */
  1200-  /* the program ends.                                               */
  1300-  /*****************************************************************/
  1400-
  1500-            PGM        PARM(&FROMLIB &FROMFILE &FROMMBR)
  1600-            DCL        VAR(&FROMLIB) TYPE(*CHAR) LEN(10)
  1700-            DCL        VAR(&FROMFILE) TYPE(*CHAR) LEN(10)
  1800-            DCL        VAR(&FROMMBR) TYPE(*CHAR) LEN(10)
  1900-            DCL        VAR(&PASSWORD) TYPE(*CHAR) LEN(10)                                       02/24/89
  2000-            DCL        VAR(&RTNCODE) TYPE(*CHAR) LEN(1) VALUE(' ')
  2100-            DCL        VAR(&SBMJOB) TYPE(*CHAR) LEN(1)                                          01/24/89
  2200-
  2300-    /*************************************************************/
  2400-    /* Retrieve user's password from a data area.               */                                01/24/89
  2500-    /*        (used to preserve security).                      */
  2600-    /*************************************************************/
  2700-            RTVDTAARA  DTAARA(PASSWORD (1 10)) RTNVAR(&PASSWORD)                                02/24/89
  2800-                                                                                                01/24/89
  2900-    /*************************************************************/
  3000-    /* Retrieve the member from the remote system using File    */                               01/24/89
  3100-    /* Transfer Support.  Note the call to File Transfer is     */
  3200-    /* made with both CL variables and constant values. Also    */
  3300-    /* note that File Transfer parameters are all positional.   */
  3400-    /* You must, therefore, reserve space in the call for all   */
  3500-    /* 14 parameters.  If a parameter is not used, fill its     */
  3600-    /* space with blanks.                                       */
5728SS1 R02 M00 881028                Control Language                KPSLIB/SBMRMTJOB      02/24/89 12:53:35        Page    2
                              Control Language Source
SEQNBR  *...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+.  DATE
  3700-    /*************************************************************/
  3800-            CALL       PGM(QSYS/QY2FTML) PARM(R &FROMLIB &FROMFILE +                            01/24/89
  3900                          &FROMMBR '      ' QTEMP '          ' +                                01/24/89
  4000                          '          ' '      ' Y RMTSYS &PASSWORD +                          01/24/89
  4100                          &RTNCODE '        ')                                                 01/24/89
  4200-    /*************************************************************/
  4300-    /* Check the FTS return code to insure a good completion.   */
  4400-    /*************************************************************/
  4500-            IF         COND(&RTNCODE *EQ '0') THEN(DO)
  4600-
```

Figure E-5 (Part 1 of 2). CL Coding for File Transfer Support

```
4700-     /*************************************************************/                    01/24/89
4800-     /* Display the member.                                       */                    01/24/89
4900-     /*************************************************************/                    01/24/89
5000-               DSPPFM     FILE(QTEMP/&FROMFILE) MBR(&FROMMBR)                          01/24/89
5100-                                                                                       01/24/89
5200-     /*************************************************************/
5300-     /* Send the inquiry message to the user.                     */
5400-     /*************************************************************/
5500-               SNDUSRMSG  MSG('Submit the job to batch (Y,N)') VALUES(Y +              01/24/89
5600                             N) DFT(Y) MSGRPY(&SBMJOB)                                 01/24/89
5700-                                                                                       01/24/89
5800-     /*************************************************************/
5900-     /* Check the user's response. If it is "Y", submit the job   */                    01/24/89
6000-     /* to batch using the Submit Database Jobs (SBMDBJOB) command.*/
6100-     /* If the response is "N", return to caller.                 */                    01/24/89
6200-     /*************************************************************/
6300-               IF         COND(&SBMJOB *EQ 'Y') THEN(SBMDBJOB +                        01/24/89
6400                             FILE(QTEMP/&FROMFILE) MBR(&FROMMBR))                      01/24/89
6500-
6600-               ENDDO      /* FTS return code is good */
6700-               ENDPGM
                          * * * * *  E N D  O F  S O U R C E  * * * * *
5728SS1 R02 M00 881028                 Control Language              KPSLIB/SBMRMTJOB     02/24/89 12:53:35        Page    3
                                        Cross Reference

Declared Variables
Name             Defined     Type           Length      References
&FROMFILE          1700      *CHAR            10          1500   3800   5000   6300
&FROMLIB           1600      *CHAR            10          1500   3800   5000   6300
&FROMMBR           1800      *CHAR            10          1500   3800   5000   6300
&PASSWORD          1900      *CHAR            10          2700   3800
&RTNCODE           2000      *CHAR             1          3800   4500
&SBMJOB            2100      *CHAR             1          5500   6300
* CPD0791 00 No labels used in program.
                    * * * * *  E N D  O F  C R O S S  R E F E R E N C E  * * * * *
* CPD0772 00 Program contains commands only valid when run interactively.
5728SS1 R02 M00 881028                 Control Language              KPSLIB/SBMRMTJOB     02/24/89 12:53:35        Page    4
                                        Message Summary

               Severity
Total          0-9  10-19  20-29  30-39  40-49  50-59  60-69  70-79  80-89  90-99
    2            2      0      0      0      0      0      0      0      0      0
Program SBMRMTJOB created in library KPSLIB. Maximum error severity 00.
                              * * * * *  E N D  O F  M E S S A G E  S U M M A R Y  * * * * *
                                * * * * *  E N D  O F  C O M P I L A T I O N  * * * * *
```

Figure E-5 (Part 2 of 2). CL Coding for File Transfer Support

File Transfer Support Messages

File Transfer Support messages give more information about the error condition that occurred on either the local or remote system. The FTS message number itself is available to the application via the Message-ID parameter. The FTS message is logged in the program message queue of the program using FTS and contains a description of the error that occurred.

If a local system error occurs, the FTS message directly identifies the error that occurred.

If a remote system error occurs, FTS-1007 will be logged in the program message queue. The actual remote system error is identified in the message text associated with FTS-1007, and returned to the program in the Message-ID parameter. If the remote system is a AS/400 system, you need to examine the message text of the system message associated with the remote FTS message. The correlation between the system message ID and the FTS message ID are described below. If the remote system is a System/36, the remote FTS message ID is described in the *Using S/36 Communications Manual*.

Table E-5 (Page 1 of 3). File Transfer Messages

FTS Message ID	System Message ID
FTS-1001	CPI7A01
FTS-1002	CPI7A02
FTS-1003	CPD7A03
FTS-1004	CPD7A04
FTS-1005	CPD7A05
FTS-1006	CPD7A06
FTS-1007	CPD7A07
FTS-1008	CPD7A08
FTS-1009	CPD7A09
FTS-1010	CPI7A10
FTS-1011	CPI7A11
FTS-1012	CPD7A12
FTS-1013	CPD7A13
FTS-1014	CPF7A14
FTS-1019	CPD7A19
FTS-1020	CPD7A20
FTS-1021	CPD7A21
FTS-1022	CPD7A22
FTS-1023	CPD7A23
FTS-1024	CPD7A24
FTS-1025	CPD7A25
FTS-1026	CPD7A26
FTS-1027	CPD7A27

Table E-5 (Page 2 of 3). File Transfer Messages

FTS Message ID	System Message ID
FTS-1028	CPD7A28
FTS-1030	CPD7A30
FTS-1031	CPD7A31
FTS-1032	CPD7A32
FTS-1033	CPD7A33
FTS-1034	CPD7A34
FTS-1035	CPD7A35
FTS-1036	CPD7A36
FTS-1037	CPD7A37
FTS-1038	CPD7A38
FTS-1039	CPD7A39
FTS-1040	CPD7A40
FTS-1041	CPD7A41
FTS-1042	CPD7A42
FTS-1043	CPD7A43
FTS-1044	CPD7A44
FTS-1046	CPD7A46
FTS-1049	CPD7A49
FTS-1050	CPD7A50
FTS-1051	CPD7A51
FTS-1052	CPD7A52
FTS-1053	CPD7A53
FTS-1054	CPD7A54
FTS-1055	CPD7A55
FTS-1056	CPD7A56
FTS-1057	CPD7A57
FTS-1058	CPD7A58
FTS-1059	CPD7A59
FTS-1060	CPD7A60
FTS-1063	CPD7A63
FTS-1064	CPD7A64
FTS-1065	CPD7A65
FTS-1066	CPD7A66
FTS-1067	CPD7A67
FTS-1068	CPD7A68
FTS-1069	CPD7A69
FTS-1070	CPD7A70

Table E-5 (Page 3 of 3). File Transfer Messages

FTS Message ID	System Message ID
FTS-1071	CPD7A71

Glossary

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*.

adapter. (1) A part that electrically or physically connects a device to a computer or to another device. (2) A device for attaching parts, for example, parts having different diameters or voltages.

advanced peer-to-peer networking (APPN). Data communications support that routes data in a network between two or more APPC systems that do not need to be adjacent.

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 object authority. A special authority that allows users to use all system resources without having specific authority to the resources. See also *save system authority*, *job control authority*, *security administrator authority*, *service authority*, and *spool control authority*.

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

alphabetic character. (1) Any one of the letters A through Z or a through z or one of the characters #, $, or @. (2) (COBOL) A character that is one of the 26 uppercase letters of the alphabet, or a space. (3) (DDS) (IDDU) Any one of the uppercase letters A through Z or one of the characters #, $, or @.

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*.

American National Standard Code for Information Interchange (ASCII). The code developed by American National Standards Institute for information exchange among data processing systems, data communications systems, and associated equipment. The ASCII character set consists of 7-bit control characters and symbolic characters, plus one parity-check bit.

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

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

APPN. See *advanced peer-to-peer networking (APPN)*.

ASCII. See *American National Standard Code for Information Interchange (ASCII)*.

asynchronous communications. A method of communications supported by the operating system that allows an exchange of data with a remote device, using either a start-stop line or an X.25 line. Asynchronous communications includes the file transport support and the interactive terminal facility support.

asynchronous controller description. A controller description that represents a remote system or device when using asynchronous transmission methods on an asynchronous communications line or when using non-SNA protocols on an X.25 communications line to communicate with the system. See also *generic controller description*.

asynchronous transmission. A method of transmission in which the sending and receiving of data is controlled by control characters instead of by a timing sequence. Contrast with *synchronous transmission*.

asynchronous/SDLC. A data-link level communications protocol that allows data to be transmitted over an asynchronous line using a control protocol similar to SDLC.

authority checking. A function of the system that looks for and verifies a user's authority to an object.

authorization list. A list of two or more user IDs and their authorities for system resources.

auxiliary storage. All addressable disk storage other than main storage.

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

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*.

bidder. The logical unit-to-logical unit half-session defined when a session is started as having to request and receive permission from the other logical unit-to-

logical unit half-session to begin a bracket. Contrast with *first speaker*. See also *bracket protocol*.

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)*.

binary synchronous communications equivalence link (BSCEL) support. The system support that provides BSC communication with another AS/400 system and many other BSC computers and devices.

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

block. (1) A group of records that are recorded or processed as a unit. (2) A set of adjacent records stored as a unit on a disk, diskette, or magnetic tape. (3) In data communications, a group of records that are received, processed, or sent as a unit. (4) A sequential group of statements (defined using line commands) that are processed as a unit.

bps. Bits per second.

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*.

bracket protocol. The rules for controlling the data flow in which exchanges between the two logical unit (LU) half-sessions are achieved through the use of brackets, with one LU assigned at the beginning of the session as first speaker and the other LU as the bidder. The bracket protocol involves bracket start and stop rules. See also *first speaker*.

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

BSCEL support. See *binary synchronous communications equivalence link (BSCEL) support*.

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.

chain. (1) A group of logically linked records. (2) A character set on an impact printer. (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*.

change-direction protocol. A data flow control function in which the sending logical unit stops sending requests, signals the receiving logical unit using the change-direction indicator (in the request/response header of the last request), and prepares to receive requests.

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

CICS/VS. See *Customer Information Control System for Virtual Storage (CICS/VS)*.

CL. See *control language (CL)*.

class of service. The priority level (high, medium, or low) assigned to the transmission groups and intermediate routing nodes included in an advanced peer-to-peer networking (APPN) session after the session is established. See also *class-of-service description*.

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.

CLEAR. A command used to delete all requests and responses related to the active session.

close. The function that ends the connection between a file and a program, and ends the processing. Contrast with *open*.

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

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.

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 attention (CA) key. A keyboard key that can be specified with the CA keyword to request the function specified by the keyword. Data is not returned to the system. Contrast with *command function (CF) key*.

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*.

command function (CF) key. A keyboard key that can be specified with the CF keyword to request the function specified by the keyword. Data is returned to the system. Contrast with *command attention (CA) key*.

command line. The blank line on a display where commands, option numbers, or selections can be entered.

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 configuration. The physical placement of communications controllers, the attachment of communications lines, 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*, *controller configuration*, and *device configuration*.

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 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.

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.

compiler listing. A printout that is produced by compiling a program or creating a file and that optionally includes, for example, a line-by-line list of the high-level language source, a cross-reference list, diagnostic information; and for programs, the description of the externally described files. See also *source listing*.

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*.

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

contention state. In data communication, 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.

control language (CL) program. A program that is created from source statements consisting entirely of control language commands.

control storage. Storage in the computer that contains the programs used to control input and output operations and the use of main storage. Contrast with *main storage*.

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.

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

Customer Information Control System for Virtual Storage (CICS/VS). A licensed program that operates on a host system, such as System/370, 30xx, or 43xx, which can be used in a communications network.

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 dictionary. An object for storing field, record format, and file definitions.

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. Contrast with *source file*. (3) (RJE) A remote job input stream that can contain host system commands and job control language as well as data. Contrast with *command 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 terminal equipment (DTE). That part of a data link that sends data, receives data, and provides the data communications control function according to protocols.

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

database file. An object that contains descriptions of how input data is to be presented to a program from internal storage and how output data is to be presented to internal storage from a program. See also *physical file* and *logical file*.

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

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

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

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*.

delimiter token. A string constant, a delimited identifier, a symbol (for example, ||, /, *, +, or -), or other special characters.

demodulate. To return the frequency of a signal to its original level.

DEVD. See *device description*.

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 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.

DHCF. See *distributed host command facility (DHCF)*.

display file. A device file created by the user to support a display station.

distributed host command facility (DHCF). A function of the operating system that supports the data link between a System/370 terminal using an AS/400 application in an HCF (Host Command Facility) environment.

do group. (1) A set of commands in a control language program defined by a DO command and an ENDDO command that is conditionally processed as a group. (2) (RPG/400) A group of calculations done one or more times based on the results of comparing factor 1 and factor 2 of certain calculation operations (for example, DOUXX). A DO operation and an END operation are the delimiters for a do group.

document library. The system library named QDOC that contains all documents and folders.

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

EBCDIC. See *extended binary-coded decimal interchange code (EBCDIC)*.

EBCDIC character. Any one of the symbols included in the 8-bit EBCDIC set.

ELLC. See *enhanced logical link control (ELLC)*.

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)*.

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*.

expression. (1) (DDS) A pair of values that represents a single parameter value. (2) (C) A group of constants or variables separated by operators that supply a single value.

extended binary-coded decimal interchange code (EBCDIC). A coded character set of 256 eight-bit characters.

external procedure. A procedure that is not contained within a block.

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.

field level specifications. Specifications coded on the same line as a field name or on lines immediately following a field name. See also *file level specifications*, *record level specifications*, *help level specifications*, *join level specifications*, *key field level specifications*, and *select/omit level specification*.

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 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 level specifications. Specifications coded on the lines before the first record format name. See also *field level specifications*, *key field level specifications*, *record level specifications*, *join level specifications*, *select/omit level specifications*, and *help level specifications*.

file name. (1) The name used by a program to identify a file. See also *label*. (2) (COBOL) A name associated with a file and defined in a file description entry or in a sort-merge file description entry.

file transfer support. A function of the operating system that moves file members from one system to another by using asynchronous, APPC, or BSCEL communications support.

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

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. See also *bracket protocol*. Contrast with *bidder*.

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. (1) A defined arrangement of such things as characters, fields, and lines, usually used for displays, printouts, files, or documents. (2) A group of related fields, such as a record, in a file. (3) The arrangement or layout of data on a storage medium, such as disk, tape, or diskette.

FTS. See *file transfer support*.

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.

general-purpose library. The library shipped with the system that contains IBM-provided objects required for many system functions and user-created objects that are not explicitly placed in a different library when they are created. Named QGPL.

generic controller description. An asynchronous controller description that is reserved for incoming calls on an X.25 packet-switching data network from a remote system or device that does not use SNA transmission protocols and whose location name and identifier are defined in configuration list QASYNCLOC in library QSYS. See also *asynchronous controller description*.

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*.

HCF. See *Host Command Facility (HCF)*.

help level specifications. In a display file, specifications coded between the record and field level which define areas on the screen and associate help information with those areas. See also *file level specifications*, *field level specifications*, *join level specifications*, *key field level specifications*, *record level specifications*, and *select/omit level specifications*.

hex. See *hexadecimal*.

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

high-level language (HLL). A programming language, such as RPG, BASIC, PL/I, Pascal, COBOL, and C used to write computer programs.

HLL. See *high-level language (HLL)*.

Host Command Facility (HCF). A feature available on a System/370, 43XX, and 30XX host system that enables a user on the host system to use applications on an AS/400 system or other systems as if they were using remotely attached 5250-type display stations. See also *distributed host command facility (DHCF)*.

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.

IDDU. See *interactive data definition utility (IDDU)*.

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.

indicator. (1) 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. (2) An internal switch used by a program to remember when a certain event occurs and what to do when that event occurs.

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.

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

inquiry message. A message that gives information and requests a reply.

inquiry program. (1) A program that allows an operator to get information from a disk file. (2) A program that runs while the system is in inquiry mode.

interactive data definition utility (IDDU). A function of the operating system that can be used to externally define the characteristics of data and the contents of files.

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

Internet Protocol (IP). A protocol that routes data from its source to its destination in an Internet environment.

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.

invite-program-device operation. An input/output operation that invites an acquired program device to send input to a program and returns control to the program without waiting for the input to arrive.

IPL. See *initial program load (IPL)*.

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

job control authority. A special authority that allows a user to: change, delete, display, hold, and release all files on output queues; hold, release, and clear job queues and output queues; start writers to output queues; hold, release, change, and end other users' jobs; change the class attributes of a job; end subsystems; and start (IPL) the system. See also *all object authority*, *save system authority*, *security administrator authority*, *service authority*, and *spool control authority*.

join level specifications. In a join logical file, specifications coded between the record and field level, which define how to join two physical files. See also *file level specifications*, *field level specifications*, *key field level specifications*, *help level specifications*, *record level specifications*, and *select/omit level specifications*.

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

key field level specifications. Specifications coded on the lines following the last field specification. Key field level specifications are permitted only for physical files or logical files. See also *field level specifications*, *file level specifications*, *record level specifications*, *help level specifications*, *join level specifications*, and *select/omit level specifications*.

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*.

label. (1) The name of a file on a diskette or tape. (2) An identifier of a command or program statement generally used for branching. (3) (RPG/400) A symbolic name that represents a specific location in a program. A label can serve as the destination point for one or more branching operations.

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* and *document library*.

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

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*.

line description. An object that contains information describing a particular communications line that is attached to the system.

line number. The number that precedes a line of information in a printout or on a display. This number can be up to 5 digits long, from 00001 through 99999. See also *sequence number*.

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 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 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 channel. In a packet-switching data network, a path over which data flows between the network and the sending or receiving data terminal equipment.

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 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 6.2. A type of session for communications between peer systems. Synonymous with APPC protocol.

main storage. The part of the processing unit where programs are run. Synonymous with *memory*. Contrast with *auxiliary storage*.

member. Different sets of data within one file. See also *source member*.

memory. Program-addressable storage from which instructions and other data can be loaded directly into registers for subsequent processing. See also *main storage*.

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.

mode description. A system object created for advanced-program-to-program communications (APPC) devices that describe the session limits and the characteristics of the session, such as the maximum number of sessions allowed, maximum number of conversations allowed, the pacing value for incoming and outgoing request/response units, and other controlling information for the session.

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)*.

NAU. See *network addressable unit (NAU)*.

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 addressable unit (NAU). A logical unit, a physical unit, or a system services control point. It is the origin or the destination of information sent by the path control network. See also *logical unit*, *physical unit*, and *system services control point (SSCP)*.

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.

object name. The name of an object. Contrast with *qualified name*.

open. The function that connects a file to a program for processing. Contrast with *close*.

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 token. A numeric constant, an ordinary identifier, a host variable, or a keyword.

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 queue. An object that contains a list of spooled files to be written to an output device, such as a printer.

packet. A data transmission information unit. A group of data and control characters, transferred as a unit, determined by the process of transmission. Commonly used data field lengths in packets are 128 or 256 bytes.

packet assembler/disassembler (PAD). A functional unit that enables data terminal equipment (DTE) not equipped for packet switching to use a packet-switched network.

packet switching. The act of sending and routing packets from source to destination based on information contained in their heading record.

packet-switching data network (PSDN). A communications network that uses packets to send data.

packet-switching host. Any non-SNA, X.25 host system.

PAD. See *packet assembler/disassembler (PAD)*.

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.

parameter list. A list of values that provide a means of associating addressability of data defined in a called program with data in the calling program. It contains parameter names and the order in which they are to be associated in the calling and called program.

peer-to-peer networking. See *advanced peer-to-peer networking (APPN)*.

permanent virtual circuit (PVC). The permanent virtual circuit establishes the identity of the called party within the network services contract. There is no need to identify who is being called when.

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 file member. A named subset of the data records in a physical file. See also *member*.

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.

point-of-sale (POS). Pertaining to a method of providing information to support sales and of collecting the resulting sales information that is collected at retail devices located in stores.

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*. Contrast with *secondary logical unit*.

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.

program initialization parameters (PIP). The initial parameter value(s) passed to a target program as input or used to set up the process environment.

program name. A user-defined word that identifies a COBOL source program.

program object. One of two machine object classifications. It includes those objects used in programs that get their definition from an object definition table. Program objects are used as the parameter or values of machine instructions. Contrast with *system object*.

prompt. (1) A reminder or a displayed request for information or user action. The user must respond to allow the program to proceed. (2) A list of values or a request for information provided by the system as a reminder of the type of information or action required.

PSDN. See *packet-switching data network (PSDN)*.

PVC. See *permanent virtual circuit (PVC)*.

QGPL. See *general-purpose library*.

QLLC. See *qualified logical link control (QLLC)*.

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)*.

qualified name. The name of the library containing the object and the name of the object. Contrast with *object name*.

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.

read operation. An input operation that obtains a record from a file and passes it to a program.

read-from-invited-program-devices operation. An input operation that waits for input from any one of the invited program devices for a user-specified time. Contrast with *read-from-one-program-device operation*.

read-from-one-program-device operation. An input operation that will not complete until the specified device has responded with input. Contrast with *read-from-invited-program-devices operation*.

Recommendation X.25. A document, CCITT Recommendation X.25, that outlines standards for the connection of processing equipment to a packet-switching data network.

record. A collection of related data or words, treated as a unit; such as one name, address, and telephone number.

record level specifications. Specifications coded on the same line as a record format name or on lines immediately following a record format name (until the first field is specified). See also *field level specifications*, *file level specifications*, *key field level specifications*, *help level specifications*, *join level specifications*, and *select/omit level specifications*.

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 job entry (RJE). A function of the AS/400 Communications Utilities licensed program that allows a user to submit a job from a display station on the AS/400 system to a System/370-type host system.

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 system. Any other system in the network with which your system can communicate.

remote work station. A work station that is connected to the system by data communications. Contrast with *local work station*.

request header (RH). A 3-byte header preceding a request unit. See *request/response header*. Contrast with *response header*.

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.

restore. To copy data from tape, diskette, or a save file to auxiliary storage. Contrast with *save*.

retail communications. The data communications support that allows programs on an AS/400 system to communicate with programs on point-of-sale systems, using SNA LU session type 0 protocol.

return code. In data communications, a value sent by the system to a program to indicate the results of an operation by that program.

RJE. See *remote job entry (RJE)*.

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.

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*.

save system authority. A special authority that allows the user to save and restore all objects on the system and free storage of all objects on the system. See also *all object authority*, *job control authority*, *security administrator authority*, *service authority*, and *spool control authority*.

SCS. See *SNA character string*.

SDLC. See *synchronous data link control (SDLC)*.

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 administrator authority. A special authority that allows a user to add users to the system distribution directory, to create and change user profiles, to add and remove access codes, and to perform office tasks, such as delete documents, folders, and document lists, and change distribution lists for other users. See also *all object authority*, *save system authority*, *job control authority*, *service authority*, and *spool control authority*.

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.

select/omit level specifications. Specifications coded on the lines following the last key-field specification. These specifications are permitted only in a logical file. See also *field level specifications*, *file level specifications*, *key field level specifications*, *record level specifications*, *help level specifications*, and *join level specifications*.

selective prompting. A function of the operating system that allows the user to tailor command prompts at a parameter level.

sense data. Data sent with a negative response, indicating the reason for the response.

sequence number. The number of a record that identifies the record within the source member.

service authority. A special authority that allows the user to perform the alter function in the service functions. See also *all object authority*, *save system authority*, *job control authority*, *security administrator authority*, and *spool control authority*.

session. (1) The length of time that starts when a user signs on and ends when the user signs off at a display station. (2) In communications, the logical connection by which a program or device can communicate with a program or device at a remote location. (3) (SNA) A logical connection between two network locations that can be started, tailored to provide various connection protocols, and stopped, as requested. Each session is uniquely identified in a header by a pair of network addresses identifying the origin and destination of any transmissions exchanged during the session. See also *half-session*. (4) (3270 emulation) The activity that occurs on the communications line between the time that the user enters the command to start emulation and the time the user ends the emulation job. (5) (RJE) The activity of all tasks within a single AS/400 system communicating with a single host system.

SNA. See *Systems Network Architecture (SNA)*.

SNA character string (SCS). A data stream composed of EBCDIC controls, optionally intermixed with end-user data, which is carried within a request/response unit.

SNA distribution services (SNADS). An IBM asynchronous distribution service that defines a set of rules to receive, route, and send electronic mail in a network of systems.

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.

SNA remote job entry. See *remote job entry (RJE)*.

SNA upline facility (SNUF). The communications support that allows the AS/400 system to communicate with CICS/VS and IMS/VS application programs on a host system. For example, DHCF communicates with HCF and DSNX communicates with NetView Distribution Manager.

SNA 3270 API. See *SNA 3270 application program interface (SNA 3270 API)*.

SNA 3270 application program interface (SNA 3270 API). A function that allows an application program with a System/370, 30xx, or 43xx VTAM program by sending and receiving 3270 data streams.

SNADS. See *SNA distribution services (SNADS)*.

SNUF. See *SNA upline facility (SNUF)*.

source file. (1) A file of programming code that is not compiled into machine language. Contrast with *data file*. (2) A file created by the specification of FILETYPE(*SRC) on the Create command. A source file can contain source statements for such items as high-level language programs and data description specifications.

source listing. A portion of a compiler listing that contains source statements and, optionally, test results. See also *compiler listing*.

source member. A member of a database source file that contains source statements such as RPG/400, COBOL, BASIC, PL/I, or DDS statements. See also *member*.

source program. (1) A set of instructions that are written in a programming language and must be translated to machine language before the program can be run. (2) In communications, the program that starts a session with a remote system. Contrast with *target program*.

source system. The system that issues a request to establish communications with another system. (DDM) The system on which an application program issues a request to use a remote file. Contrast with *target system*.

spool control authority. A special authority that allows the user to perform spooling functions, such as display, delete, hold, and release spooled files on the output queue for himself and other users. This authority also allows the user to change the spooled file attributes, such as the printer used to print the file. See also *all object authority*, *save system authority*, *job control authority*, *security administrator authority*, and *service authority*.

spooled file. A file that holds output data waiting to be printed, or input data waiting to be processed by the program.

spooled input file. See *inline data file*.

spooled output file. A file that causes output data to be held for later printing.

statement. An instruction in a program.

storage pool. A logical division of storage reserved for processing a job or group of jobs.

store controller. A controller in a network that is used to collect data from and provide support for the point-of-sale and administrative devices within the retail system. The store controller also provides some local data processing capabilities.

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.

SVC. See *switched virtual circuit (SVC)*.

switched virtual circuit (SVC). A circuit established to the called party when the calling party requests a connection. Contrast with *permanent virtual circuit (PVC)*.

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 transmission. A method of transmission in which the sending and receiving of data is controlled by timing signals. Contrast with *asynchronous transmission*.

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 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/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.

Systems Network Architecture distribution services. See *SNA distribution services (SNADS)*.

target. (1) In advanced program-to-program communications, the program or system to which a request for processing is sent. (2) (DDM) The remote system where the request for a file is sent.

target program. (1) In communications, the program that is started on the remote system at the request of the source system. Contrast with *source program*. (2) In display station pass-through, a program that runs on the remote system.

target system. In a distributed data management (DDM) network, the system that receives a request from an application program on another system to use one or more files located on the target system. Contrast with *source system*.

TCP. See *Transmission Control Protocol (TCP)*.

TCP/IP. See *Transmission Control Protocol/Internet Protocol (TCP/IP)*.

token. A predefined message or character pattern that gives the receiver of the token the permission to transmit information.

token-ring network. A local area network that sends data in one direction throughout a specified number of locations by using the symbol of authority for control of the transmission line, called a token, to allow any sending station in the network (ring) to send data when the token arrives at that location.

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*.

transmission control characters. In data communications, special characters that are included in a message to control communications over a data link. For example, the sending station and the receiving station use transmission control characters to exchange information; the receiving station uses transmission control characters to indicate errors in data it receives.

transmission control layer. The layer within a half-session that synchronizes and controls the speed of session-level data traffic, checks sequence numbers of requests, and enciphers and deciphers end-user data.

Transmission Control Protocol (TCP). A host-to-host protocol that provides transmission in an internet environment. TCP assumes Internet Protocol as the underlying protocol.

Transmission Control Protocol/Internet Protocol (TCP/IP). A set of vendor-independent communications protocols that support peer-to-peer connectivity functions for both local and wide area networks.

transmission header. Control information, optionally followed by a basic information unit or a basic information unit segment, that is created and used by path control to route messages within the network. Abbreviated TH.

truncate. (1) To cut off data that cannot be printed or displayed in the line width specified or available. (2) To cut off data that does not fit in the specified field length in a field definition.

turnaround. Pertaining to changing a communications line from being able to send to being able to receive, or from being able to receive to being able to send.

unbind command. A command used to reset the protocols for a session. Contrast with *bind command*.

user ID. See *user identification (user ID)*.

user identification (user ID). (1) The name used to associate the user profile with a user when a user signs on the system. See also *user profile name*. (2) The first part of a two-part network name used in the system distribution directory and in the office applications to uniquely identify a user. The network name is usually the same as the user profile name, but does not need to be. See also *common user identification (common user ID)*.

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)*.

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.

virtual circuits. A logical, rather than a physical, connection that is established and controlled by a managing network in a packet-switching communications environment. See also *permanent virtual circuit (PVC)* and *switched virtual circuit (SVC)*.

virtual machine (VM). (1) A system where each user appears to have his own computer and input/output devices. (2) A functional simulation of a computer and its associated input/output devices

Virtual Machine Facility (VM/370). A time sharing system control program that consists of: (a) a control program (CP) managing resources of an IBM System/370 computing system so that multiple remote terminal users have a functional simulation of a computing system (a virtual machine) at their disposal, and (b) the conversational monitor system (CMS), which provides general time sharing, program development, and problem solving functions.

Virtual Telecommunications Access Method (VTAM). A set of programs that control communications between terminals and application programs running under the DOS/VS, OS/VS1, and OS/VS2 operating systems.

VM/370. See *Virtual Machine Facility (VM/370)* .

VTAM. See *Virtual Telecommunications Access Method (VTAM)*.

work station. A device used to transmit information to or receive information from a computer; for example, a display station or printer.

write operation. An output operation that sends a processed record to an output device or output file.

X.25. In data communications, a specification of the CCITT that defines the interface to an X.25 (packet-switching) network.

3180 display station. Display station that uses the 5250 data stream.

5250 display station. Any display station from the IBM 5250 Information Display System or the 5290 Display System; or the 3180 display station. A 3270 display station is not a 5250 display station.

Index

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

S

T

U

V

W

Numerics

Special Characters

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
Alo

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
Alor

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 o
Along

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®

Cu
Al

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
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
Alo

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 o
Along

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

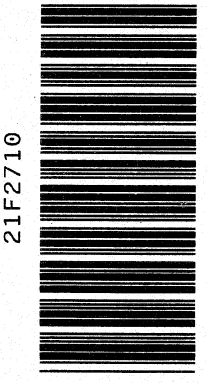
International Business Machines Corporation
Information Development
Department 245
3605 North Hwy 52
ROCHESTER MN 55901-9986

Fold and Tape **Please do not staple** Fold and Tape

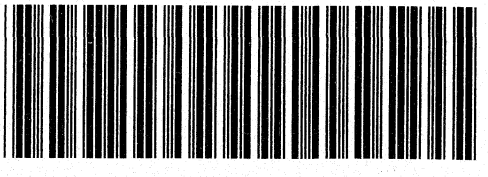
IBM®

Cut
Alon

Program Number
5728-SS1



SC21-9590-1



Printed in U.S.A.