AS/400

ICF Programming

Version 4

SC41-5442-00

AS/400

ICF Programming

Version 4

SC41-5442-00

Take Note!

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

First Edition (August 1997)

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

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

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

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

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

or you can fax your comments to:

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

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

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

Contents

Figures

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Subject to IBM's valid intellectual property or other legally protectable rights, any functionally equivalent product, program, or service may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

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

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

Address your questions to:

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

This publication could contain technical inaccuracies or typographical errors.

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

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

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

Programming Interface Information

This programming book is intended to help application programmers write communications programs that use the intersystem communications function (ICF). It primarily contains reference information which allows the customer to write programs that use the services of ICF. ICF contains no programming interfaces for customers.

Trademarks and Service Marks

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

Advanced Peer-to-Peer Networking
AnyNet
Application System/400
APPN
AS/400
C/400
CICS
COBOL/400
FORTRAN/400
IBM
ILE
Integrated Language Environment
Operating System/2
Operating System/400
OS/2
OS/400
RPG IV
RPG/400
SAA
Systems Application Architecture
System/370
System/390
VTAM
400

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

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

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

C-bus is a trademark of Corollary, Inc.

Java and HotJava are trademarks of Sun Microsystems, Inc.

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

About ICF Programming, SC41-5442

This book contains programming information for writing application programs that use the intersystem communications function (ICF).

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

Who Should Use This Book

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

To work with the information in this book, you should have knowledge of general communications concepts. AS/400 communications concepts are covered in the *AS/400 Advanced Series Handbook*.

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

AS/400 system programming terminology and programming using the ILE C, ILE COBOL, or ILE RPG languages.

Prerequisite and Related Information

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

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

Information Available on the World Wide Web

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

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

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

Chapter 1. Introduction to AS/400 System Communications

This chapter describes, in general, the different sources and background information needed to use intersystem communications function (ICF) communications on the IBM* AS/400 Advanced Series* (AS/400*) system. **ICF** is a function of the operating system that allows a program to communicate interactively with another program or system. Detailed instructions are available in other books referred to in this chapter.

Planning for Data Communications

Data communications planning should already be complete.

Installing Communications Hardware

Communications hardware, such as modems and cables, must be installed before you can start running your programs. (The exception is intrasystem communications, which requires no hardware installation.) However, if your hardware is not yet installed, you can read this book and begin writing your programs.

Configuring Your System for Data Communications

The *Communications Configuration* book explains how to configure for communications. Although you cannot run your application programs until the system is properly configured, you can read this book and begin writing your programs.

You need to configure the remote system to allow communications with the AS/400 system. The *Communications Configuration* book contains configuration considerations for some remote systems when communicating with an AS/400 system.

Writing Programs that Use the Intersystem Communications Function (ICF)

You can write communications programs using the Integrated Language Environment* (ILE*) C/400*, ILE COBOL*, and ILE RPG* languages. 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 book for the communications type you are using (for example, the *APPC Programming* book), the programming language books for the language you plan to use, and the *DDS Reference* book.

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 *System Operation* book and the *CL Reference* book 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)
  - APPC over Transmission Control Protocol/Internet Protocol (TCP/IP)
  - 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 and non-IBM systems that support the SNA logical unit type 6.2 (LU 6.2) architecture. Using APPC allows system functions and application programs on the system to communicate with other system functions or application programs on:

- The same system
- Another AS/400 system
- A System/38
- A System/36
- Any other system (such as CICS* with similar levels of APPC support.

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. APPC also allows AS/400 system application programs to start other application programs on the local system. The networking capability of data communications support that routes data in a network between two or more APPC systems that do not need to be adjacent, called **Advanced Peer-to-Peer Networking* (APPN*)** support, is available through the APPC interface.

An APPC conversation cannot be used by both the System/38 environment and the AS/400 operating environment. A diagnostic message is sent to an application attempting to open an ICF file in the AS/400 operating environment, to accept a conversation using Common Programming Interface (CPI) Communications, or to open either a communications file or a mixed device file in the System/38 environment for the same APPC conversation. Only one interface can be used for any conversation.

Refer to the *APPC Programming* book for more information.

APPC over TCP/IP: APPC over TCP/IP allows the system to communicate with other systems that support the SNA logical unit type 6.2 (LU 6.2) architecture running over TCP/IP. This support must be compliant with the Multiprotocol Transport Networking (MPTN) architecture, such as the support in the IBM AnyNet* products. Refer to the *Multiprotocol Transport Networking (MPTN) Architecture: Technical Overview* book, GC31-7073, for more information about MPTN.

APPC programs running over TCP/IP networks should see little or no difference than if they ran over SNA networks. Therefore, information in this book that applies to APPC also applies to APPC over TCP/IP (unless otherwise noted). Some additional configuration is required for APPC over TCP/IP. Refer to the *Communications Configuration* book for information about configuring for APPC over TCP/IP.

Examples of systems that support APPC over TCP/IP include:

- Operating System/400* (OS/400*)
- Operating System/2* (OS/2*)
- Multiple Virtual Storage (MVS)

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* computer
- System/390* computer

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.

SNA 3270 Program Interface: The SNA 3270 program interface allows an AS/400 application to communicate with a host application by sending and receiving 3270 data streams.

Refer to the *SNA Upline Facility Programming* book 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 Programming* book for more information.

Asynchronous Communications: Asynchronous communications is 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. 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 application programs to start programs on remote systems, and allows remote programs to start programs on the AS/400 system.

Refer to the *Asynchronous Communications Programming* book 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 book with regard to intrasystem communications, they refer to the program with which your program is communicating.

Refer to the *Intrasystem Communications Programming* book for more information.

Finance Communications: Finance communications allows you to attach 3601, 3694, 4701, 4702, 4730, 4731, 4732, 4736, 4737, and Financial Branch System Services (FBSS) controllers to your AS/400 system using **synchronous data link control (SDLC)**. SDLC is a form of communications line control that uses commands to control the transfer of data over a communications line. You can also attach the 4701, 4702, 4737, and Financial Branch System Services controllers using X.25.

Note: 4737 self-service transaction station controllers are configured as FBSS controllers.

In addition, you can attach controllers configured as FBSS controllers using a token-ring or Ethernet local area network (LAN). Since these controllers do not support Ethernet networks, you must use an 8209 bridge when you use an Ethernet configuration on the AS/400 system.

While programs using ICF can communicate with any of the finance controllers, programs that do not use ICF can communicate only with controllers configured as a 3694, 4701, and 4702 on the AS/400 system.

Refer to the *Finance Communications Programming* book 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. X.25 is supported for a 4684 controller, provided it has Retail Industry Programming Support Services (RIPSS) 3.01, a program that provides access to the application files on the 4684 controller.

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 Programming* book 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)
- Common Programming Interface (CPI) Communications
- User-defined communications
- Sockets

Because these communications functions are not part of ICF, they are described in other books, which are identified in the list of related books in the Bibliography. Refer also to the *Publications Reference* book.

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 program interface for remote work stations is the same as the program interface for local work stations. Refer to the *Application Display Programming* book 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**, or the physical placement of communications controllers, the attachment of communications lines, and so on, 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
- Ethernet network
- Frame Relay
- IDLC
- DDI
- Wireless

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

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 network interface, network server, line, controller, and device descriptions.

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

An **integrated services digital network (ISDN)**, which is a network that can provide voice, data, and image over the same communications line, requires network interface descriptions and connection lists. The configuration support provides this function.

A **File Server I/O Processor (FSIOP)**, which is an input/output processor (IOP) that serves files, requires a network server description.

Refer to the *Communications Configuration* book 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 *APPN Support* book 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* book.)

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 Configuration* book.

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 – Reference* book 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 the ILE C, ILE COBOL, and ILE RPG programming languages can access the return codes to help diagnose problems. In addition, messages are entered in the job log to identify the error that occurred. The ILE COBOL and ILE RPG programming languages provide language-defined file status that can be used either in place of, or in addition to, the major and minor return code. The ILE C programming language 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 Management* book 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.

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

The programs presented in this book 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; this program then starts the program on the remote AS/400 system.
2. Although communications types in these examples are advanced program-to-program communications (APPC) and binary synchronous communications equivalence link (BSCEL), the examples provide 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 book 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.

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 book. Refer to the appropriate language reference book for more information.

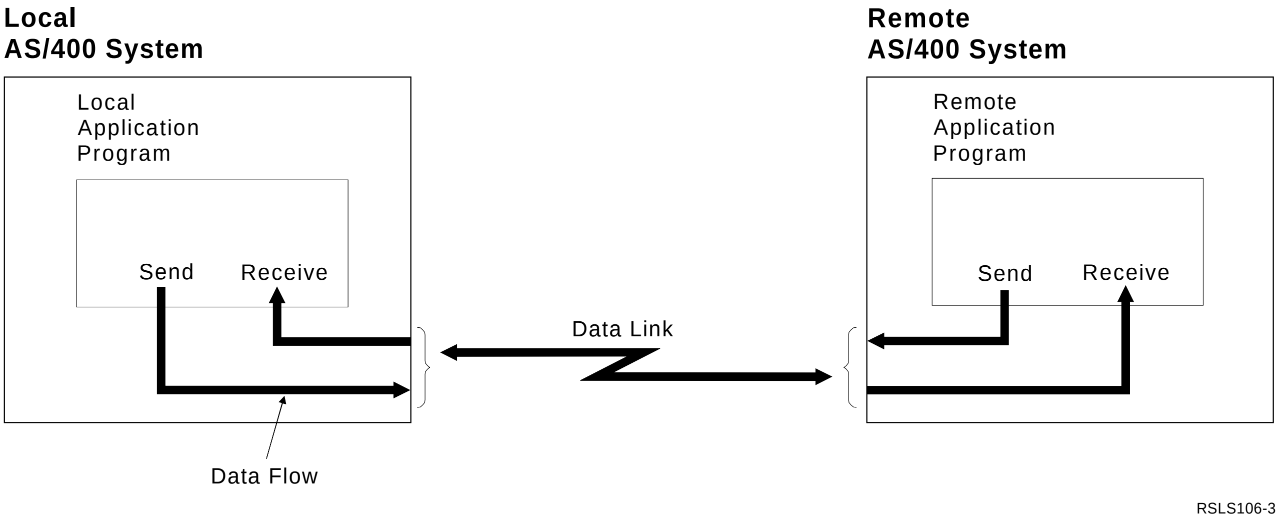


Figure 3-1. Sending Data from a Local Program to a Remote Program

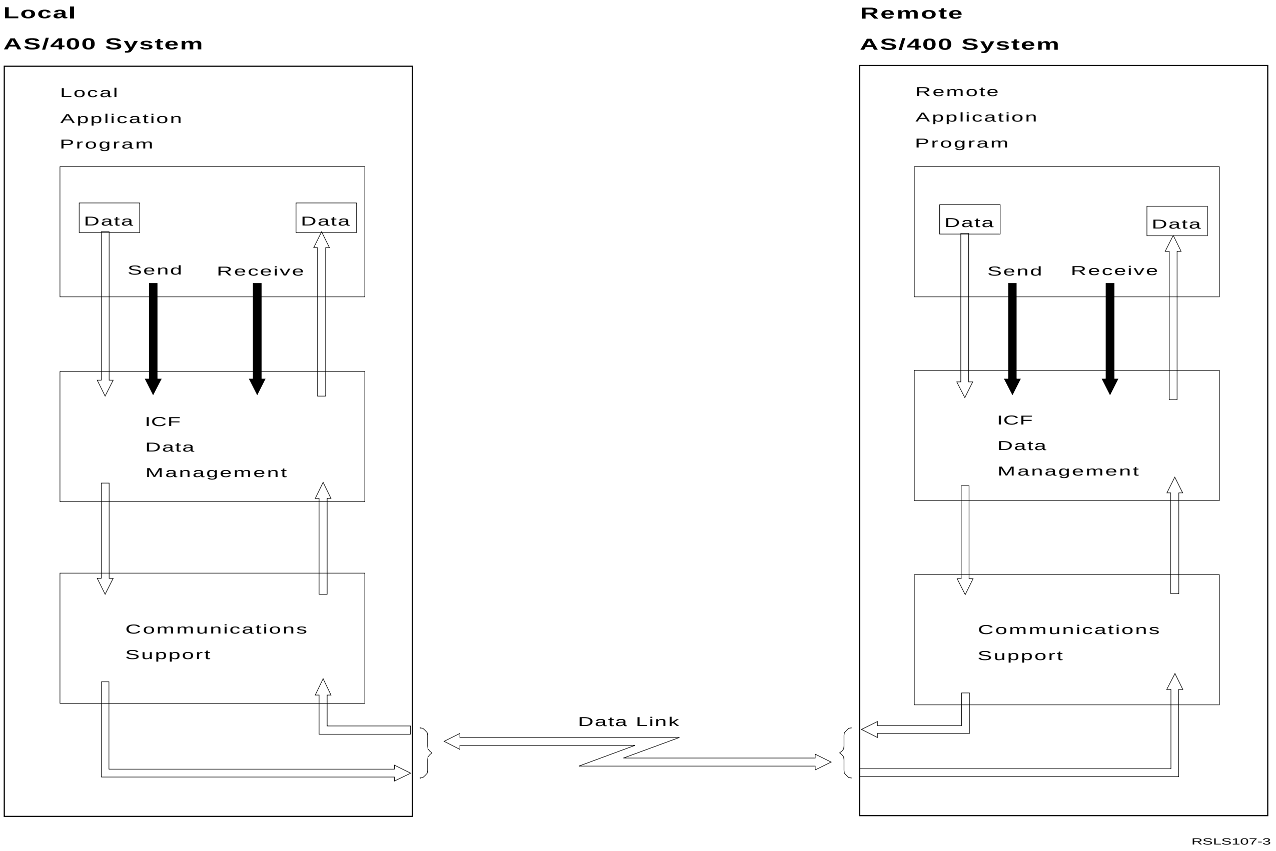


Figure 3-2. Major Parts of ICF Data Management

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.

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 book 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 book.

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 configuration objects:

- Line descriptions
- Controller descriptions
- Device descriptions
- Mode descriptions
- Class-of-service descriptions
- Network interface descriptions
- Network server descriptions
- Connection lists
- Configuration lists

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

Part of the connection between the application and configuration is through the remote location name that is defined as part of the device description. Refer to “The Intersystem Communications Function File” for more information on how this connection is made.

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. Refer to the *Communications Management* book for more information on using these commands. Refer to the *Work Management* book 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 network interface, line, controller, network server, and device descriptions.

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

  Note: 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 Management* book for more information on the VRYCFG command 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 3-4 shows the relationship between the program, the ICF file, and the communications configurations.

Not all of the communications types require that all of these configurations be explicitly created. Also note that the network interface description is only required when communicating across an ISDN.

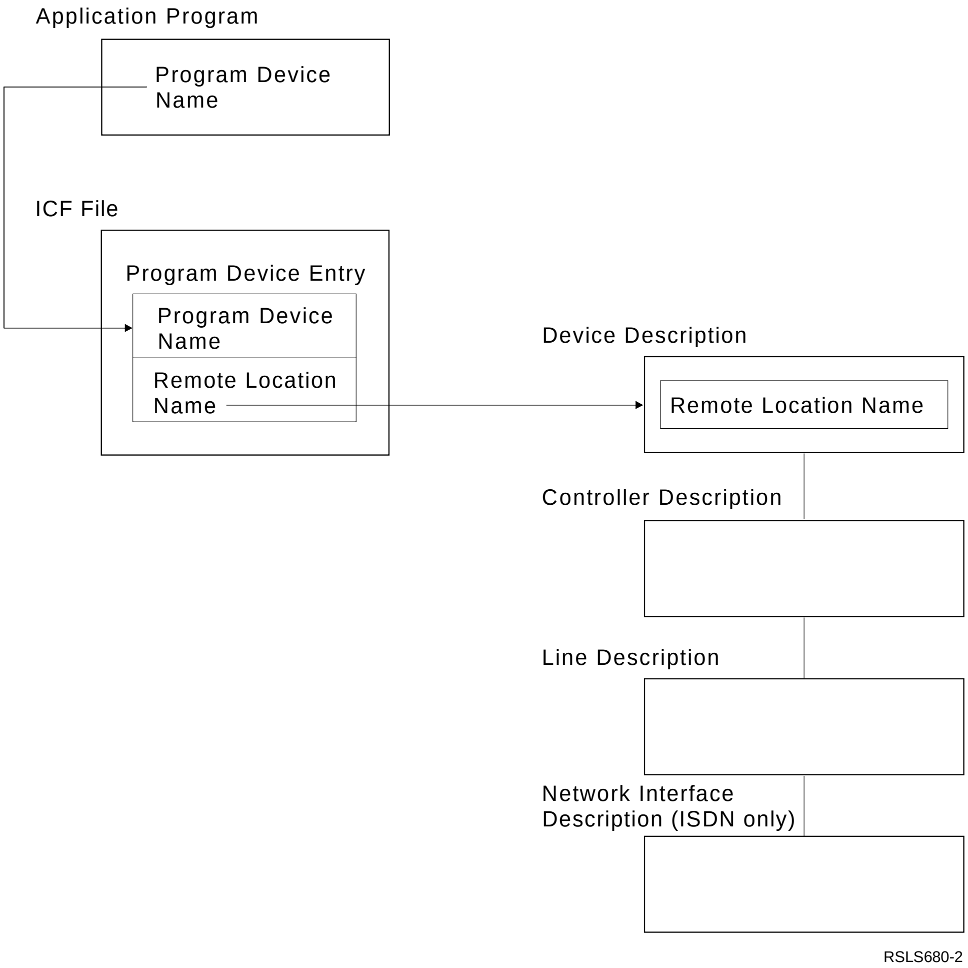


Figure 3-3. ICF File-Configuration Relationship

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 3-6 for more information about the program start request. Refer to the appropriate communications programming book 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 on page 3-5, your program establishes a session using an acquire operation with PGMDEVA specified as the program device name.

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.

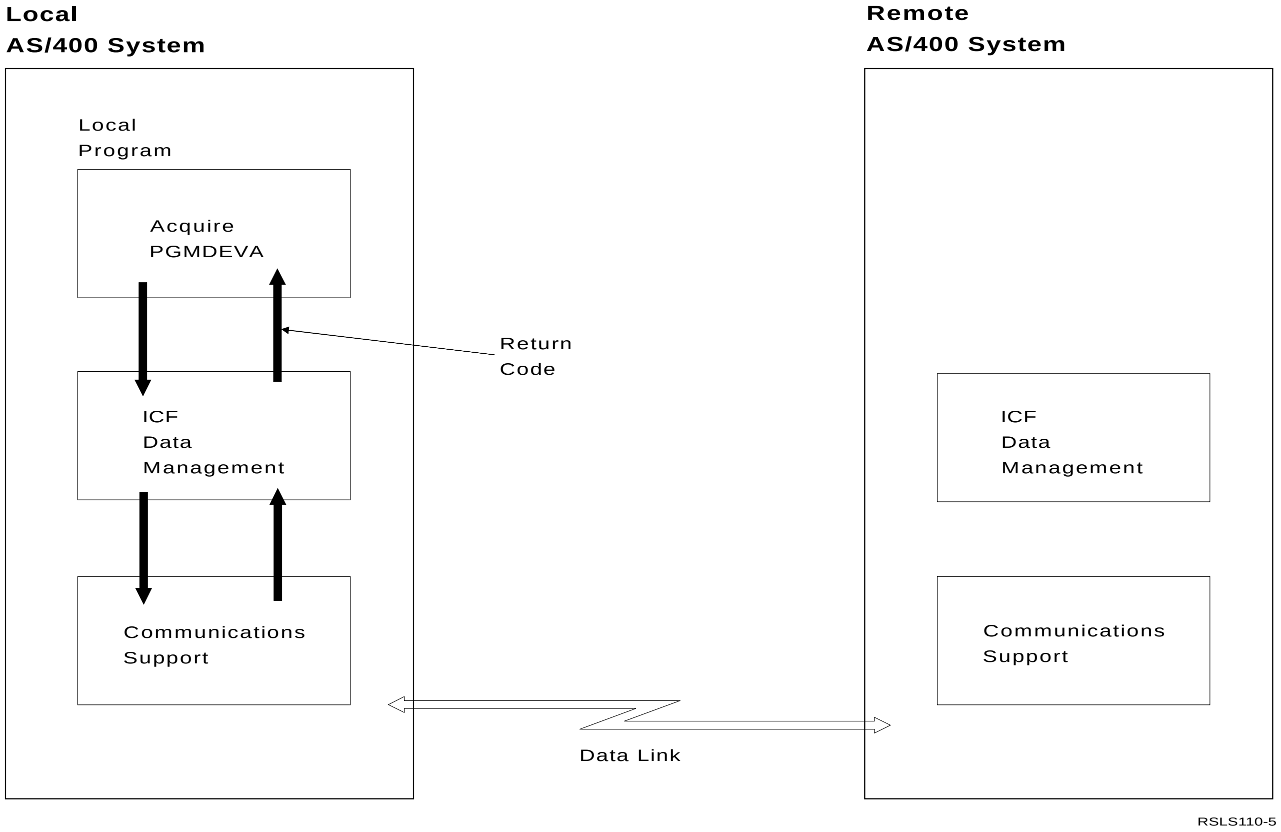


Figure 3-4. Establishing a Session

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

   .
   .
   .
   OPEN ICFFILE
   .
   .
   .
   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 system, as shown in Figure 3-5.

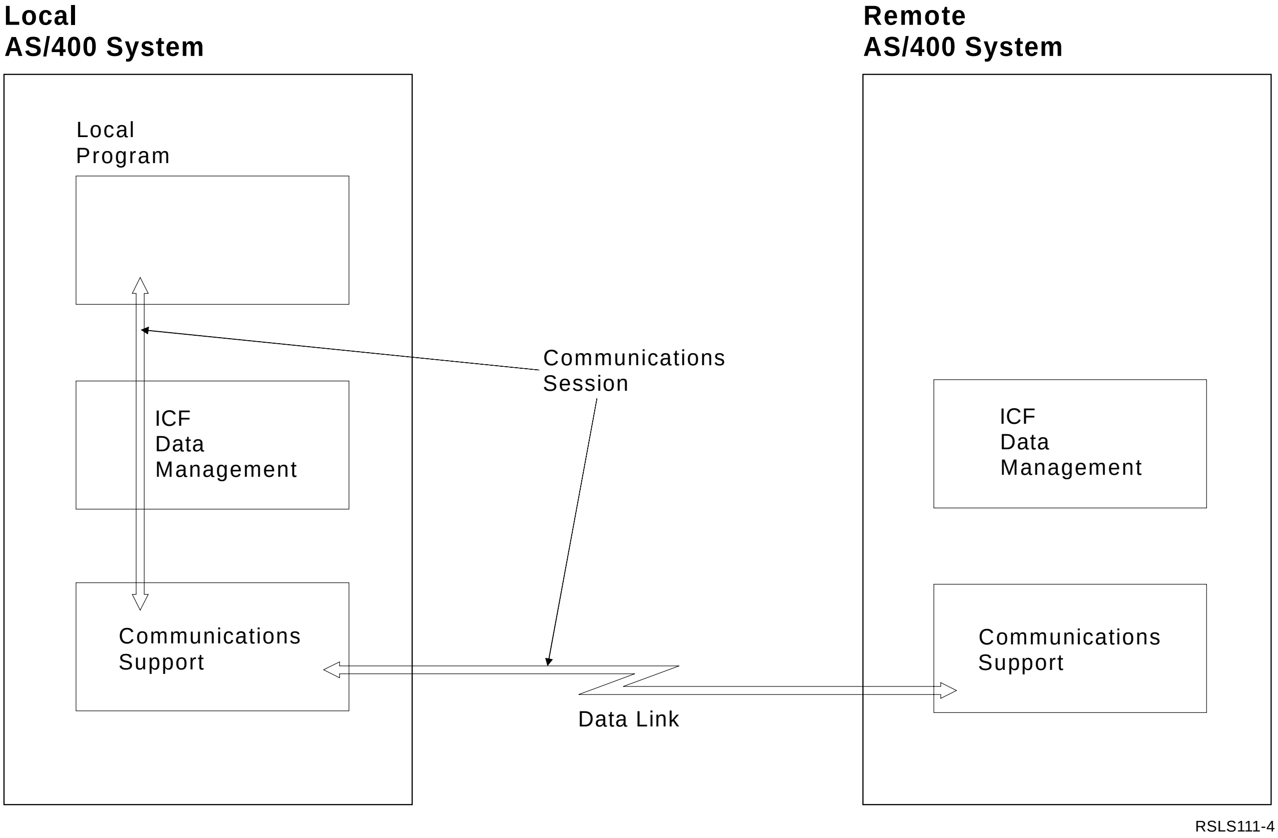


Figure 3-5. Communications Session Established

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” 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 on page 3-7, the evoke function is used to start the program named TGTPGM at the remote system.

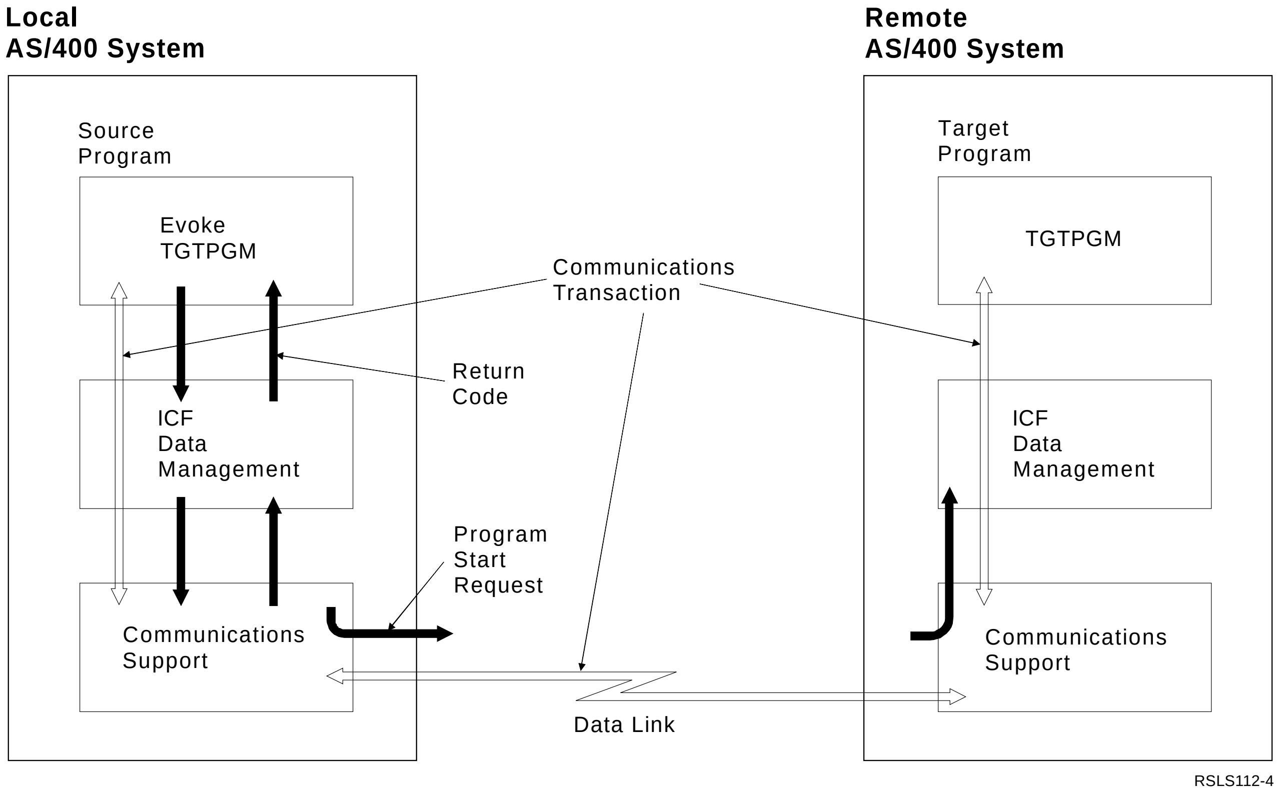


Figure 3-6. Program Started at Remote System by Evoke Function

A return code is always given to your program to indicate the status of the evoke function unless the program start request is delayed by use of the DFREVOKE keyword. 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 book 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.

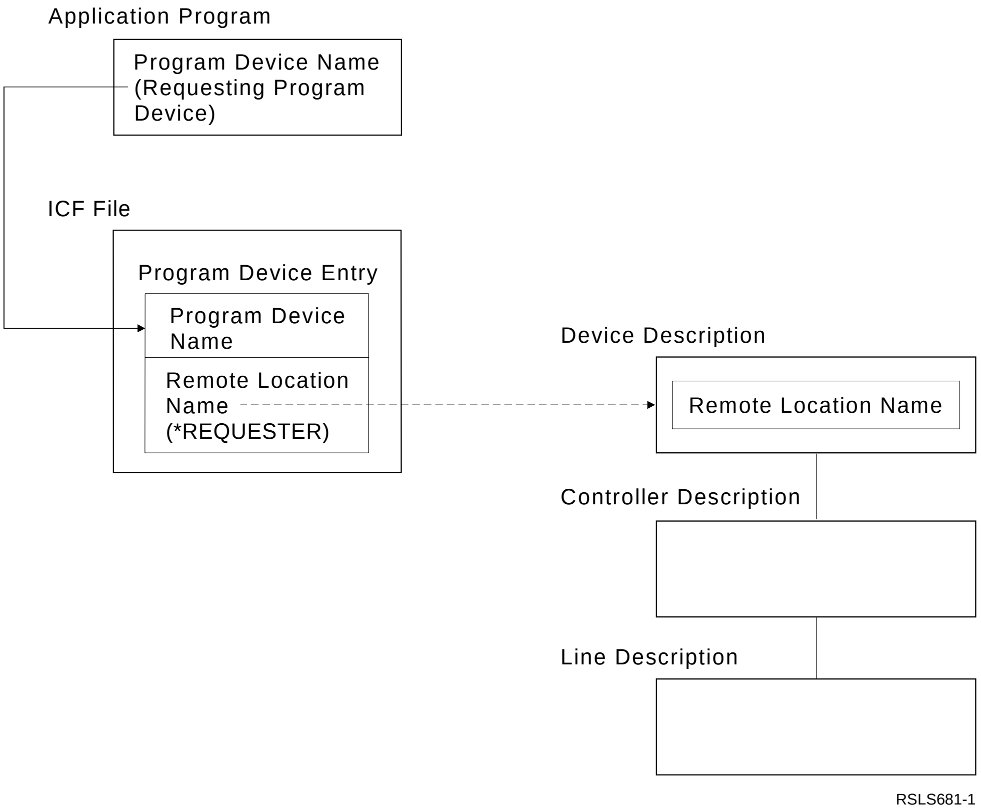


Figure 3-7. Requesting Program Device Relationship

Figure 3-7 on page 3-8 shows the relationship between the program, the ICF file, and the communications configurations for a requesting program device.

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

   .
   .
   .
   OPEN ICFFILE
   .
   .
   .
   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 on page 3-9, 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).

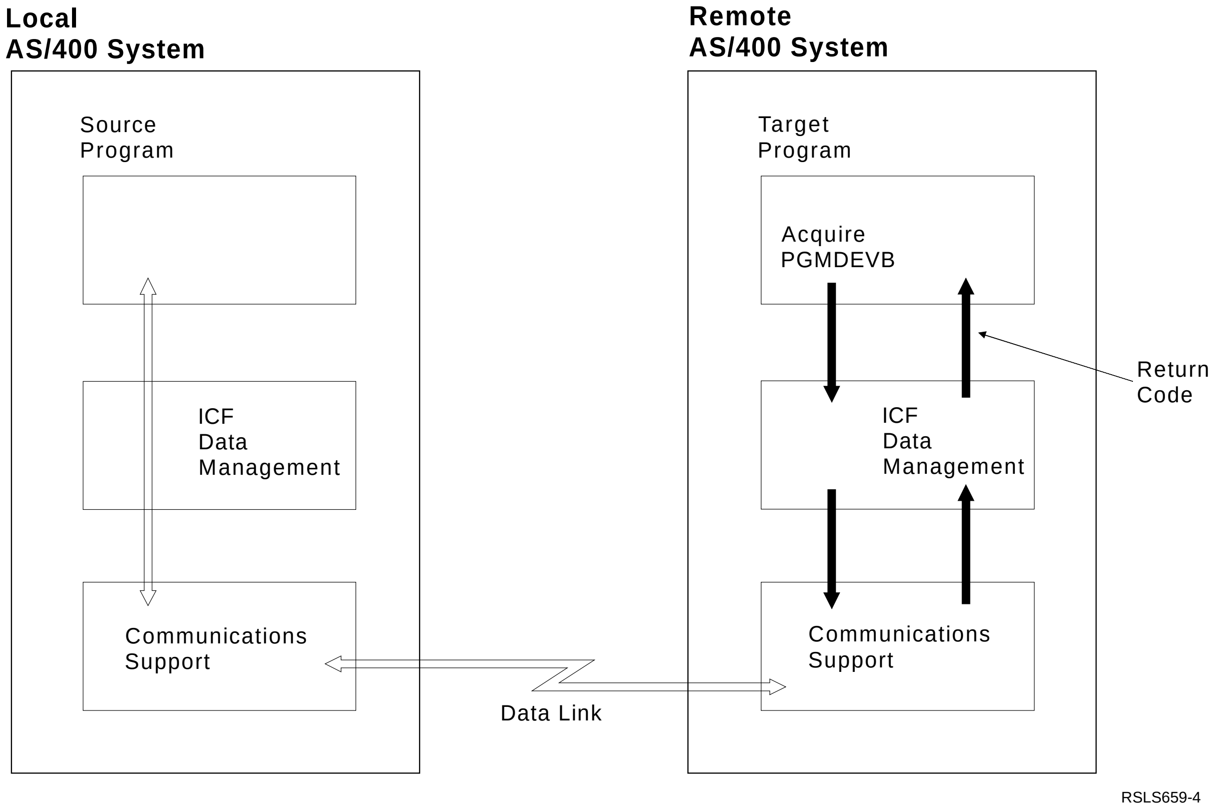


Figure 3-8. Establishing a Logical Connection between the Target Program and the Session

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 on page 3-10, 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.

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.

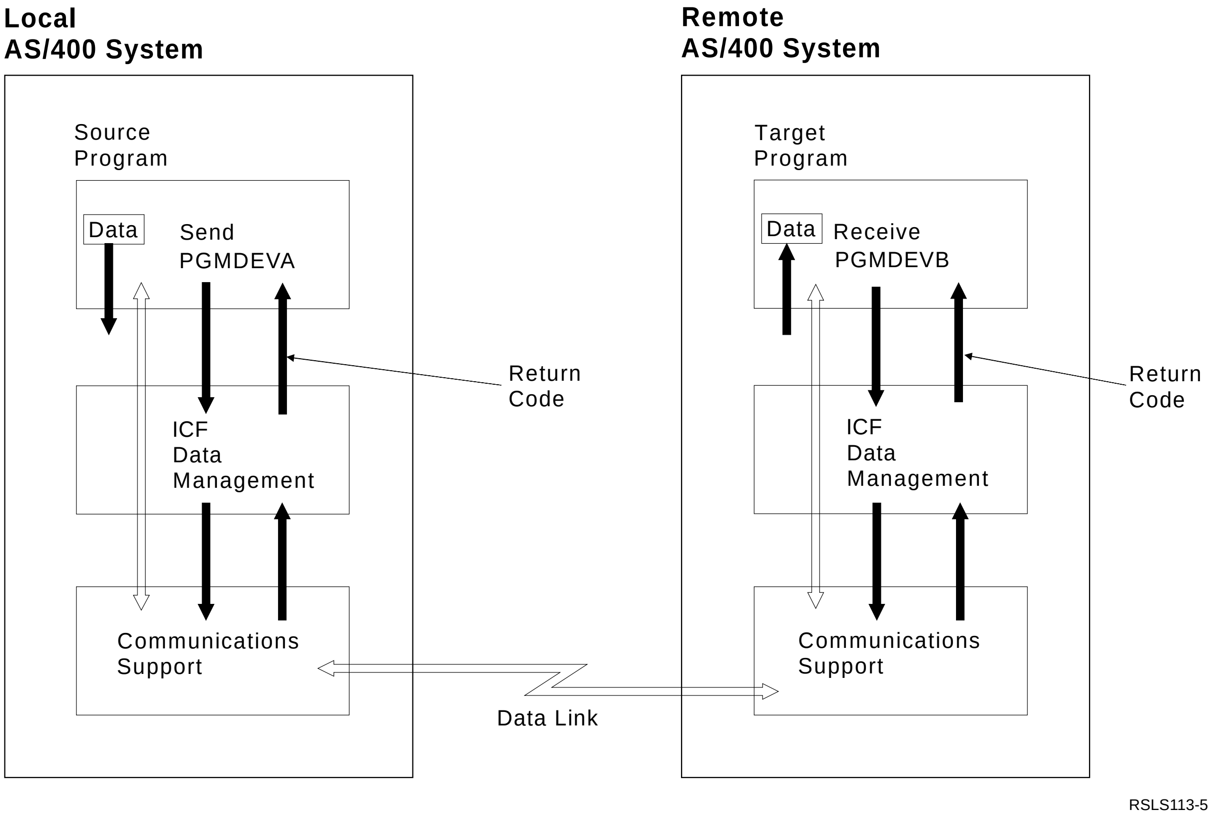


Figure 3-9. Data Sent by a Send Request

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 on page 3-11 and Figure 3-11 on page 3-12
- 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 on page 3-11 shows the source program issuing a detach function to end the communication transaction.

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.

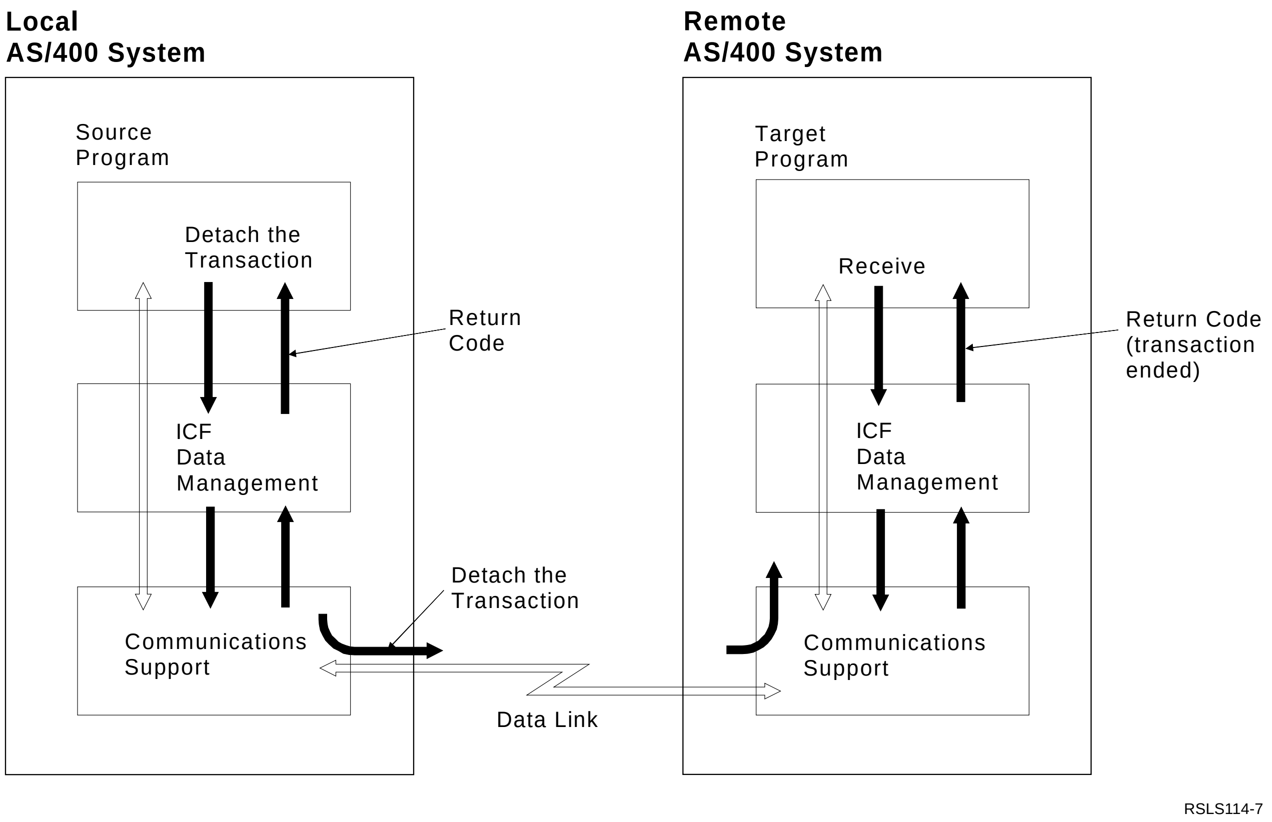


Figure 3-10. Ending a Communications Transaction: Detach Function

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 on page 3-12 shows the source program using the release operation and the target program using an end-of-session function to end the session.

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.

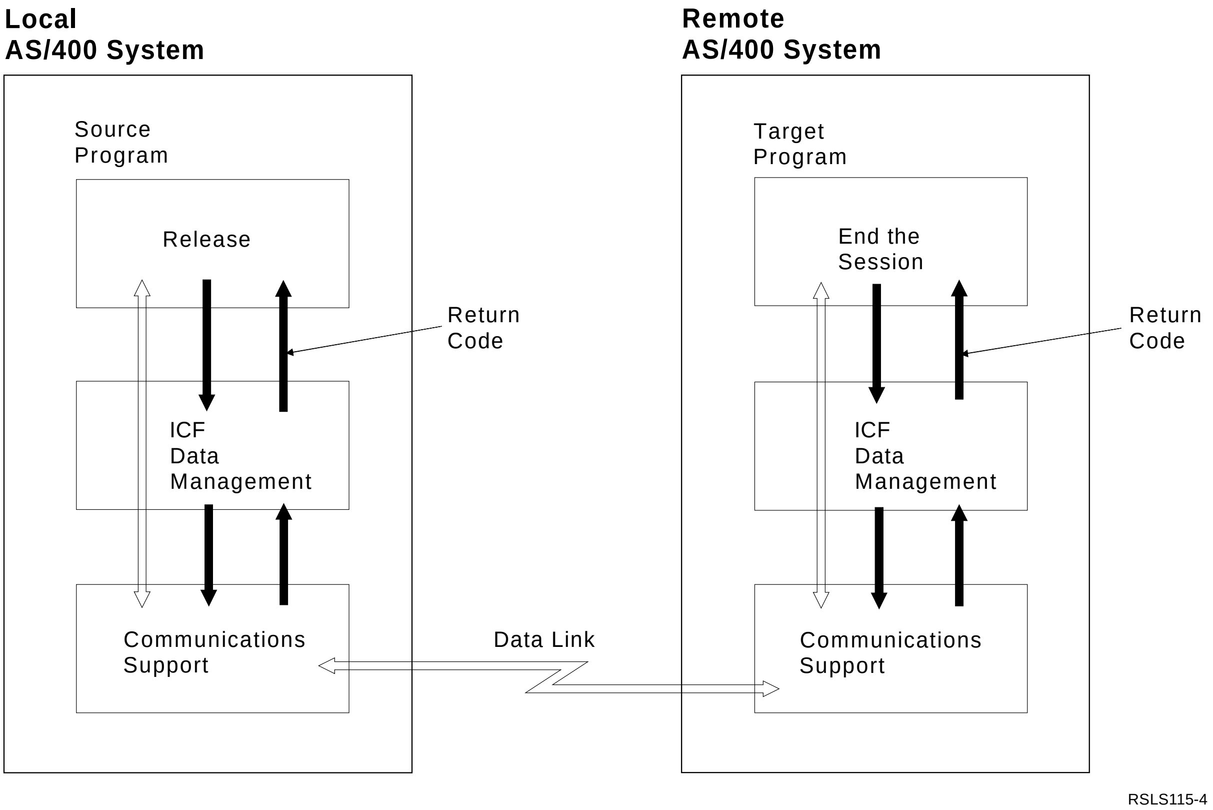


Figure 3-11. Ending a Session: Release Operation and End-of-Session Function

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 Vary Configuration (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 Management* book for information on the VRYCFG and ENDMOD commands.

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 on page 3-13 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.

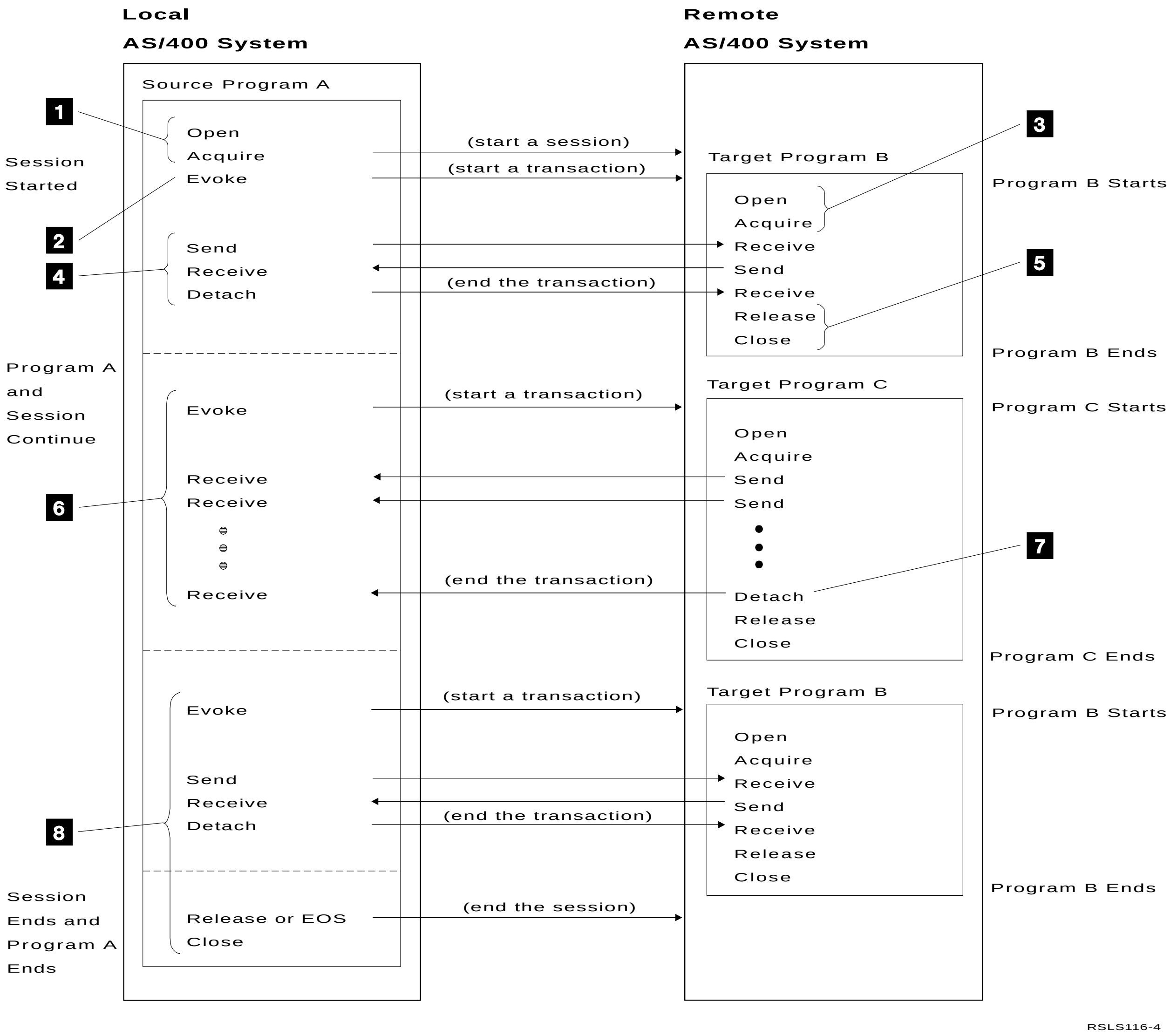


Figure 3-12. Starting and Ending Sessions and Transactions

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 on page 3-14 shows how a target program can start a session and a 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.

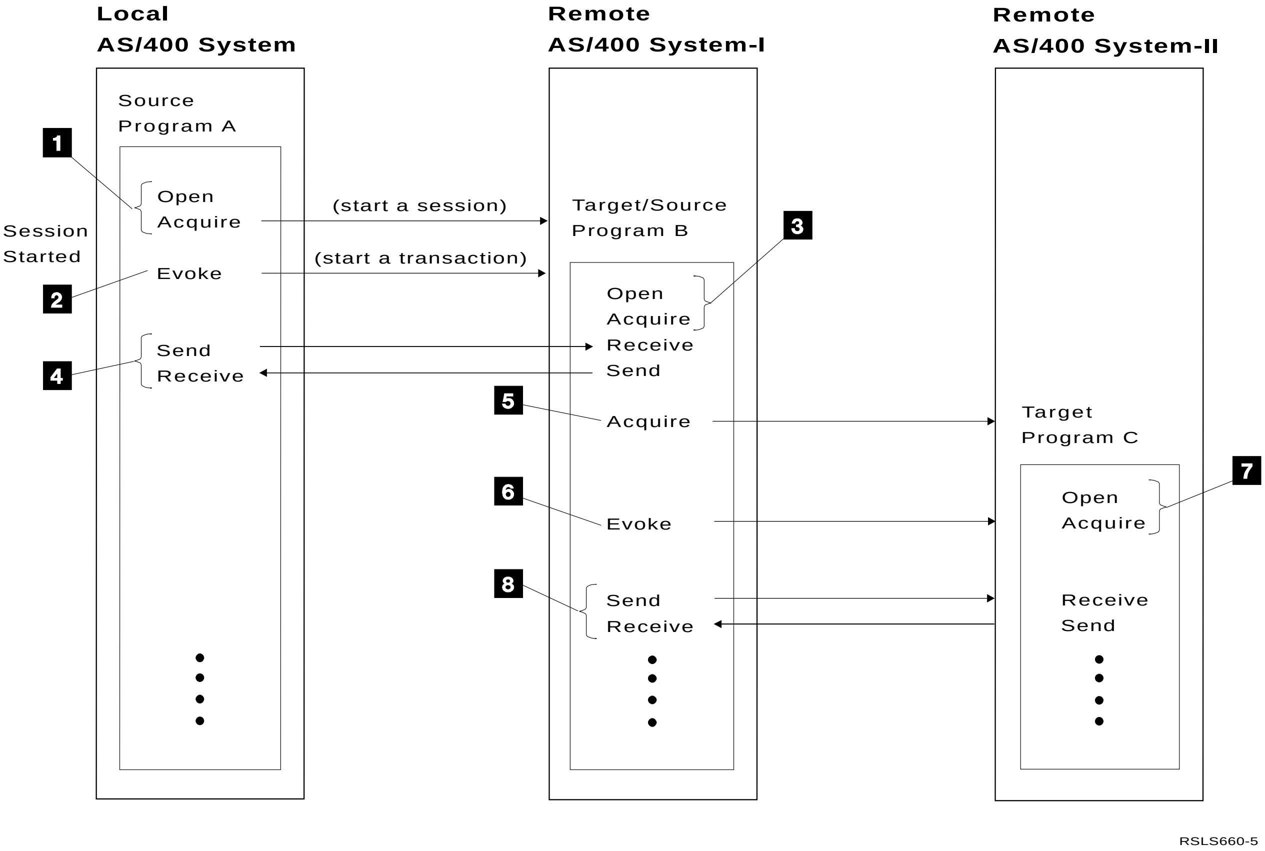


Figure 3-13. Remotely Started Program Starts a Session and Transaction

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 on page 3-15 shows the sequence of events your AS/400 system application program follows when it starts a session with the 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

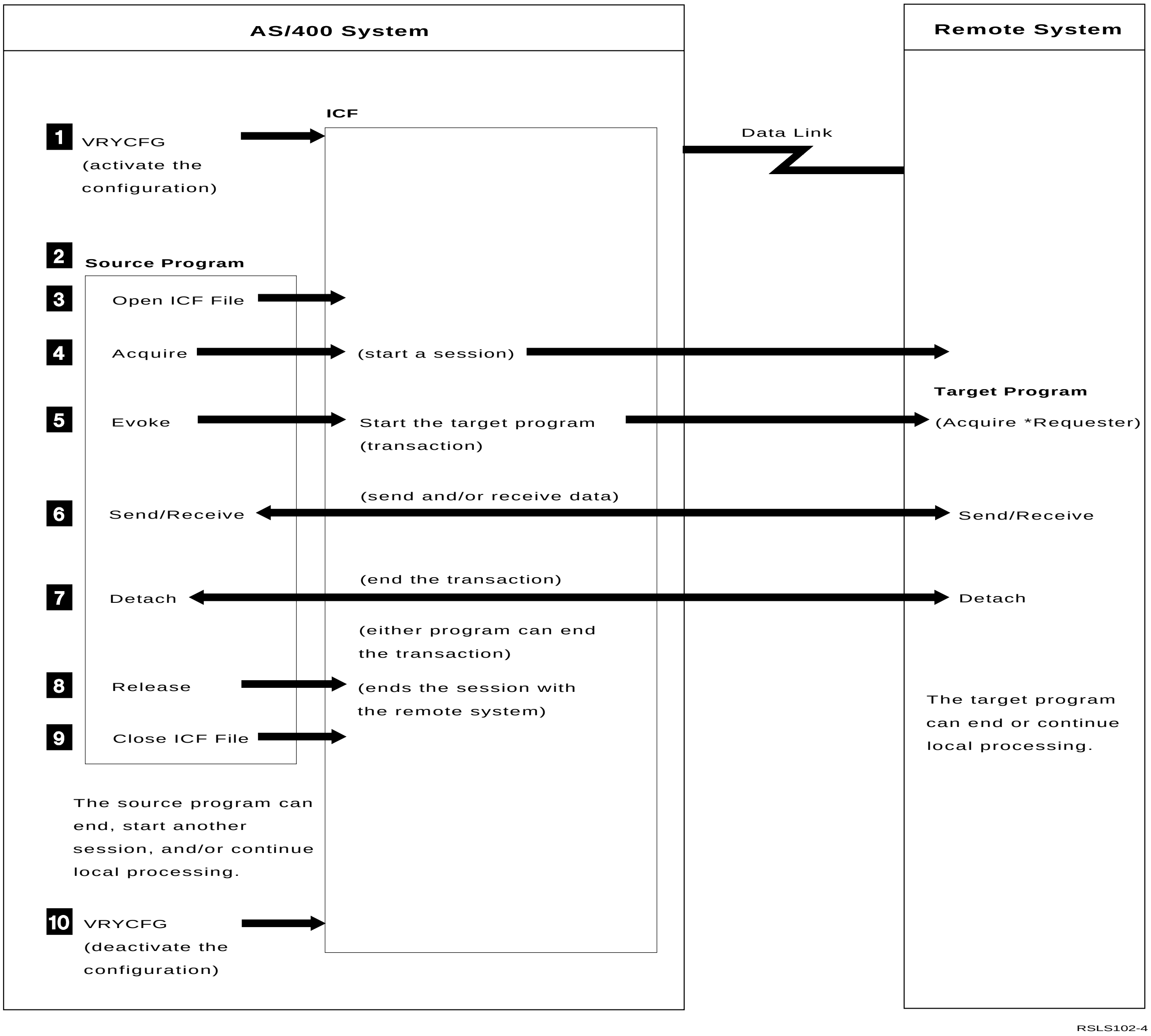

AS/400 System
Remote System
ICF
Data Link
1 VRYCFG (activate the configuration)
2 Source Program
3 Open ICF File
4 Acquire
(start a session)
Target Program
5 Evoke
Start the target program (transaction)
(Acquire *Requester)
(send and/or receive data)
6 Send/Receive
Send/Receive
(end the transaction)
7 Detach
Detach
(either program can end the transaction)
8 Release
(ends the session with the remote system)
The target program can end or continue local processing.
9 Close ICF File
The source program can end, start another session, and/or continue local processing.
10 VRYCFG (deactivate the configuration)
RSLS102-4

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

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.

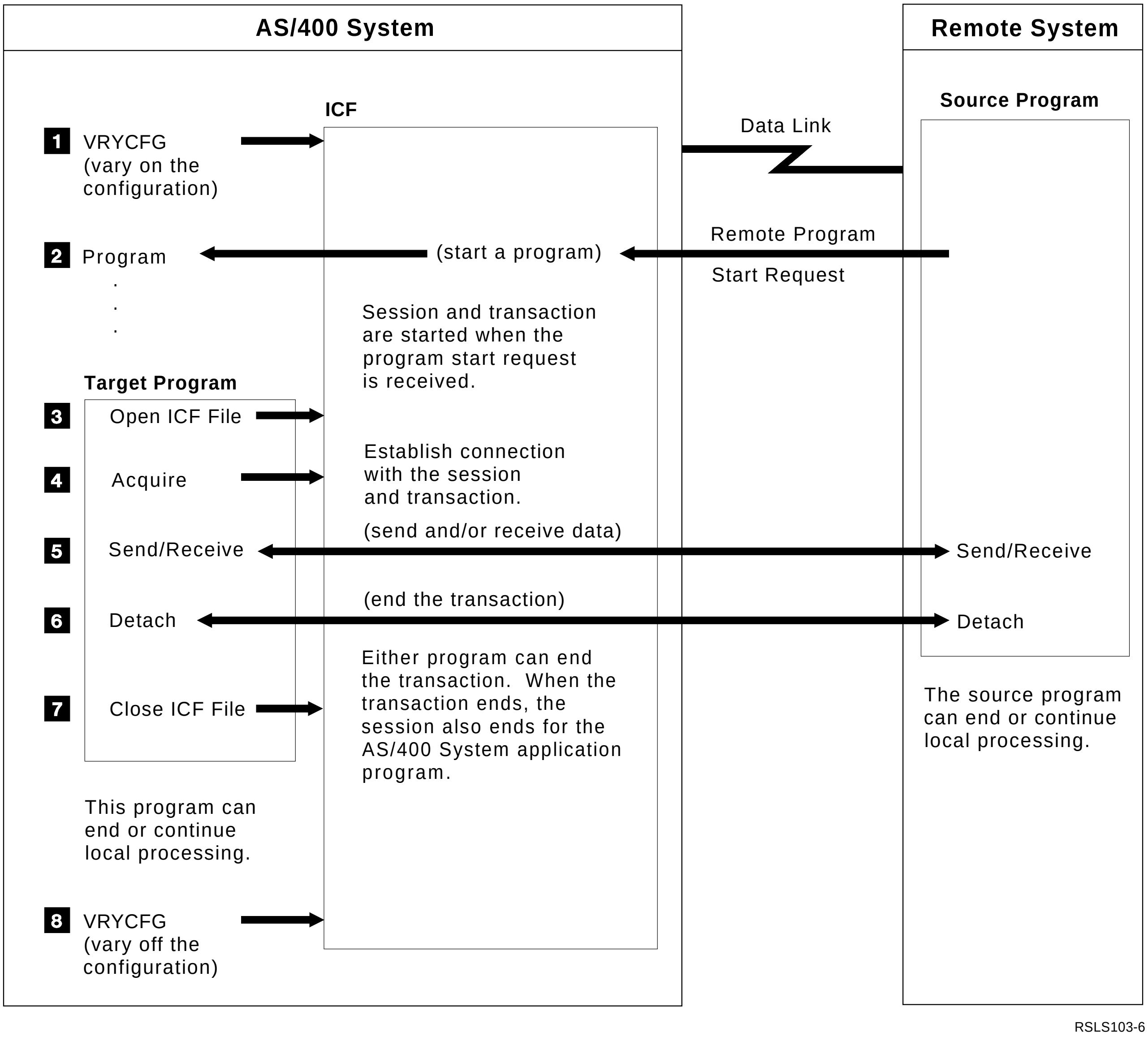

AS/400 System
Remote System
ICF
Source Program
Data Link
1 VRYCFG (vary on the configuration)
2 Program . . .
(start a program)
Remote Program Start Request
Session and transaction are started when the program start request is received.
Target Program
3 Open ICF File
4 Acquire
Establish connection with the session and transaction.
(send and/or receive data)
5 Send/Receive
Send/Receive
(end the transaction)
6 Detach
Detach
Either program can end the transaction. When the transaction ends, the session also ends for the AS/400 System application program.
7 Close ICF File
The source program can end or continue local processing.
This program can end or continue local processing.
8 VRYCFG (vary off the configuration)
RSLS103-6

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

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.

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 communications entries using the ADDCMNE command. Refer to the *Communications Management* book 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 advanced program-to-program communications (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.

Note: References to APPC apply to APPC over TCP/IP communications also.

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.

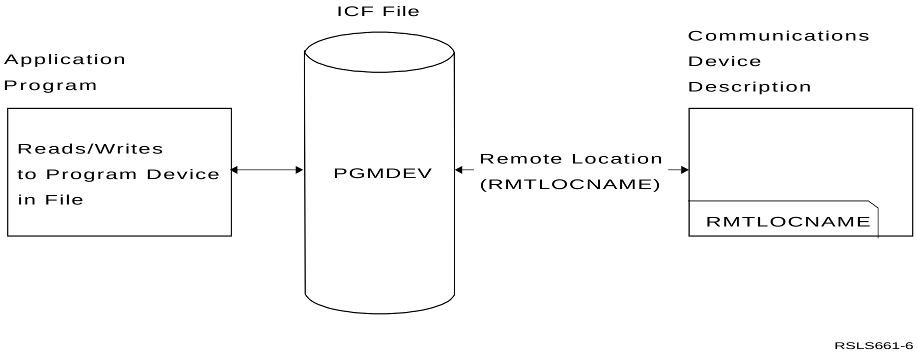

ICF File
Application Program
Reads/Writes to Program Device in File
PGMDEV
Remote Location (RMTLOCNAME)
Communications Device Description
RMTLOCNAME
RSLS661-6

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 book 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.

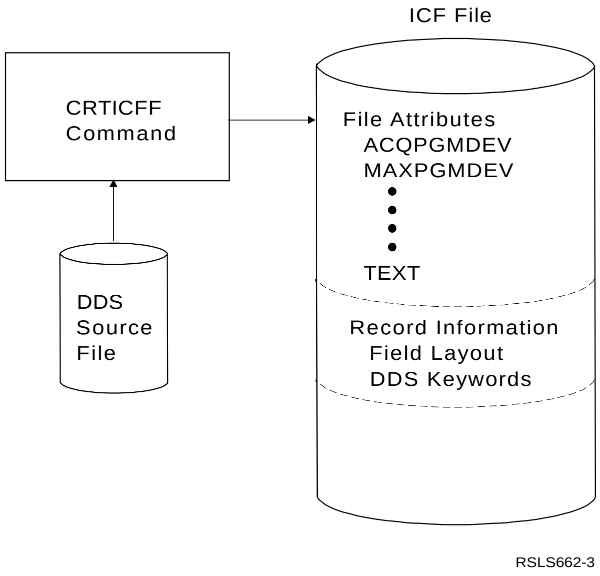


Figure 4-2. Creating an ICF File

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* book 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* book.

File Attributes

Figure 4-3 identifies the attributes used with an ICF file. These attributes are specified by the parameters on the CRTICFF command.

Figure 4-3. 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
FLAG	Minimum security level of error messages to be listed
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
DTAQ	Name and library of the data queue on which entries are placed
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
REPLACE	Specifies whether an existing ICF file is replaced
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 4-7 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 book 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 4-5 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.

Using a Data Queue: If you want your program to wait for an ICF file and a data queue at the same time, use the data queue (DTAQ) parameter on the CRTICFF command. The program can also wait for a display file if the same data queue is specified on the CRTDSPF, CHGDSPF, or OVRDSPF commands. Refer to “Waiting for a Display File, an ICF File, and a Data Queue” on page 5-15 for more information.

Determining Other CRTICFF Command Parameter Values: Refer to the CRTICFF command in the *CL Reference* book to determine the appropriate values for the SRCFILE, SRCMBR, OPTION, GENLVL, FLAG, LVLCHK, SHARE, AUT, REPLACE, 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 Figure 4-4 for changing file-level attributes values specified on the CRTICFF command.

Figure 4-4 (Page 1 of 2). File Attributes for Changing an ICF File

Parameter	Description
ACQPGMDEV	Program device to be acquired when the file is opened

Figure 4-4 (Page 2 of 2). File Attributes for Changing an ICF File

Parameter	Description
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
DTAQ	Name and library of the data queue on which entries are placed
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 Figure 4-5 for overriding file-level attribute values specified on either the CRTICFF or CHGICFF command.

Figure 4-5. 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
DTAQ	Name and library of the data queue on which entries are placed
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

Override commands can be scoped to the job level, the activation group level (the default), or the call level. Overrides scoped to the job level remain in effect until they are deleted, replaced, or until the job in which they are specified ends. Overrides scoped to the activation group level remain in effect until they are deleted, replaced, or until the activation group is deleted. Overrides scoped to the call level remain in effect until they are deleted, replaced, or until the program in which they were issued ends.

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

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

In the example, the OVRICFF applies to all the other programs, because overrides that are scoped to the job level affect all programs in the job. 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 highest 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 CBLPGM or RPGPGM is called, the WAITRCD value in effect is 4 seconds.

Refer to the *Data Management* book 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.

The WAITRCD parameter has no meaning in the FORTRAN/400 language, as the FORTRAN/400 language does not allow the read-from-invited-program-device operation.

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.

Note: The FORTRAN/400 language does not support program device names. To establish an ICF session using the FORTRAN/400 language, you must use the ACQPGMDEV parameter on the CRTICFF, CHGICFF, or OVRICFF commands. You can have only one ICF session for each ICF file opened using the FORTRAN/400 language.

Defining Program Device Entries Permanently

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 4-6 on page 4-8 shows how to define a program device entry to an ICF file.

Note: Figure 4-6 on page 4-8 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 permanently or temporarily. A permanent definition is system-wide and affects all users of the file. A permanent definition adds the program device entry to the specified file. A temporary definition does not change an ICF file. The definition is only associated with the job in which the command is entered. Because temporary changes are not directed to a specific ICF file, all ICF files associated with the job or call level are affected.

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.

Defining Program Device Entries Temporarily

In addition to the file attributes and record formats similar to those in other device files, an ICF file also contains program device entries that provide the link between the application and each of the remote systems or devices with which your program communicates.

The following lists the CL commands that provide override functions for device entries:

DLTOVRDEVE
: Delete Override Device Entry: Deletes one or more program device overrides that were previously specified in a call level.

OVRICFDEVE
: Override with Intersystem Communications Program Function Device Entry: Used to temporarily add the program device entry and the remote location name to the ICF file or to override a program device entry with the specified remote location name and attributes for an ICF file.

A program device entry has two functions:

- It associates a program device name with a remote location.
- It establishes a set of program communications-type dependent attributes.

Multiple program device entries can be defined. Each program device entry must have a unique program device name. The maximum number of entries is determined by the MAXPGMDEV parameter specified at file creation.

Program device entries may be defined by the ADDICFDEVE command or the OVRICFDEVE command. The ADDICFDEVE command makes a permanent addition to the file, and the OVRICFDEVE command makes a temporary change to the program device information. It is not neces-

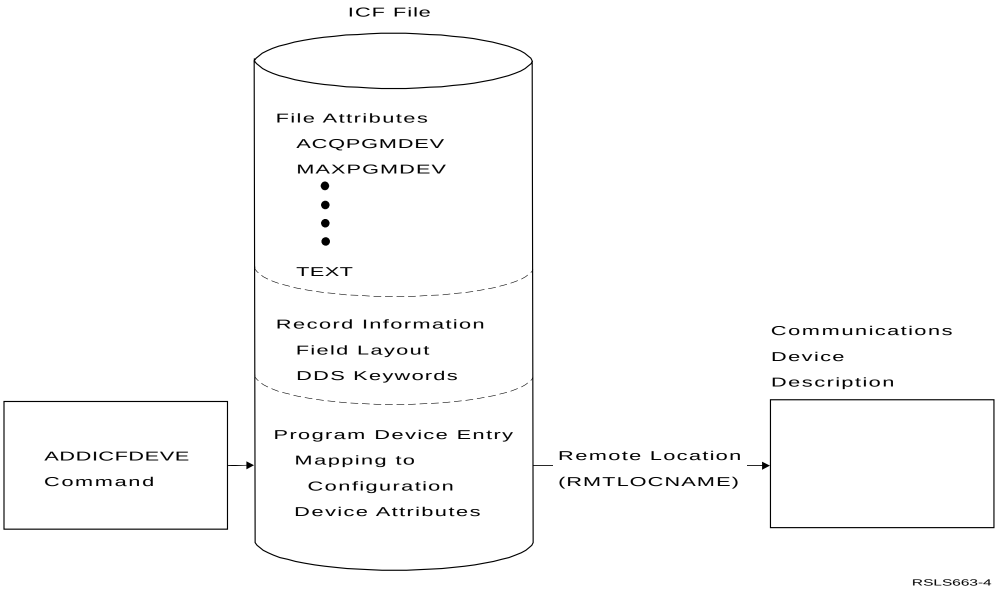


Figure 4-6. Defining a Program Device Entry to an ICF File

sary to add a program device entry before overriding it. Several ADDICFDEVE commands may be used to add multiple program devices to the same file. Several OVRICFDEVE commands may be used to change different device entries.

Overriding Remote Location Name: The device entry override may be used to temporarily define or change the remote location name associated with the program device entry.

The following example demonstrates the use of the OVRICFDEVE command to override the remote location name:

```
OVRICFDEVE PGMDEV(PGMDEVA) RMTLOCNAME(CHICAGO)
CALL RPGPGM
```

In this example, when RPGPGM specifies PGMDEVA, remote location CHICAGO is used.

Overriding Session Attributes: The device entry override may also be used to temporarily change the characteristics of the communications session that is established when the program device is acquired.

Although some of the session attributes have system-level defaults, the default for the majority of these attributes is information supplied during communications configuration.

Session attributes are identified as parameters on the ADDICFDEVE or OVRICFDEVE command. Parameters not specified on either command take on the appropriate system default or specified configuration value. If the same parameter is specified on both the ADDICFDEVE and OVRICFDEVE commands, the value specified on OVRICFDEVE overrides the value declared on the ADDICFDEVE command.

The following example demonstrates the use of the OVRICFDEVE command to override the format selection processing attribute:

```
OVRICFDEVE PGMDEV(PGMDEVA) FMTSLT(*PGM)
```

In this example, format selection is changed to *PGM. This overrides what was previously defined in the program device entry. Refer to the appropriate communications programming book for more information on the use of the session attributes. Refer to the *CL Reference* book for more information on the format and allowable values of the parameters on the OVRICFDEVE command.

Overriding Remote Location Name and Session Attributes: This form of the override device entry is a combination of the previous two forms. With this form of override, you can override the remote location that is used by a program, and you can also override the session attributes.

Applying OVRICFDEVE Command: Device entry overrides follow most of the same rules as file overrides. They are effective from the time they are specified until they are replaced or deleted or until the program in which they were specified ends. Any program device entry overrides that are in effect at the time the device is acquired are applied.

The OVRICFDEVE command can be used to initialize an environment or change the environment while running.

In the following example, the OVRICFDEVE commands are initializing an environment:

```
Override 1   OVRICFDEVE PGMDEV(PGMDEV1) +
               RMTLOCNAME(BOSTON) . . .
Override 2   OVRICFDEVE PGMDEV(PGMDEV2) +
               RMTLOCNAME(ROCHMN) . . .
             CALL PGM(A)
             CALL PGM(B)
             .
             .
             .
             CALL PGM(X)
```

When the program uses any ICF file and acquires the program device named PGMDEV1, then the remote location named BOSTON and attributes from override 1 are used when establishing the communication session.

When the program uses an ICF file and acquires the program device named PGMDEV2, then the remote location named ROCHMN and attributes from override 2 are used when establishing the communication session.

In the following example, the OVRICFDEVE commands are used to change the running environment:

```
Override 1   OVRICFDEVE PGMDEVE(PGMDEV1) +
               RMTLOCNAME(BOSTON) . . .
             CALL PGM(A)
Override 2   OVRICFDEVE PGMDEVE(PGMDEV2) +
               RMTLOCNAME(ROCHMN) . . .
             CALL PGM(A)
```

The first time program A is called, an ICF file is opened and the program device named PGMDEV1 acquired. The remote location named BOSTON and attributes from override 1 are used when establishing the communication session.

The second time program A is called, an ICF file is opened and the program device named PGMDEV2 is acquired. The remote location named ROCHMN and attributes from override 2 are used when establishing the communication session.

Applying OVRICFDEVE from Multiple Call Levels: When you have more than one override for the same program device at several call levels (nested calls), the order in which the overrides are applied to the program device is from the highest call level to the lowest call level. Any job level overrides are applied last.

To prevent overrides at lower call levels from being applied, see “Applying OVRICFDEVE with SECURE.”

In the following example, override 2 is in the highest call level and override 1 is in the lowest call level.

```
Override 1   OVRICFDEVE PGMDEV(PGMDEV1) +
               FMTSLT(*PGM) BATCH(*NO)
             CALL PGM(A)
               Program A
Override 2     OVRICFDEVE PGMDEV(PGMDEV1) +
                 FMTSLT(*RECID) APPID(PAYROLL)
               CALL PGM(X)
```

When program X acquires program device PGMDEV1, the following attributes are used:

FMTSLT(*PGM)	From Override 1
BATCH(*NO)	From Override 1
APPID(PAYROLL)	From Override 2

The attribute of FMTSLT(RECID) specified in override 2 is not used because it was overridden by FMTSLT(*PGM) specified in override 1. Override 1 overrides override 2. If there is a third override for program device PGMDEV1 embedded in program X, it is overridden by override 2 and then override 1.

A similar situation exists when you change the remote location to be used with the program device and you also change some of the attributes of the program device. For example:

```
Override 1   OVRICFDEVE PGMDEV(PGMDEV1) +
               RMTLOCNAME(NYCAPPC)
             CALL PGM(A)
               Program A
Override 2     OVRICFDEVE PGMDEV(PGMDEV1) +
                 RMTLOCNAME(MPLSAPPC) +
                 CNVTYPE(*USER)
               CALL PGM(X)
```

When program X is ready to acquire PGMDEV1, it acquires remote location NYCAPPC instead of MPLSAPPC (because override 1 overrides override 2 remote location). Also, the conversation type is *USER (because of override 2).

Applying OVRICFDEVE with SECURE: On occasion, you may want to protect program devices used by a program from overrides at lower call levels.

You can prevent additional program device overrides by coding the SECURE(*YES) parameter on a program device override command for each program device needing protection. This protects you from overrides at lower call levels.

The following shows an example of a protected program device:

```
Override 1  OVRICFDEVE PGMDEV(PGMDEV1) +
              RMTLOCNAME(BOSTON)
Override 2  OVRICFDEVE PGMDEV(PGMDEV4) +
              RMTLOCNAME(ROCHMN)
            CALL PGM(A)
              Program A
Override 3    OVRICFDEVE PGMDEV(PGMDEV5) +
                RMTLOCNAME(NYC)
              CALL PGM(B)
                Program B
Override 4      OVRICFDEVE PGMDEV(PGMDEV1) +
                  RMTLOCNAME(MPLS) SECURE(*YES)
                CALL PGM(X)
```

When program X acquires program device PGMDEV1 for an ICF file, the remote location MPLS and attributes from override 4 are used. Because override 4 specifies SECURE(*YES), override 1 is not applied.

Deleting Device Entry Overrides: When a program returns from a call level containing program device entry overrides, the overrides are deleted, just as any file overrides are deleted. When control is transferred to another program (TFRCTL command) so that the program is running at the same call level, the overrides are not deleted. If you want to delete an override before the run is completed, you can use the Delete Override Device Entry (DLTOVRDEVE) command. This command only deletes overrides in the call level in which the command is entered. A DLTOVRDEVE command does not delete the effects of an ADDICFDEVE command. To remove an ADDICFDEVE command, you must use the Remove Intersystem Communications Function Program Device Entry (RMVICFDEVE) command. To identify an override, use the program device name specified on the PGMDEV parameter of the override. You can delete all overrides at this call level by specifying value *ALL for the PGMDEV parameter. For example:

```
Override 1          OVRICFDEVE PGMDEV(PGMDEV1) +
                      RMTLOCNAME(BOSTON)
Override 2          OVRICFDEVE PGMDEV(PGMDEV4) +
                      RMTLOCNAME(ROCHMN)
Override 3          OVRICFDEVE PGMDEV(PGMDEV5) +
                      RMTLOCNAME(NYC)
Delete Override 1   DLTOVRDEVE PGMDEV(PGMDEV1)
Delete Override 2   DLTOVRDEVE PGMDEV(*ALL)
```

Delete override 1 causes override 1 to be deleted. Delete override 2 causes the remaining overrides (overrides 2 and 3) to be deleted.

Displaying Device Entry Overrides: Device entry overrides are not displayed by the Display Override (DSPOVR) command. There is no corresponding command to display device entry overrides.

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 Figure 4-7 on the ADDICFDEVE, CHGICFDEVE, and OVRICFDEVE commands.

Note: References to APPC apply to APPC over TCP/IP communications also.

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.

Figure 4-7. Mapping Parameters for All Communications Types

Parameter	Description	APPC	SNUF	BSCEL	Async	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	Communications device description	X	X			X		
LCLLOCNAME	Local location name	X						
MODE	Mode	X						
RMTNETID	Remote network ID	X						

The other 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.

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

Figure 4-8 on page 4-12 shows the relationship of the remote location to the device description. 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 4-8 on page 4-12, 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 Programming* book.

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 is being defined to process incoming program start requests, the special value of *REQUESTER must be used for the RMTLOCNAME parameter. The remaining parameters on the ADDICFDEVE, OVRICFDEVE, and CHGICFDEVE commands in Figure 4-7 on page 4-10 do not apply (except PGMDEV) and should not be specified.

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).

Note: If you rename a device after specifying the device name and remote location, you must update your ICF file accordingly.

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.

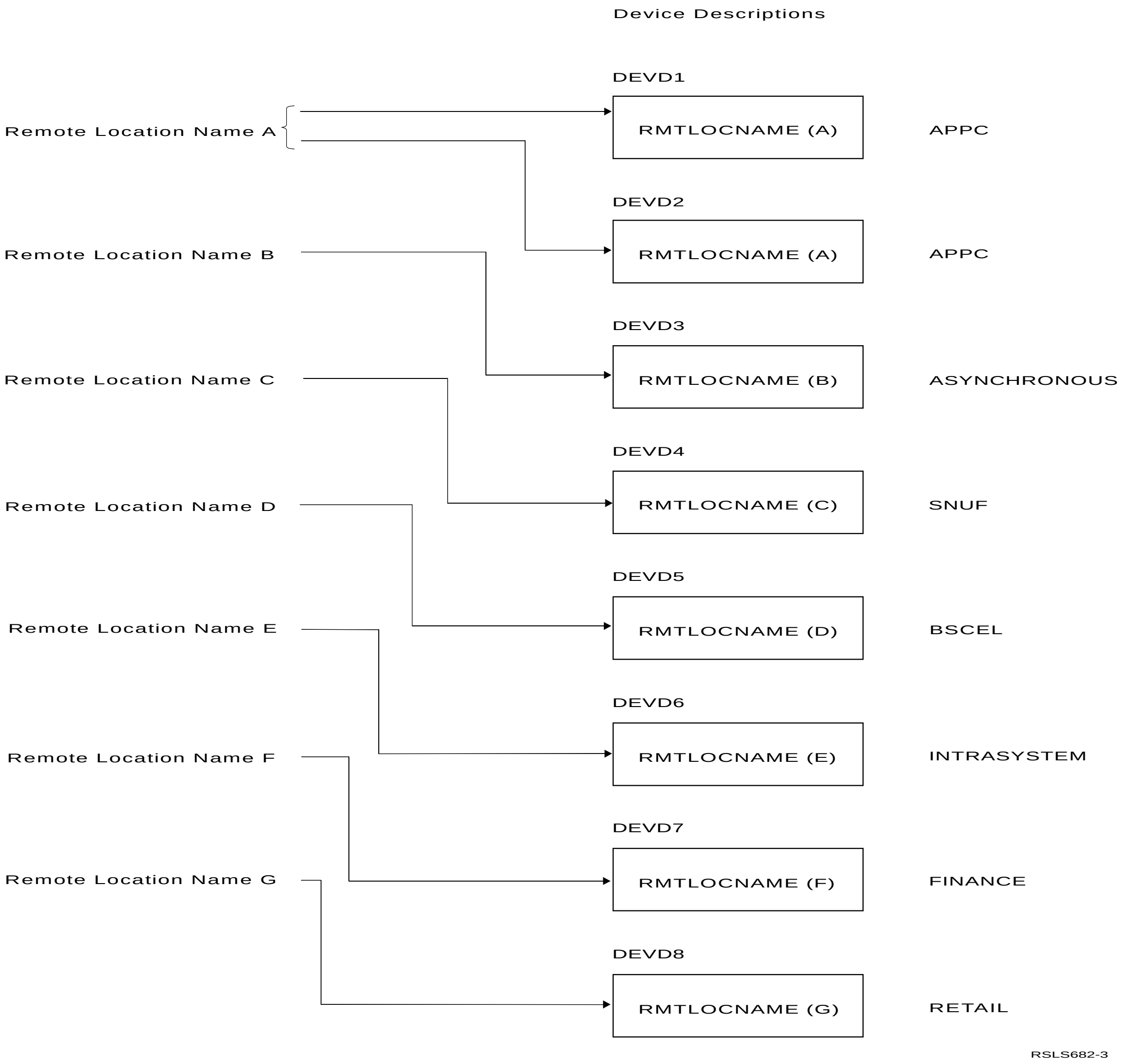


Figure 4-8. Relationship of Remote Location Name to Device Description

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

For more specific information, refer to *Communications Configuration* book and the appropriate communications programming book 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.

Figure 4-9 describes the communications-dependent attributes used with an ICF file and the communication types that support each attribute.

Note: References to APPC apply to APPC over TCP/IP communications also.

Figure 4-9 (Page 1 of 3). Communications-Type-Dependent Attributes

Parameter	Description	APPC	SNUF	BSCEL	Async	Intra-system	Finance	Retail
FMTSLT	Record format selection technique.	X	X	X	X	X	X	X

Figure 4-9 (Page 2 of 3). Communications-Type-Dependent Attributes

Parameter	Description	APPC	SNUF	BSCEL	Async	Intra-system	Finance	Retail
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					
BATCH	Specifies whether this session is used for batch activity with IMS/VS host system.		X			X		
HOST	Identifies type of host system with which to communicate.		X					
ENDSSNHOST	Specifies the command used to end a session with the host system.		X					
SPCHOSTAPP	Specifies whether the support should be customized for special host applications outside the CICS/VS or IMS/VS application layer.		X					
INZSELF	Specifies whether a formatted INIT-SELF is used in place of the unformatted logon normally sent to the host system.		X					
HDRPROC	Specifies whether received function management headers should be passed to the application.		X					
MSGPTC	Specifies whether message protection should be used.		X					
EMLDEV	Specifies whether the application is using 3270 data streams.		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
BLKLEN	Maximum block length to transmit and receive.		X	X				

Figure 4-9 (Page 3 of 3). Communications-Type-Dependent Attributes

Parameter	Description	APPC	SNUF	BSCEL	Async	Intra-system	Finance	Retail
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				
OVRFLWDTA	Specifies whether overflow data (data in excess of what can be contained in the input buffer) is discarded or retained.	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 “Applying OVRICFDEVE from Multiple Call Levels” on page 4-9.

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 and APPC over TCP/IP 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* book for information about how the SECURE parameter works. Refer to the *CL Reference* book 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 book for more information on the use of these parameters.

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

Intersystem Communications Function Command Summary

Figure 4-10 on page 4-16 shows the 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.)

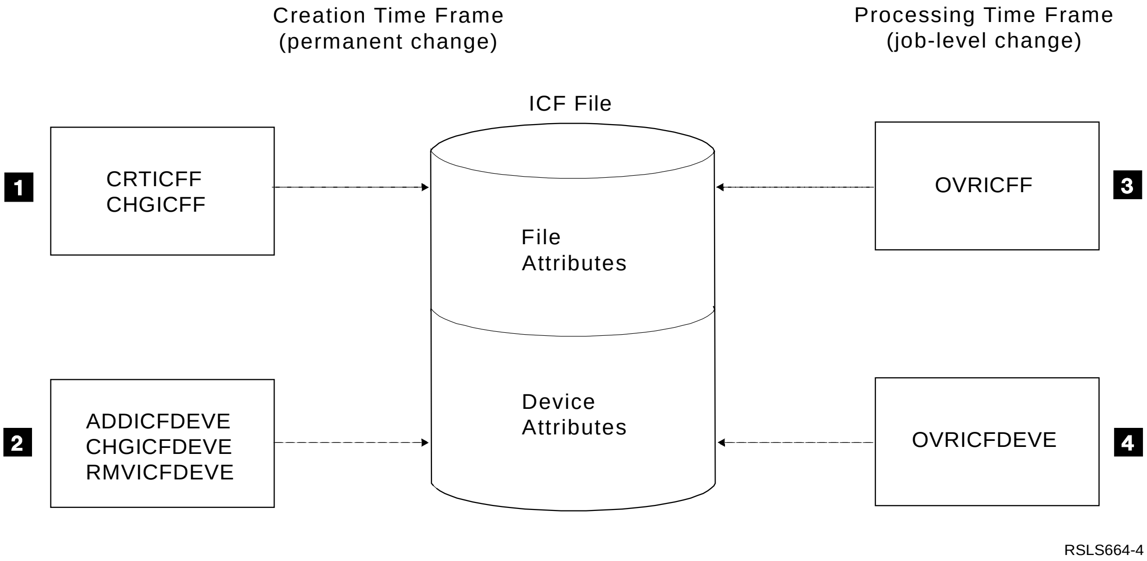


Figure 4-10. Relationship between ICF Commands

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 the ILE RPG programming language or as a TRANSACTION file in the ILE COBOL programming language. For the ILE C programming language, the type of file need not be specified.

The ILE C, ILE COBOL, and ILE RPG programming languages 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

The FORTRAN/400 language supports an interface that allows the application to perform the following operations:

- Open the file
- Read from the file (program device is implied)
- Write to the file (program device is implied)
- 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 book.

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.

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 4-3 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 5-17 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 book 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.

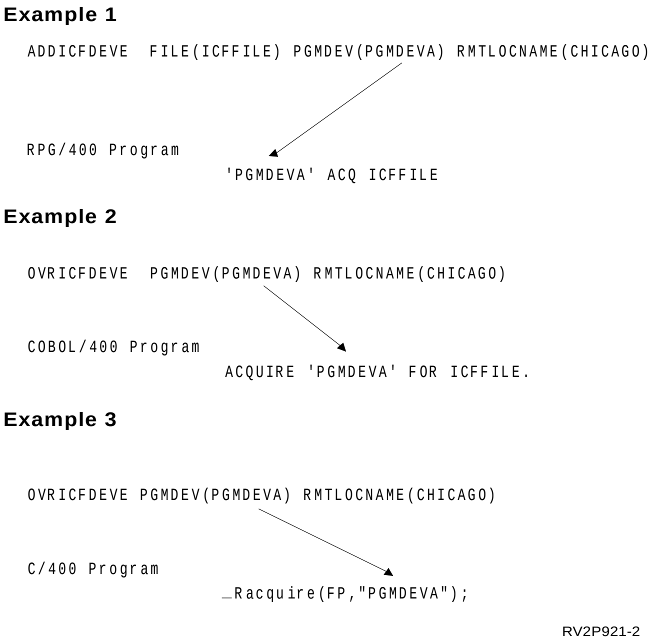
Only program devices defined to the file by use of the Add Intersystem Communications Function Device Entry (ADDICFDEVE) or the Override Intersystem Communications Function Device Entry (OVRICFDEVE) command can be acquired. See “Identifying the Devices Used with an Intersystem Communications Function File” on page 4-7 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 an ILE C, COBOL/400, and RPG/400 operation.



Example 1

```
ADDICFDEVE  FILE(ICFFILE) PGMDEV(PGMDEVA) RMTLOCNAME(CHICAGO)

RPG/400 Program
                'PGMDEVA' ACQ ICFFILE
```

Example 2

```
OVRICFDEVE  PGMDEV(PGMDEVA) RMTLOCNAME(CHICAGO)

COBOL/400 Program
                ACQUIRE 'PGMDEVA' FOR ICFFILE.
```

Example 3

```
OVRICFDEVE PGMDEV(PGMDEVA) RMTLOCNAME(CHICAGO)

C/400 Program
                _Racquire(FP,"PGMDEVA");
```

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

Note: For FORTRAN/400 programs, the acquire operation can be done implicitly by the system only when the ICF file is opened by specifying a program device name on the ACQPGMDEV parameter of the CRTICFF, CHGICFF, or OVRICFF command. The FORTRAN/400 language does not support program device names for ICF files; the program device is implied on read and write 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 book 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 based on the last ICF operation performed. The value for position 41 is an exception; this value is updated asynchronously by the system. 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 Figure 5-2.

Figure 5-2 (Page 1 of 3). 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 FINANC RETAIL	APPC or APPC over TCP/IP 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.

Figure 5-2 (Page 2 of 3). Attribute Information Fields

Position	Value	Meaning
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 2	Synchronization level is NONE. Synchronization level is CONFIRM. Synchronization level is COMMIT.
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.
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 104[1]	Reserved	Not applicable to communications.
105[1]		APPC conversation state.
	X'00'	Reset. No conversation exists.
	X'01'	Send. Program can send data.
	X'02'	Defer receive. Program enters receive state after a confirm, flush, or commit operation completes successfully.
	X'03'	Defer deallocate. Program enters deallocate state after a commit operation completes successfully.
	X'04'	Receive. Program can receive data.
	X'05'	Confirm. Program received a confirmation request.
	X'06'	Confirm send. Program received a confirmation request and send control.
	X'07'	Confirm deallocate. Program received a confirmation request and deallocate notification.
	X'08'	Commit. Program received a commit request.
	X'09'	Commit send. Program received a commit request and send control.
	X'0A'	Commit deallocate. Program received a commit request and deallocate notification.
	X'0B'	Deallocate. Program received a deallocate notification.
	X'0C'	Rollback required. Program must roll back changes to protected resources.
106 through 113[1]	Name	LU 6.2 conversation correlator
114 through 144[1]	Reserved	
145 through 146[1]	Binary	ISDN remote number length in bytes, including type and plan.
147 through 148[1]	00 01 02 03 04 06	ISDN unknown remote number type. ISDN international remote number type. ISDN national remote number type. ISDN network specific remote number type. ISDN subscriber remote number type. ISDN abbreviated remote number type.
149 through 150[1]	00 01 03 04 08 09	ISDN unknown remote number plan. ISDN/telephony remote number plan. ISDN data remote number plan. ISDN telex remote number plan. ISDN national standard remote number plan. ISDN private remote number plan.

Figure 5-2 (Page 3 of 3). Attribute Information Fields

Position	Value	Meaning
151 through 154[1]	Reserved	
155 through 190[1]	Character	ISDN remote number (blank-padded EBCDIC).
191 through 194	Reserved	
195 through 196[1]	Binary	ISDN remote subaddress length in bytes, including type.
197 through 198[1]	00 02	ISDN NSAP remote subaddress type. ISDN user defined remote subaddress type.
199 through 238[1]	Character	ISDN remote subaddress (EBCDIC representation of hexadecimal data padded on the right with zeros)
239[1]	Reserved	
240	0 1 2	Incoming ISDN call. Outgoing ISDN call. Not a switched ISDN connection.
241 through 242[1]	Binary	X.25 remote network address length in bytes.
243 through 274[1]	Character	X.25 remote network address (blank-padded EBCDIC).
275 through 278[1]	Reserved	
279 through 280[1]	Binary	X.25 remote address extension length in bytes, including type and extension.
281[1]	0 2	X.25 address assigned according to ISO 8348/AD2. Not X.25 ISO 8348/AD2 address type.
282 through 321[1]	Character	X.25 remote address extension (EBCDIC representation of hexadecimal data)
322 through 325[1]	Reserved	
326	0 1 2	Incoming X.25 switched virtual circuit (SVC). Outgoing X.25 SVC. Not X.25 SVC.
327 through 390[1]	Character	Name of program specified to be started as a result of the received program start request, even if a routing list caused a different program to be started.
391	Binary	Length of the protected logical unit of work identifier (LUWID). Must be from 0 to 26.
392	Binary	Length of the qualified LU name. Must be from 0 to 17.
393 through 409	Character	Network-qualified protected LU name in the following form: *netid.luname*. *netid* is the network identifier. *luname* is the logical unit name. This field may be blank.
410 through 415	Character	Protected LUWID instance number.
416 through 417	Binary	Protected LUWID sequence number. **Note:** The protected LUWID identifies the current logical unit of work for a protected conversation.
418	Binary	Length of the unprotected LUWID. Must be from 0 to 26.
419	Binary	Length of the qualified LU name. Must be from 0 to 17.
420 through 436	Character	Network-qualified unprotected LU name in the following form: *netid.luname*. *netid* is the network identifier. *luname* is the logical unit name. This field may be blank.
437 through 442	Character	Unprotected LUWID instance number.
443 through 444	Binary	Unprotected LUWID sequence number. **Note:** The unprotected LUWID identifies the current logical unit of work for conversations with a synchronization level of none or confirm.

[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, not including any indicators or program-to-system fields (P-data fields).
- **Blocked record count.** The number of records sent or received on an I/O operation. For ICF, the value is always 1.
- **Record length.** The record length of the last I/O operation based on the record format processed, including indicators and P-data.

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 unpredictable.

  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 following books:
  - *APPC Programming*
  - *Finance Communications Programming*
  - *SNA Upline Facility Programming*
  - *Retail Communications Programming*

  for more information on this field, depending on what communications type you are using.
- **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 Programming* book 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 Programming* book 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 book 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 ILE COBOL file status values. The ILE C programming language does not have file status values. These values are shown in a chart in the appropriate language chapter of this book.

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 5-3 on page 5-8 shows the use of the 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.

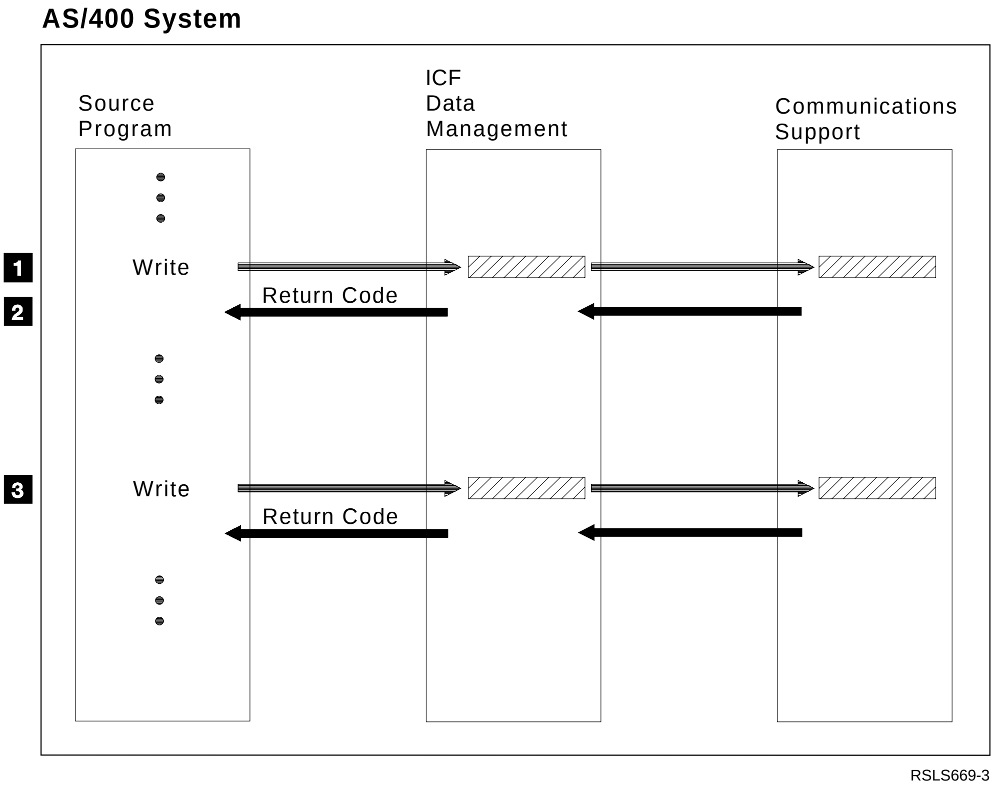


Figure 5-3. Using Write Operation When Sending Data

You can receive the response from an invited program device by doing an input operation. See “Reading from Invited Program Devices” on page 5-10 and “Reading from One Program Device” on page 5-14 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 5-14 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 operation 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 book for more information about accessing the I/O feedback area.

Summary of Format Selection Processing:

Figure 5-4 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.

Figure 5-4 (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.

Figure 5-4 (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
*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.
*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 complete record arrives for an invited program device.
- 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. The ILE C, ILE COBOL, and ILE RPG languages consider the ICF file to be a multiple device file, while the FORTRAN/400 language does not. See the appropriate language reference book for more information.

Figure 5-5 on page 5-11 shows how you can use the invite function and read-from-invited-program-devices operation to receive data from two different program devices.

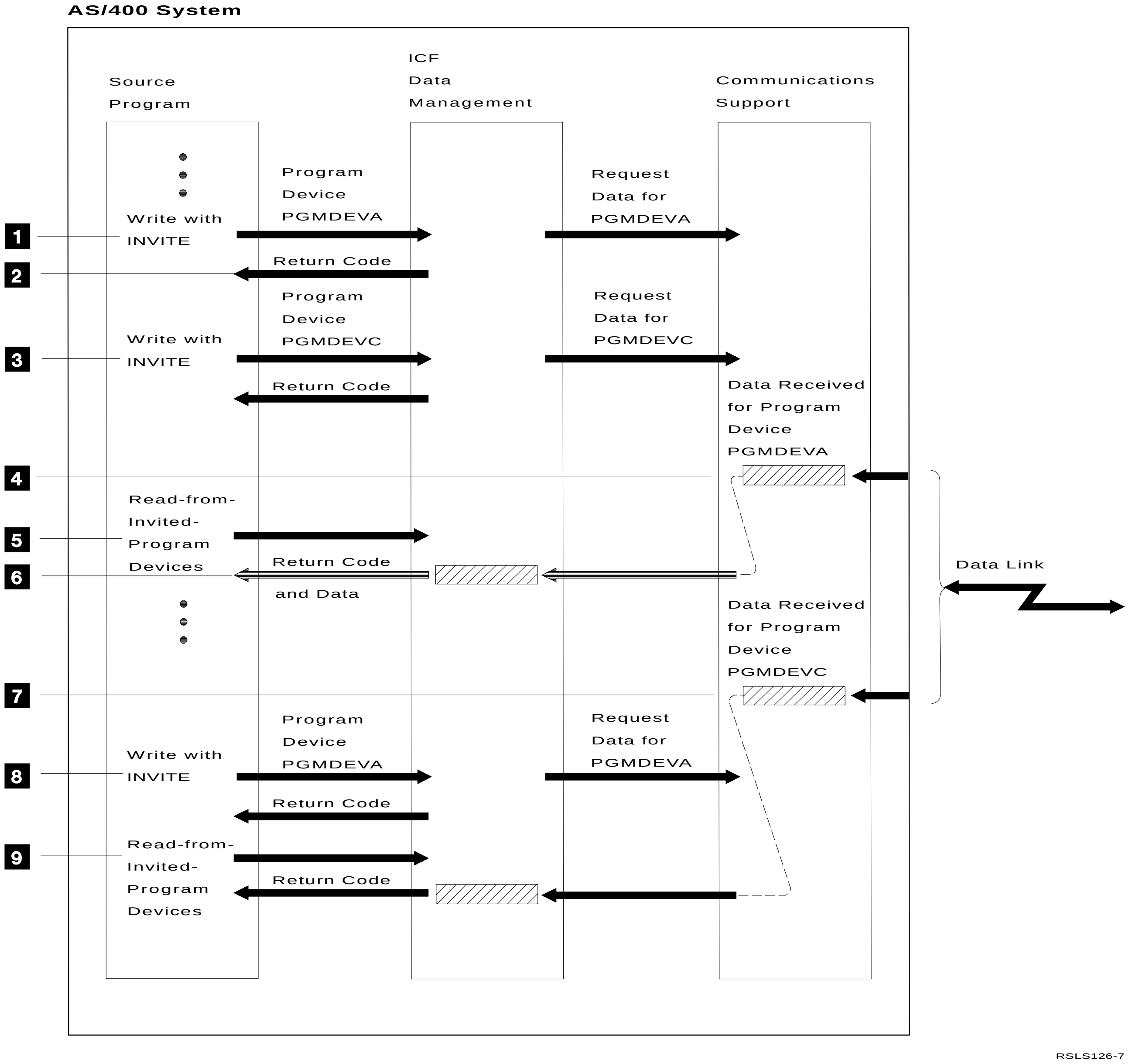


Figure 5-5. Using the Invite Function and Read-from-Invited-Program-Devices Operation to Receive Data

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 5-6 shows the relationship between the timer function and the read-from-invited-program-devices operation.

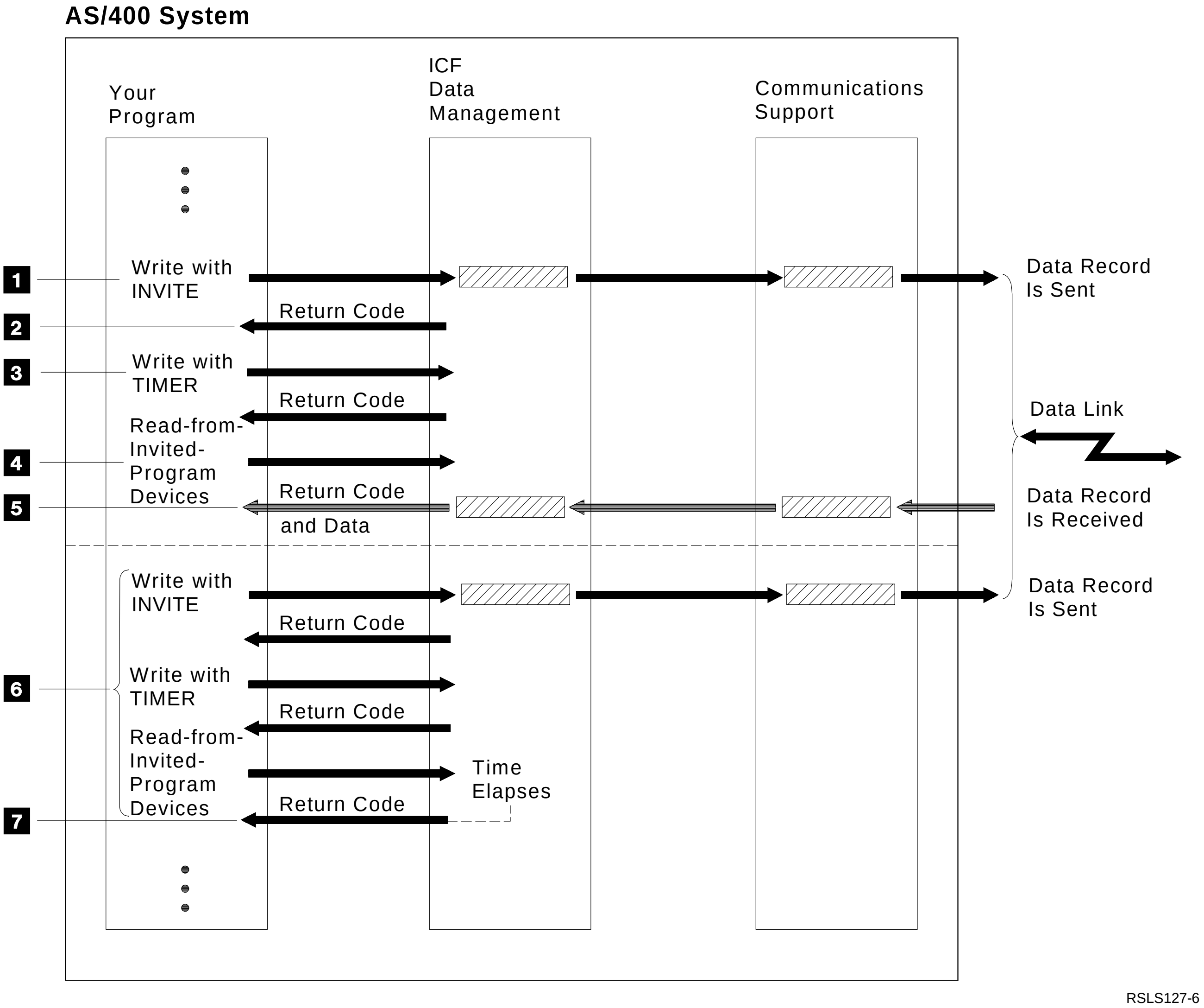


Figure 5-6. Relationship between Timer and Read-from-Invited-Program-Devices Operations

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.

The ILE COBOL programming language provides a means of calling the read-from-invited-program-devices operation as if WAITRCD(*IMMED) is specified. Refer to the *ILE COBOL/400 Reference* book for information about the NO DATA phrase of the READ statement. See “Determining the Wait-for-Record Value” on page 4-5 for information about specifying the WAITRCD parameter.

Note: WAITRCD has no meaning to the FORTRAN/400 language, since the read-from-invited-program-devices operation is not supported.

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 5-14 for information about how to determine which program device responded. See “Determining the Record Format Returned” on page 5-14 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 book 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 book 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.

Note: When a program device is invited, it is recommended that a read-from-invited-program-devices operation be performed rather than a read operation to receive data. Performance may be degraded if your program issues multiple read operations to invited program devices.

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 book for information about calling the read operation explicitly or implicitly and about which program device is used if the read operation is implicitly called.

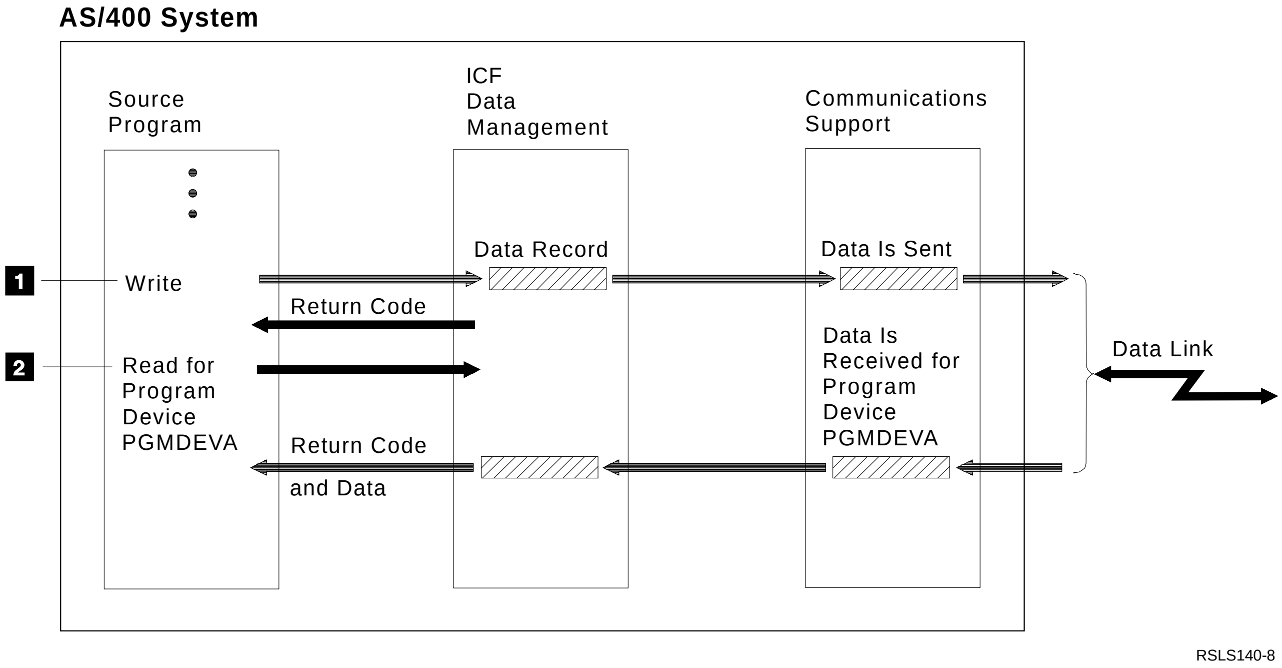


Figure 5-7. Using the Read Operation

Figure 5-7 shows how to use 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.

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

You can use data queues for a program that waits for data on a display file, an ICF file, and a data queue at the same time (in any combination). When you specify the DTAQ parameter for certain commands, you can indicate a data queue that will have entries placed on it when any of the following occurs:

- An enabled function key or the Enter key is pressed from an invited display device.
- Data becomes available from an invited ICF session.
- A user-defined entry is made to the data queue by a job running on the system.

The commands that allow you to indicate a data queue with the DTAQ parameter are:

- Create Display File (CRTDSPF)
- Change Display File (CHGDSPF)
- Override Display File (OVRDSPF)

- Create ICF File (CRTICFF)
- Change ICF File (CHGICFF)
- Override ICF File (OVRICFF)

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

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

The display file and ICF file entry that is put on the data queue is 80 characters in length and contains the field attributes described in Figure 5-8. Therefore, the data queue that is specified using the commands listed above must have a length of at least 80 characters.

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

Figure 5-8. Display File and ICF File Entry Field Attributes

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

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 5-9 on page 5-18 shows the relationship between the program, the ICF file, and the communications configuration on a local and remote AS/400 system.

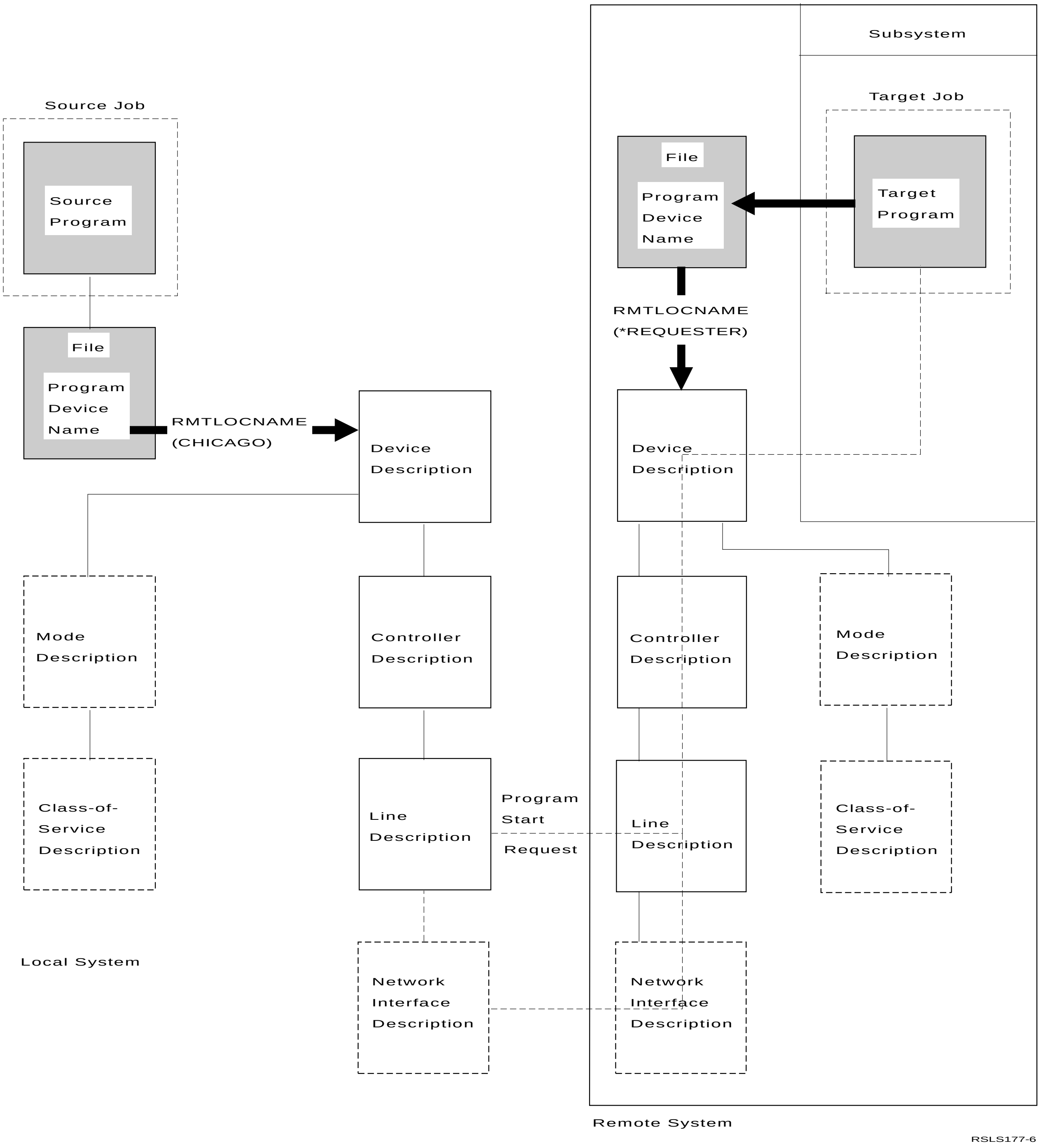


Figure 5-9. Relationship of Program, File, and Configuration

Notes:

1. The mode description and class-of-service description apply to APPC only.
2. Network interface description applies only to ISDN.

Local System

The source program is your ILE C, ILE COBOL, or ILE RPG application, which is 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 (Only one is allowed per ICF file for FORTRAN/400 applications.)
- 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 ILE C, ILE COBOL, or ILE RPG 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:

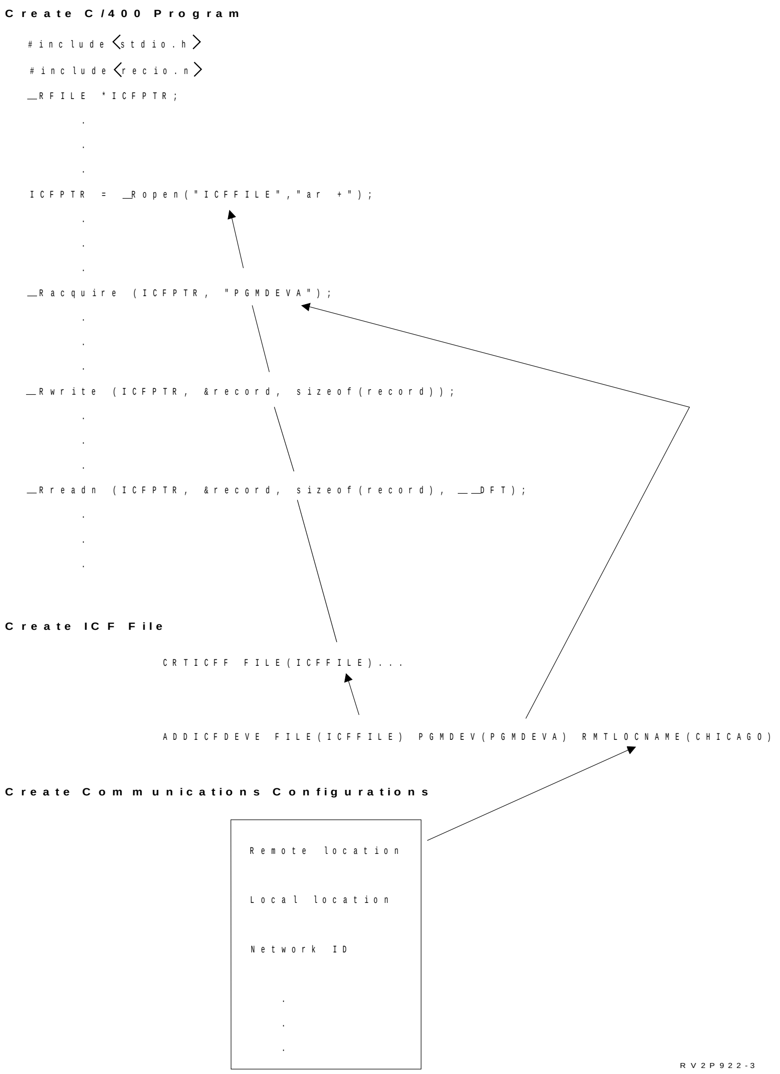
- Opens the file
- Acquires the requesting program device (This must be done implicitly on the open for FORTRAN/400 programs.)
- Reads and writes to the requesting program device for the file to receive and send data
- Releases the requesting program device (This must be done implicitly on the close for FORTRAN/400 programs.)
- 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 8-5 for more information on subsystems.

Figure 5-10 shows how the program, file, and configuration names are mapped to each other in ILE C programming language.



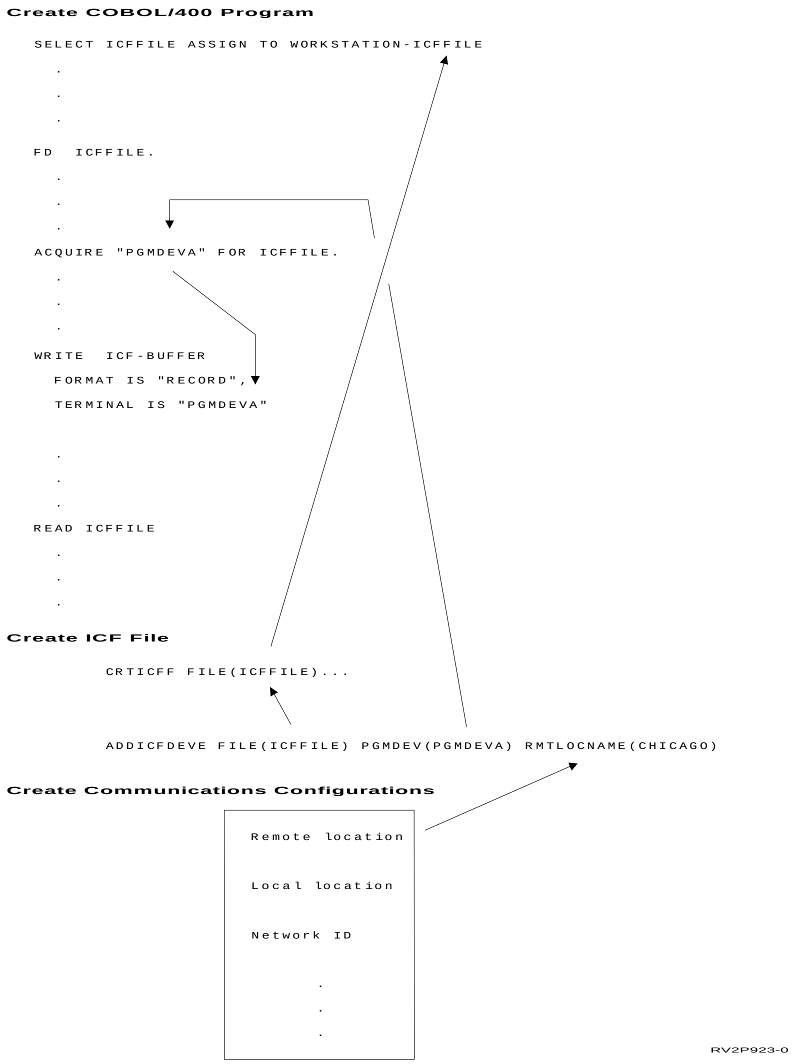
```
#include<stdio.h>
#include<recio.n>
_RFILE *ICFPTR;
      .
      .
      .
ICFPTR = _Ropen("ICFFILE","ar +");
      .
      .
      .
_Racquire (ICFPTR, "PGMDEVA");
      .
      .
      .
_Rwrite (ICFPTR, &record, sizeof(record));
      .
      .
      .
_Rreadn (ICFPTR, &record, sizeof(record), __DFT);
      .
      .
      .
```

```
CRTICFF FILE(ICFFILE)...

ADDICFDEVE FILE(ICFFILE) PGMDEV(PGMDEVA) RMTLOCNAME(CHICAGO)
```

Figure 5-10. ILE C 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 5-11 shows how the program, file, and configuration names are mapped to each other in the COBOL/400 programming language.



Create COBOL/400 Program

```
SELECT ICFFILE ASSIGN TO WORKSTATION-ICFFILE
   .
   .
   .
FD  ICFFILE.
   .
   .
   .
ACQUIRE "PGMDEVA" FOR ICFFILE.
   .
   .
   .
WRITE  ICF-BUFFER
  FORMAT IS "RECORD",
  TERMINAL IS "PGMDEVA"

   .
   .
   .
READ ICFFILE
   .
   .
   .
```

Create ICF File

```
CRTICFF FILE(ICFFILE)...

ADDICFDEVE FILE(ICFFILE) PGMDEV(PGMDEVA) RMTLOCNAME(CHICAGO)
```

Create Communications Configurations

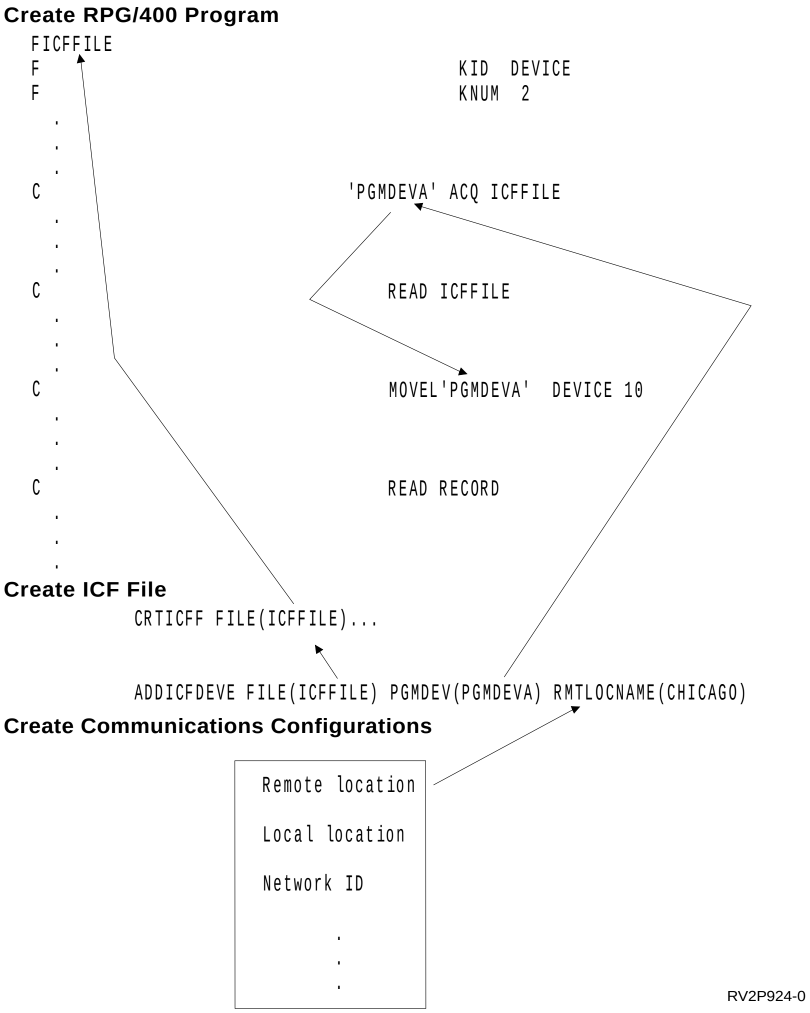
```
Remote location

Local location

Network ID

      .
      .
      .
```

Figure 5-11. 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 5-12 shows how the program, file and configuration names are mapped to each other in RPG/400 programming language.

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.



Create RPG/400 Program

```
FICFFILE
F                                     KID  DEVICE
F                                     KNUM  2
  .
  .
  .
C                          'PGMDEVA' ACQ ICFFILE
  .
  .
  .
C                              READ ICFFILE
  .
  .
  .
C                              MOVEL'PGMDEVA'  DEVICE 10
  .
  .
  .
C                              READ RECORD
  .
  .
  .
```

Create ICF File

```
CRTICFF FILE(ICFFILE)...

ADDICFDEVE FILE(ICFFILE) PGMDEV(PGMDEVA) RMTLOCNAME(CHICAGO)
```

Create Communications Configurations

```
Remote location

Local location

Network ID

      .
      .
      .
```

Figure 5-12. RPG/400 Program, File, and Configuration Mapping

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 the *DDS Reference* book. The *DDS Reference* book 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 6-19 on page 6-23 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. Figure 6-17 on page 6-21 and Figure 6-18 on page 6-22 summarize the support provided by each communications type. Refer to the appropriate communications programming book 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 book. These keywords are ALIAS, FLTPCN, INDARA, INDTXT, REF, REFFLD, and TEXT. Refer to the *DDS Reference* book 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. A program start request is sent to the remote system when your program issues the evoke function (unless, for APPC applications, the evoke is delayed by specifying the DFREVOKE keyword).

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

Evoke (EVOKE, DFREVOKE, 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 book 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. In general, it is best to specify the library separate from the program. If you specify the target program as a single literal, then it must be specified in the format required by the remote system or in the architected format. For example, if you are using APPC with another AS/400 system, the program name can be in the form *library/program* or *program.library*. 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 book 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.

DFREVOKE Keyword: You can use the DFREVOKE DDS keyword to delay sending a program start request until the output buffer is full of data or until the output buffer is flushed, using the FRCDTA or CONFIRM keyword, for example.

The DFREVOKE keyword is valid only for APPC and used only for specialized applications that must have data sent with the EVOKE keyword. See the *APPC Programming* book for more information.

SECURITY Keyword: 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.

SYNLVL Keyword: Use the SYNLVL DDS keyword to specify the level of synchronization supported on this transaction. It determines whether the programs support no synchronization, confirmation-level synchronization (using CONFIRM and RSPCONFIRM keywords), or commit-level synchronization. Commit-level synchronization is a two-phase commit protocol using the PRPCMT keyword and commit and rollback operations. The SYNLVL keyword is valid only in conjunction with the EVOKE keyword.

The format of the SYNLVL keyword is:

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

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 with SYNLVL (*NONE).

*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 6-4, “Receive-Confirm” on page 6-19, and “Respond-to-Confirm (RSPCONFIRM)” on page 6-12 for additional information on CONFIRM processing.

*COMMIT. Allows the programs to operate as described for the *CONFIRM value. Moreover, *COMMIT requires programs to use two-phase commit processing to protect their resources. Two-phase commit processing allows programs to synchronize updates to protected resources (such as databases). If necessary, updates can be rolled back, so that the resources remain synchronized. Refer to the descriptions of the PRPCMT, RCVROLLB, and RCVTKCMT keywords for more information on two-phase commit processing. The *Backup and Recovery* book has information about commitment control and the commit and rollback operations, which are an essential part of two-phase commit processing.

Evoke Illustration: Figure 6-1 shows how to start a target program on the remote system.

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.

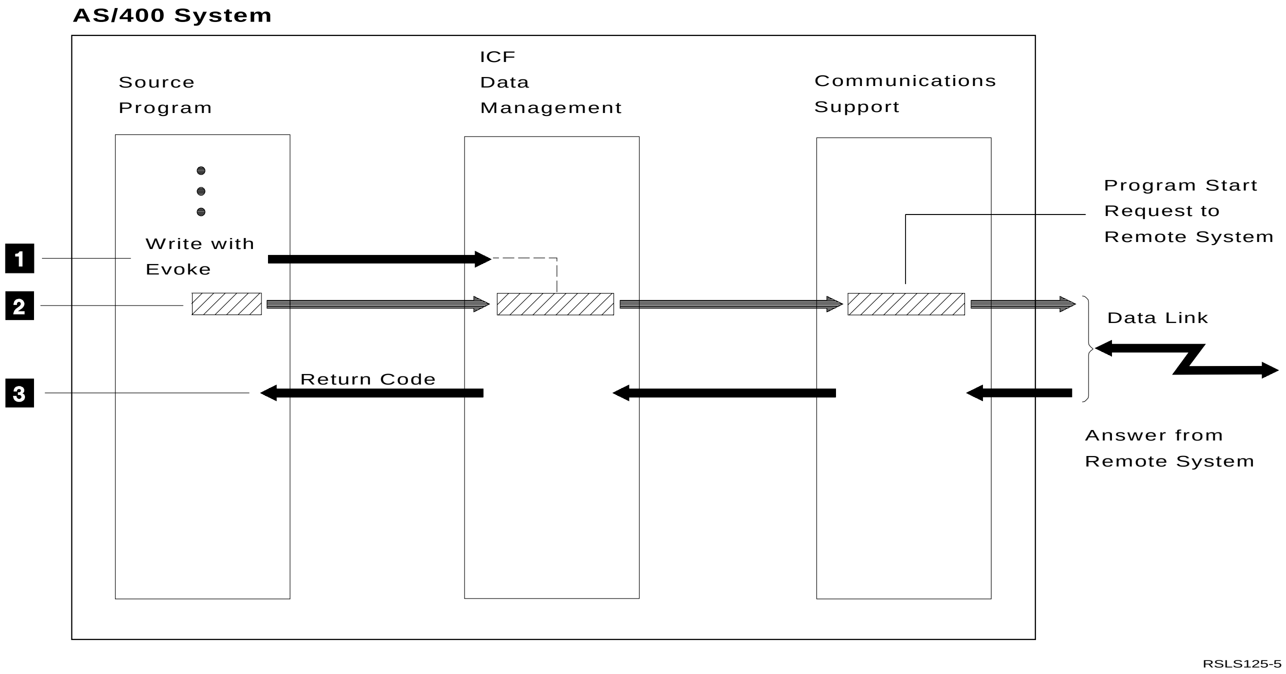


Figure 6-1. Starting a Target Program

Differences between DDS and System-Supplied Evoke Functions

The DDS EVOKE keyword is handled differently from the system-supplied evoke formats when data is also sent. The data passed on a system-supplied evoke format is treated as a parameter and is passed along with the program start request to the remote system. Any data passed using the EVOKE keyword, except for user-defined parameters, program names, and library names, is not treated as a parameter. This information is sent separately from the program start request after the evoke request completes successfully. Therefore, the remote program must issue a read operation to receive the data when the DDS EVOKE keyword is used.

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 6-7 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.

Variable-Buffer-Management (VARBUFMGT)

Use the VARBUFMGT keyword to send or receive multiple or partial records, rather than just one record, with one record format per write or read operation.

Using the VARBUFMGT keyword allows you to specify the length of data independently of the data itself. A program uses the data length specified as the value passed in the variable length (VARLEN) DDS keyword, or if VARLEN is not used, the length of the record format specified on the read or write operation. The length specified must be greater than zero.

This function is valid only for APPC. Refer to the *APPC Programming* book for more information.

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 Indicator Keywords” on page 6-18 and the RSPCONFIRM keyword described in “Additional Keywords” on page 6-12 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 Indicator Keywords" on page 6-18 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 Indicator Keywords" on page 6-18 for information on how to handle receiving a function-management-header.

Control-Data (CTLDTA)

Use the CTLDTA keyword to send control data to the remote program. Control data has meaning only to the partner transaction programs. For example, this data can be used as prefix control information for application data that follows it, or it can be used to carry special data for mapped conversation transactions.

This keyword is valid only for APPC. See the *APPC Programming* book for more information.

Prepare-for-Commit Function

Your program uses the **prepare-for-commit** (PRPCMT) function to request one of its partners to prepare to commit its protected resources. The partner can respond with a commit, a rollback, or a FAIL operation. If the partner responds with a FAIL operation, the partner program is in control and can attempt to correct any errors that it detected.

The PRPCMT function contrasts with the commit operation in the following ways:

- PRPCMT only works with one conversation at a time. The commit operation attempts to commit all protected resources in the two-phase commit transaction.
- PRPCMT only prepares the remote protected resources to be committed. In other words, the remote resources have been locked and cannot be changed. They are in a state in which they can either be committed or rolled backed. Eventually, the remote resources are committed or rolled back depending on whether the rest of the two-phase commit transaction commits or rolls back its protected resources.

  The commit operation ends only after all remote protected resources in the two-phase commit transaction. have either been committed or rolled back.
- PRPCMT allows the application program to attempt error recovery without rolling back the protected logical unit of work (LUW). When the application program issues a PRPCMT and the partner responds with a fail function, the PRPCMT function completes. The application program can then attempt error recovery, and issue the PRPCMT function again.

  Note: The remote program is in send state after responding with the fail function. The local application program cannot issue the PRPCMT function again until the conversation states change.

  When the application program issues a commit operation and the partner responds with a fail function, the logical unit of work is rolled back.

An operation that includes the prepare-for-commit function does not complete until the remote program responds with a commit or rollback operation or a FAIL or EOS function.

After the PRPCMT function completes successfully, your program can do any one of the following.

- Use the commit operation to commit protected resources.
- Use the rollback operation to roll back the protected logical unit of work (LUW).
- Use the end-of-session function to end the attachment of the program to a session and roll back the protected LUW.

Note: The prepare-for-commit function only applies when SYNLVL(*COMMIT) is specified in the EVOKE DDS record format used by the source program, or when the program start request received by a target program establishes a synchronization level of commit. An AS/400 target program can determine the synchronization level established by the source program by using the get-attributes operation.

The prepare-for-commit function causes any data currently held in the buffer to be sent, including any data on a write operation that specified the prepare-for-commit function.

Transaction-Synchronization-Level Function

Your program uses the **transaction-synchronization-level** (TNSSYNLVL) function to specify that synchronization for this transaction should be done at the level that the SYNLVL keyword specified on the evoke.

The TNSSYNLVL keyword can only be used if specified with one of the following keywords.

- ALWWRT
- DETACH
- INVITE

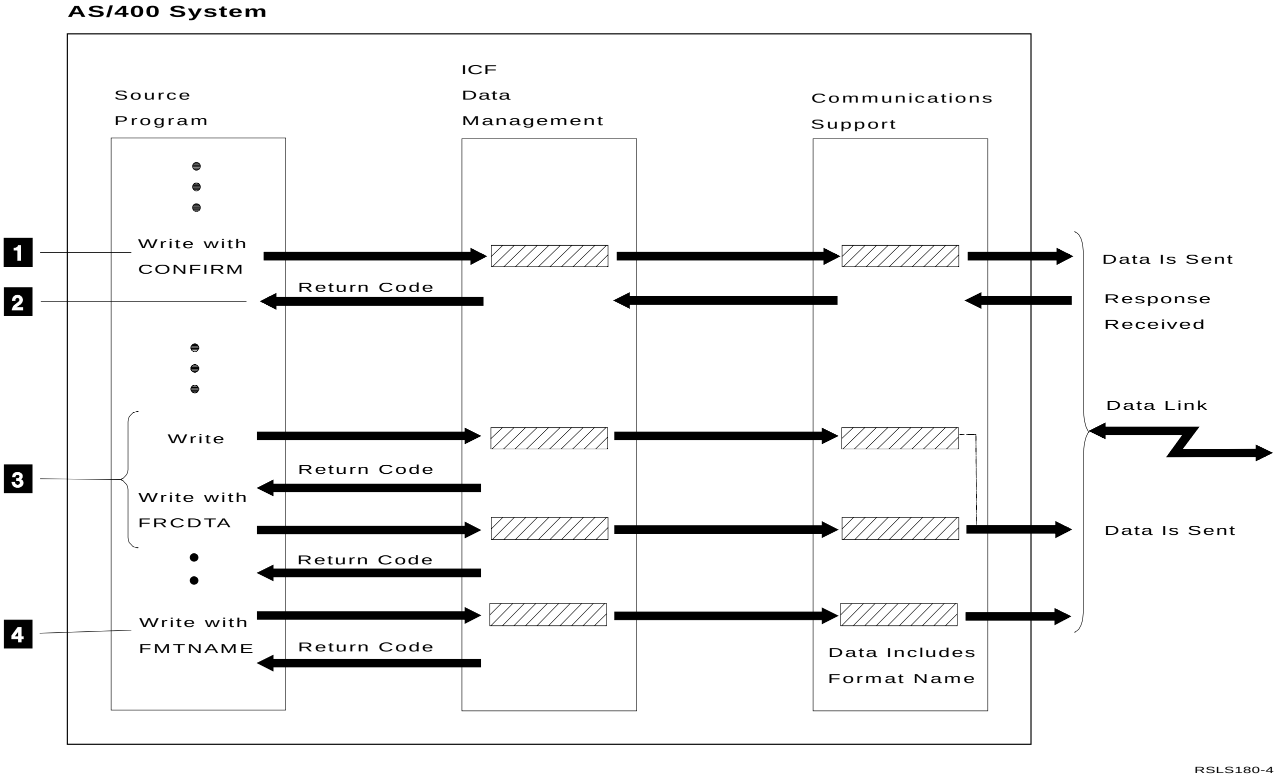


Figure 6-2. Using the CONFIRM, FRCDTA, and FMTNAME Keywords to Send Data

Examples of Sending Data

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

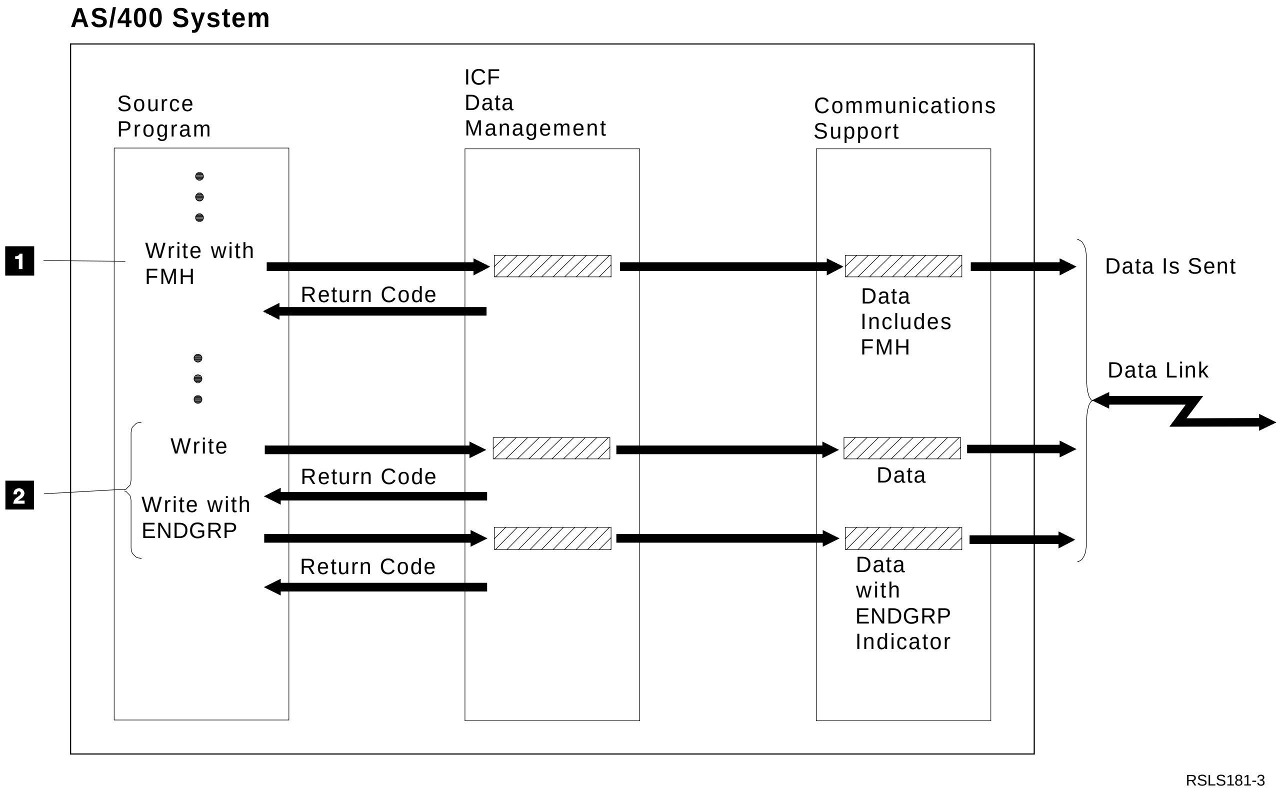


Figure 6-3. Using the ENDGRP and FMH Keywords to Send Data

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.

Note: When a program device is invited, it is recommended that a read-from-invited-program-devices operation be performed rather than a read operation to receive data. Performance may be degraded if your program issues multiple read operations to invited program devices.

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

Invite with Transaction Synchronization Level: When your application program specifies the TNSSYNLVL keyword with the invite function, the additional function performed depends on the synchronization level of the conversation. The TNSSYNLVL keyword can be specified with the invite function only if the synchronization level is *NONE or *CONFIRM. Figure 6-4 on page 6-8 shows the details.

Figure 6-4. TNSSYNLVL Function with Invite

Synchronization Level	Function
*NONE	The force-data function is performed in addition to the invite function.
*CONFIRM	The confirm function is performed in addition to the invite function.
*COMMIT	Not allowed

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.

There is a minor difference between the functions of the TIMER keyword and the WAITRCD parameter. When a write operation is done using the TIMER keyword, the timer starts immediately. The time interval is no longer in effect when a subsequent read-from-invited-program-devices operation completes or when the end of the interval is reached. When the WAITRCD parameter is used, the timer starts when a read-from-invited-program-devices operation is performed.

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 alphanumeric data in fields that should contain only numeric data. Refer to “Input Considerations” on page 8-3 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 6-5 on page 6-10 shows how to use the fail function when your program is receiving data and detects an error.

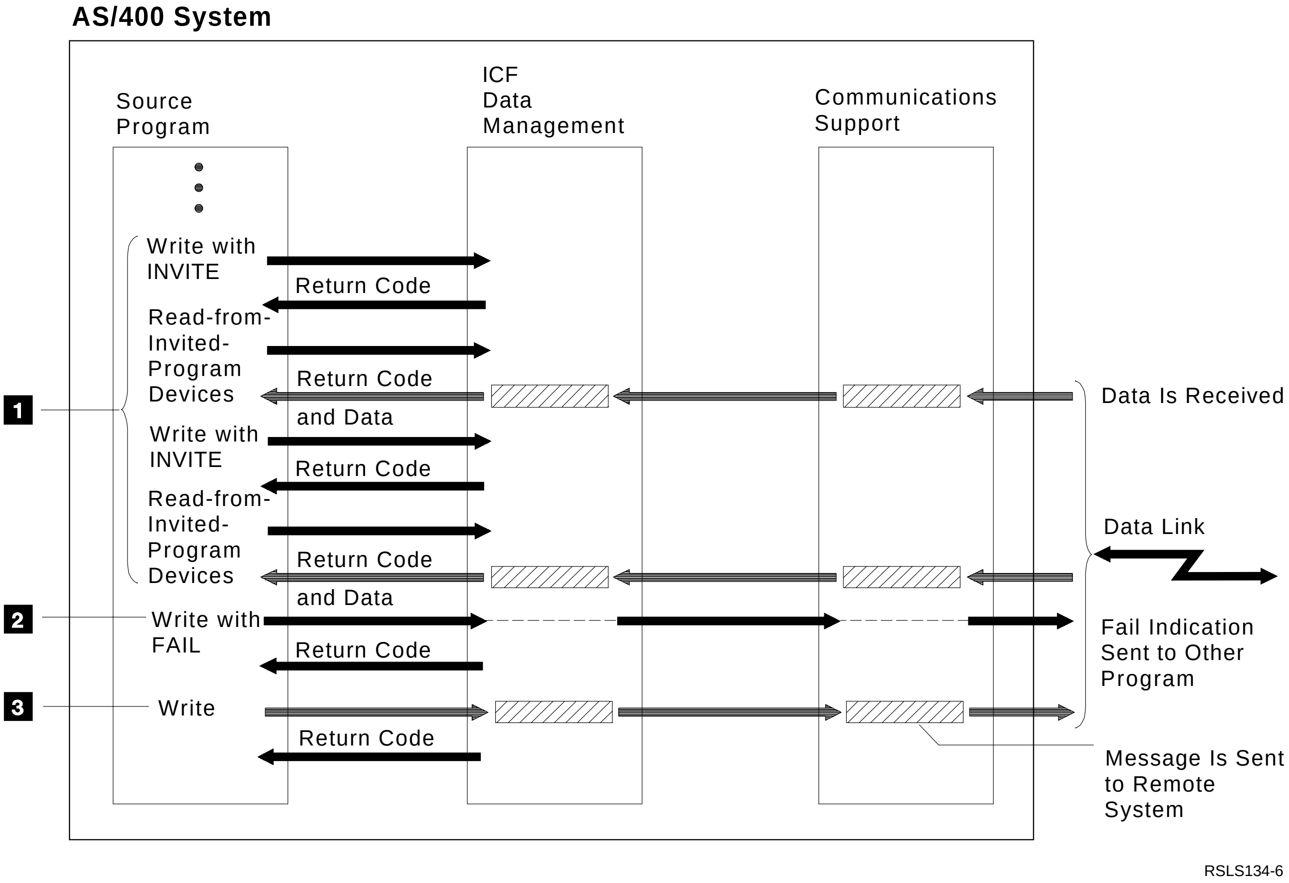


Figure 6-5. Using the FAIL Keyword to Send an Error Indication

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.

When a fail function is the response to a commit operation, the system rolls back the protected LUW on the side that issued the commit operation. The side that issued the fail function must do a rollback operation after the request to roll back is received from the partner. When a fail function is the response to a PRPCMT function, APPC does not roll back the protected LUW. Since APPC does not do a rollback for PRPCMT, the application program can try to correct the problem.

Refer to the RCVFAIL keyword described in “Using Response Indicator Keywords” on page 6-18 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 6-6 on page 6-11 shows how to use the cancel function when a program is sending data and detects an error.

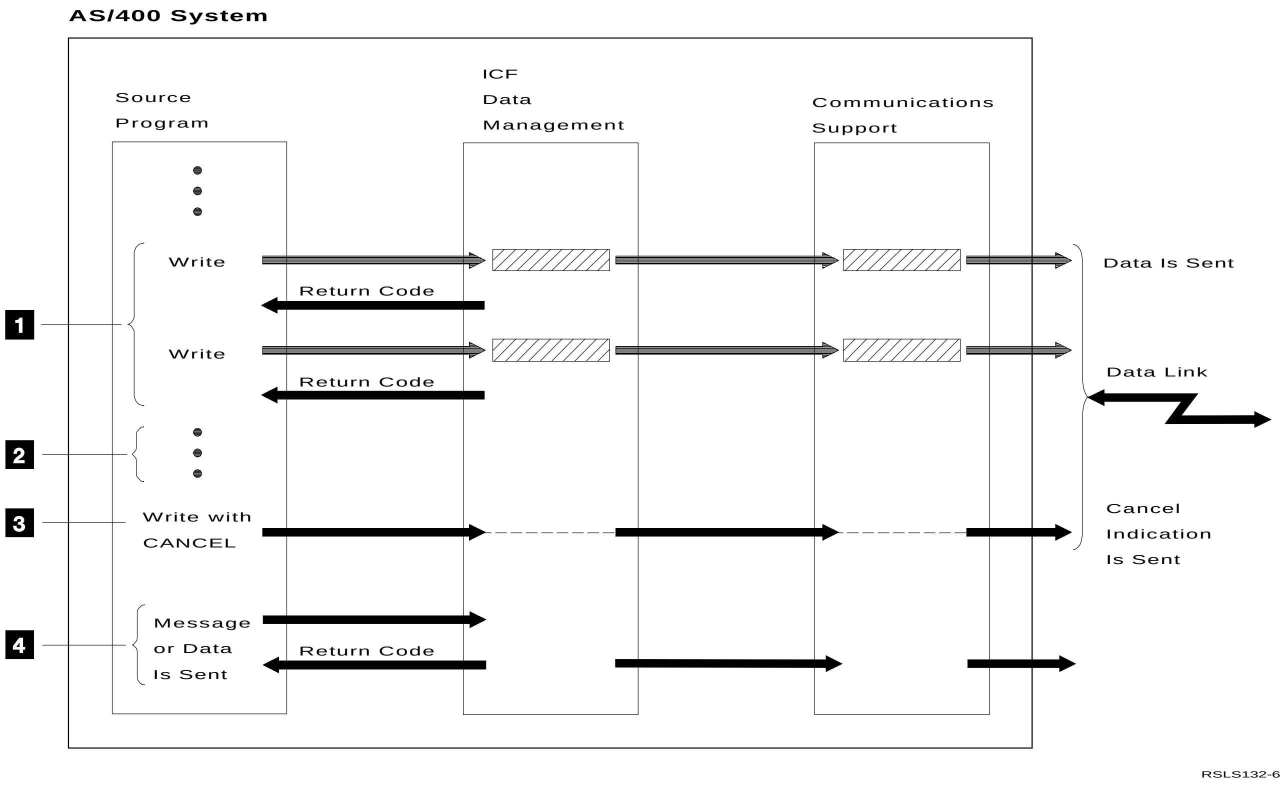


Figure 6-6. Using the CANCEL Keyword to Send an Error Indication

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 Indicator Keywords” on page 6-18 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 book 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 6-7 on page 6-12 shows how to send a negative response with a sense code to the remote system.

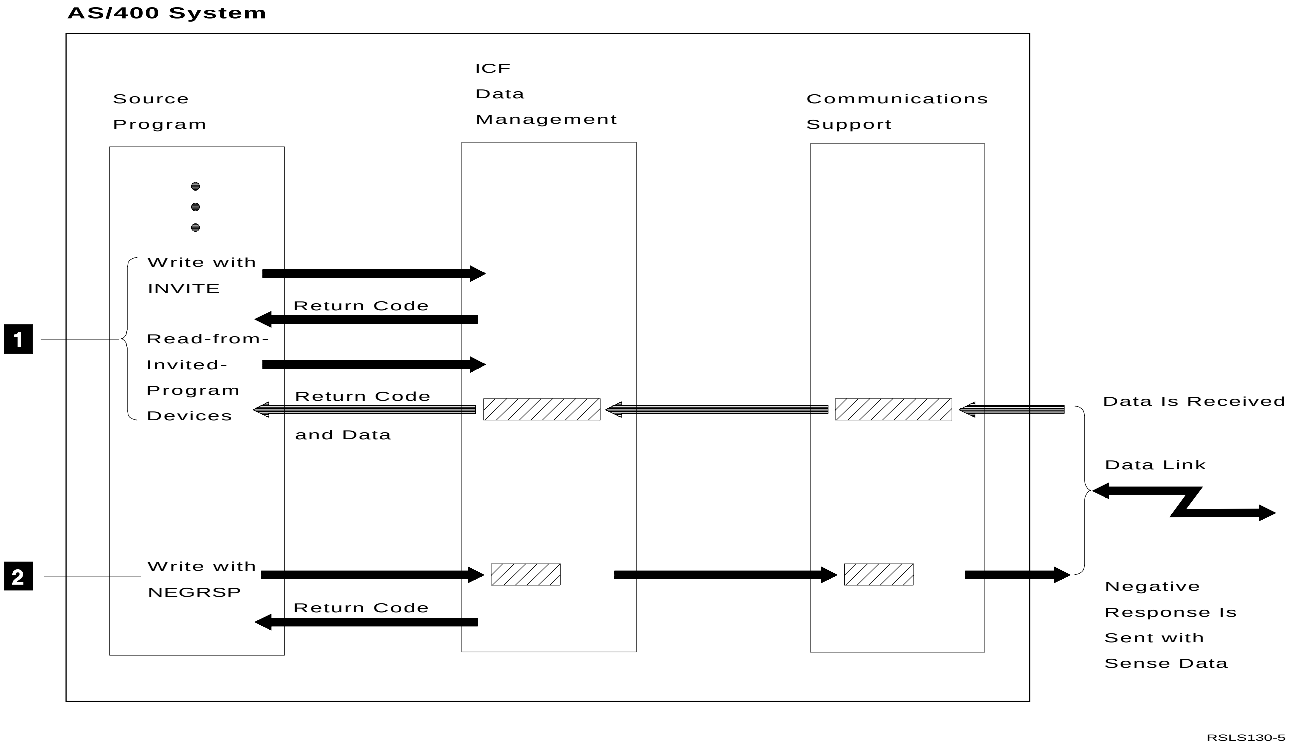


Figure 6-7. Sending a Negative Response with Sense Code to Remote System

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 Indicator Keywords” on page 6-18 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 6-8 on page 6-13 shows how to use the respond-to-confirm function.

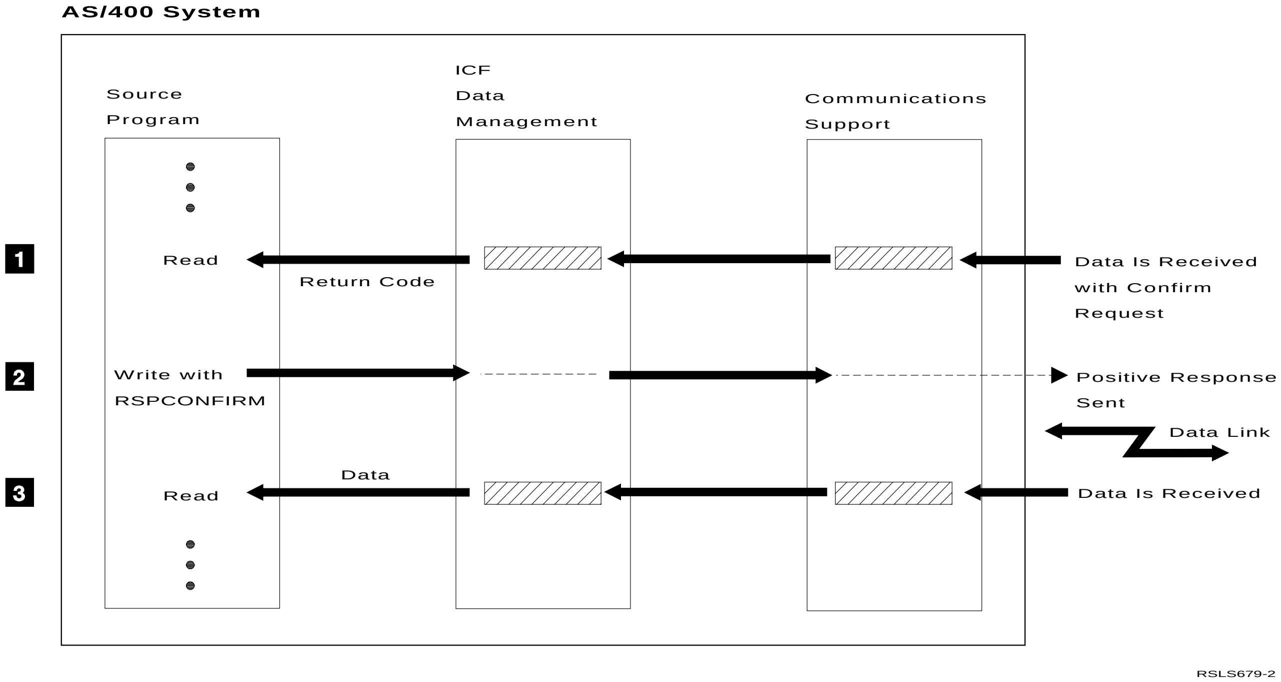


Figure 6-8. Using the Respond-to-Confirm Function

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 6-9 on page 6-14 shows how to use the request-to-write function.

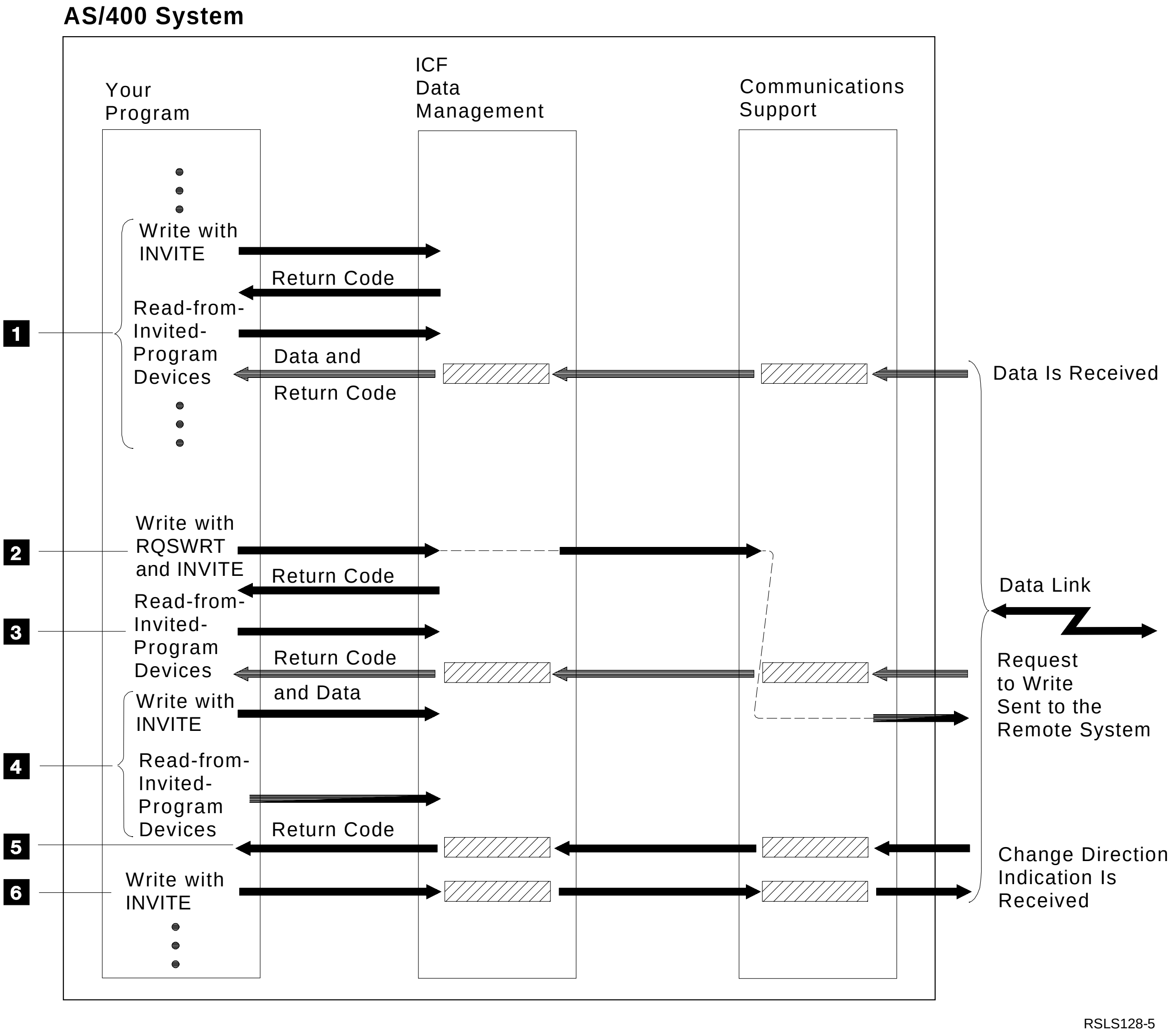


Figure 6-9. Using the Request-to-Write Function

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 Figure C-5 on page C-3 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 programming book 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 6-11 shows how to use 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 6-19 for information on handling receipt of an allow-write indication.

Your application program uses the **allow-write** (ALWWRT) function to inform the remote program that your program is done sending data and is ready to receive. This causes a change-direction indicator to be sent to the remote program.

After issuing the allow-write function, your program can then issue an input operation to receive data from the remote program.

When your application program specifies the TNSSYNLVL keyword with the ALWWRT keyword, the additional function performed depends on the synchronization level of the conversation. Figure 6-10 shows the details.

Figure 6-10. TNSSYNLVL Function with ALWWRT

Synchronization Level	Function
*NONE	The force-data function is performed in addition to the allow-write function.
*CONFIRM	The confirm function is performed in addition to the allow-write function.
*COMMIT	The conversation enters defer receive state until your application program issues a commit operation, a force-data function, or a confirm function. Once the commit operation, force-data function, or confirm function completes successfully, the conversation is in receive state.

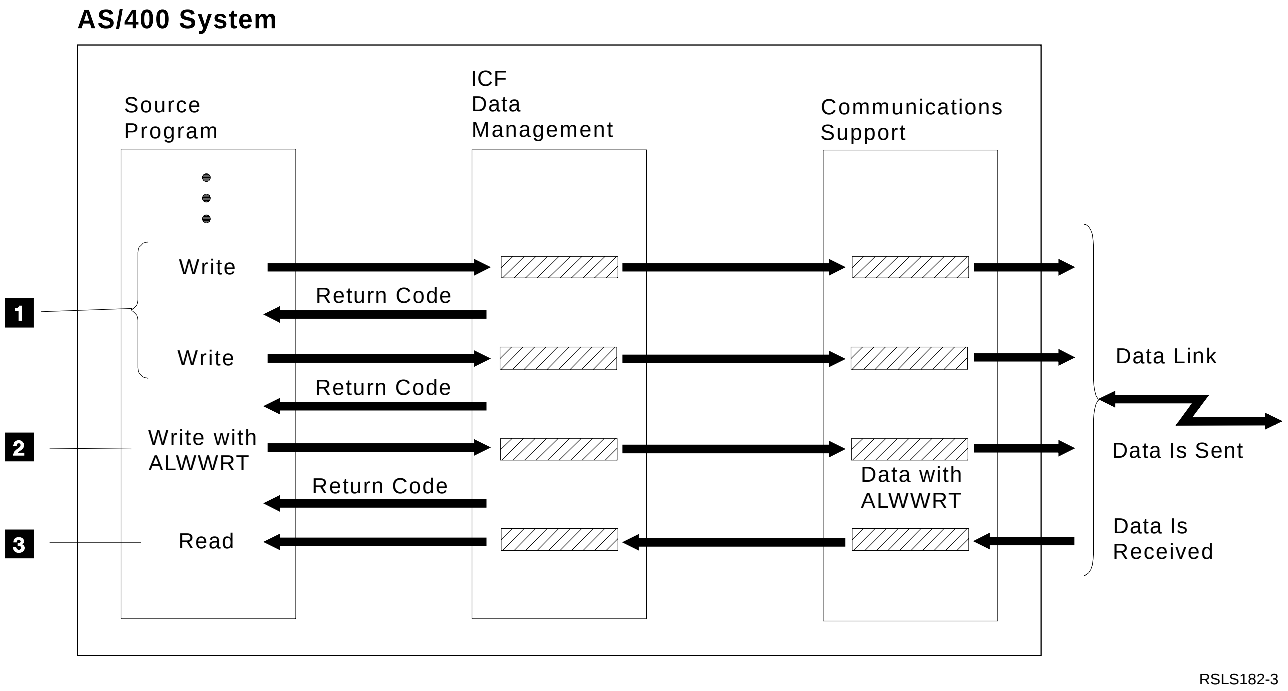


Figure 6-11. Using the Allow-Write Function

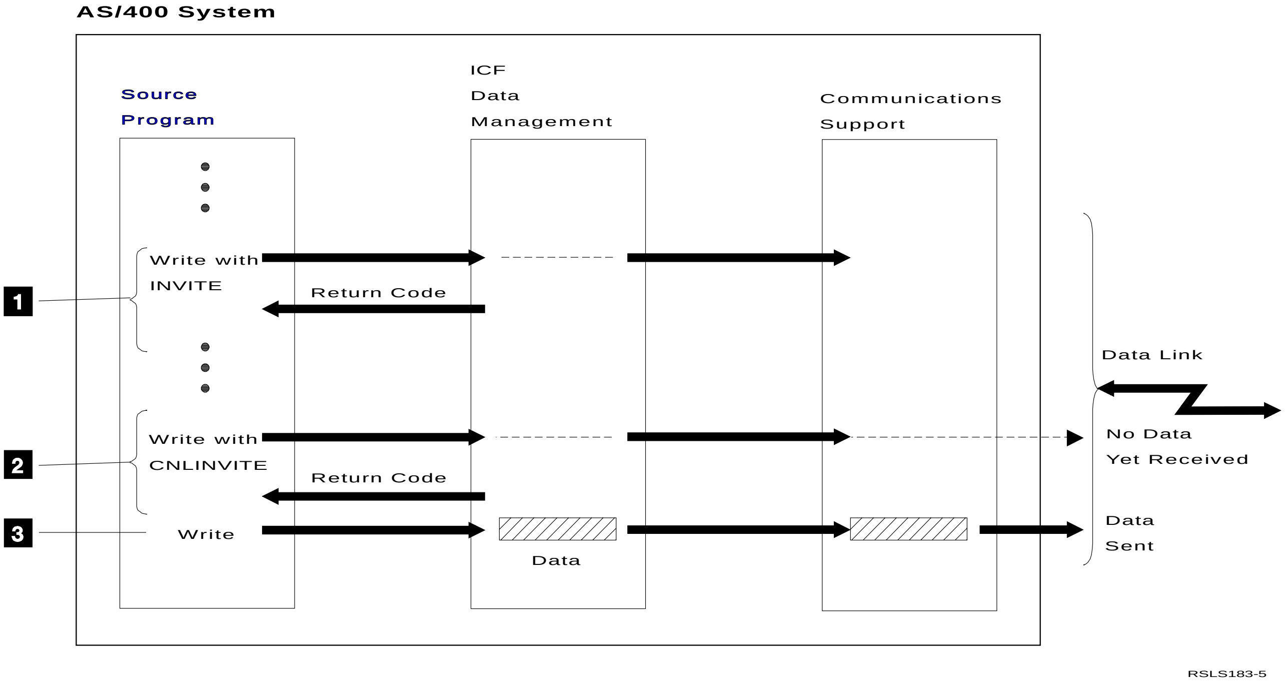


Figure 6-12. Using the Cancel-Invite Function

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 6-12 shows how to use the CNLINVITE keyword.

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 6-13 on page 6-17 shows how your program can end a communications transaction.

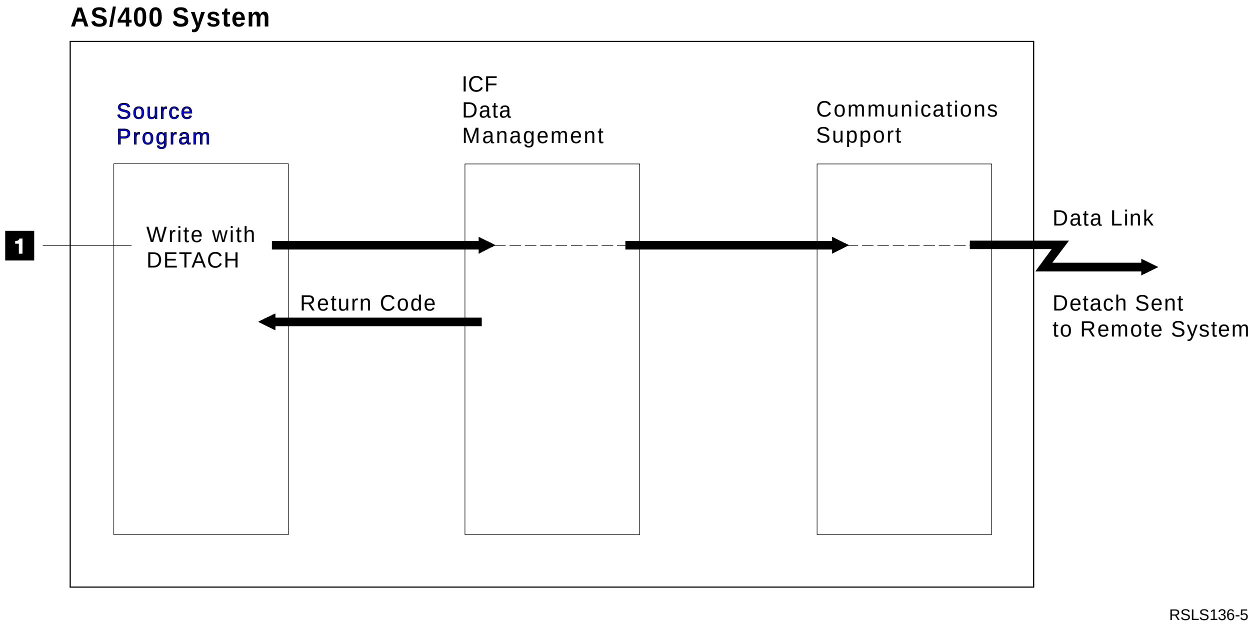


Figure 6-13. Ending a Communications Transaction

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 Indicator Keywords” on page 6-18 for information on handling receipt of a detach indication.

If a detach function is issued by the target program, the EOS function is issued after the detach function is completed.

This should be done since neither the EOS, Detach, or any other ICF function can end the session (that is, cause an UNBIND to be sent).

Using the Detach Function When the Synchronization Level is None

When the synchronization level is none and the detach and transaction-synchronization-level functions are used together, force-data and detach functions are performed.

After a detach function is accepted by your program, no further input or output operations with the remote program are allowed.

Using the Detach Function When the Synchronization Level is Confirm

When a detach function and a confirm function or transaction-synchronization-level function are used together, a confirm function is performed. If the remote program responds positively, the detach function is performed. If the remote program responds negatively, or has already sent a negative response, the transaction may not end immediately. The sender of the negative response is responsible for the initial error recovery. The point at which action is taken to recover from the error determines when the transaction is ended.

To respond positively to the detach function with a confirm or transaction-synchronization-level function, the remote program must use the respond-to-confirm function.

To respond negatively to the detach function with a confirm or transaction-synchronization-level function, the remote program should use the fail function.

After a detach function is accepted by your program, no further input or output operations with the remote program are allowed.

Using the Detach Function When the Synchronization Level is Commit

For two-phase commit processing, the detach function must be accompanied by the transaction-synchronization-level function. The transaction does not end until your program issues a commit operation, and the commit operation completes successfully. If the commit operation fails, the following is done.

- The logical unit of work is rolled back.
- The transaction is not ended.
- The conversation state is returned to what it was at the last commit boundary.

Using the Detach Function From a Target Program

After a target program issues a detach function, both the session and the transaction end. No further operations are valid on the program device.

Ending the Communications Session

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

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 target program ends the transaction by the detach function, the session is ended implicitly. If the source program ends the transaction, the target program must issue an end-of-session or go to the end of the job to end the session. Figure 6-14 shows how you can end the session using the release operation and the end-of-session function.

For conversations started using EVOKE with SYNLVL(*COMMIT) specified:

- If EOS is issued after a TAKE_COMMIT_* indication has been received by the transaction program (TP), resynchronization processing is performed.
- In all other cases, the EOS causes the logical unit of work (LUW) to be put into rollback required state. The TP must perform a rollback operation before working with any other resources involved in that LUW.

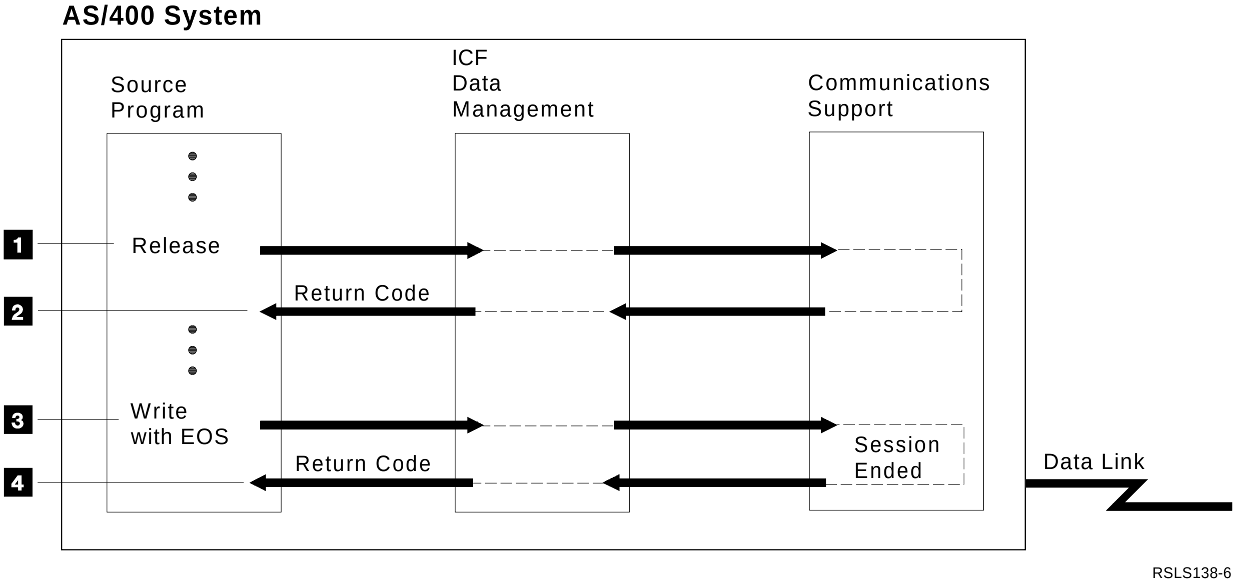


Figure 6-14. Using the Release and End-of-Session Functions

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 Indicator Keywords

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-Control-Data

Use the RCVCTLDTA keyword to request that a response indicator be set on if the record received from the remote system contains a control-data indication. The response indicator is set if the data received in the input buffer is control data.

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.

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.

Receive-Rollback

Use the RCVROLLB keyword to request that a response indicator be set on as an indication of one of the following conditions:

- The remote program sent a ROLLBACK. This indicates that the remote program expects your program to rollback its protected resources.

- The protected LUW entered the rollback required state.

Your program must respond with a rollback operation. Your program can only get this response indicator if it has a conversation with a synchronization level of commit.

This response indicator can be received with the following return codes.

- 0054
- 0254
- 80F9, 80FA, 80FB
- 81F0, 81F1, 81F2, 81F3, 81F4, 81F5
- 83FB, 83FC, 83FD, 83FE, 83FF

Receive-Take-Commit

Use the RCVTKCMT keyword to request that a response indicator be set on as an indication that the remote program sent a PRPCMT function or a commit operation. This indicates that the remote program expects your program to determine if it can commit its protected resources. Your program must either do a commit or rollback operation or a FAIL or EOS function. Your program can only get this response indicator if it has a conversation with a synchronization level of commit.

This response indicator can be received with major return codes 02 (end job or end subsystem in progress) or 03 (no data received). The major return code can be accompanied by a minor return code of 57, 58, or 59.

Example DDS Files for Creating an Intersystem Communications Function File

Figure 6-15 and Figure 6-16 on page 6-21 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 6-15. DDS Source File for a Batch Data Transfer Program

```
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                                      RCVTRNRND(90)
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(&LIB/&PGMID)
A            PGMID         10A  P
A            LIB           10A  P
A          R ITMREQ
A                                      INVITE
A            ITEMNO         6  0
A          R DTLREQ
A                                      INVITE
A            CUSTNO         6  0
                     * * * *  E N D  O F  S O U R C E  * * * *
```

Figure 6-16. 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* book 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 5-8.

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

Keyword Processing Charts

Figure 6-17 and Figure 6-18 on page 6-22 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.

Figure 6-17 lists the DDS keywords defined in this chapter that are supported by the various communications types for output operations.

Figure 6-17 (Page 1 of 2). Output DDS Processing Keyword Support

DDS Keyword	APPC	SNUF	BSCEL	Asyn-chronous	Intra-system	Finance	Retail
ALWWRT	X	X	X		X		

Figure 6-17 (Page 2 of 2). Output DDS Processing Keyword Support

DDS Keyword	APPC	SNUF	BSCEL	Asyn-chronous	Intra-system	Finance	Retail
CANCEL		X			X	X[2]	X
CNLINVITE		X	X	X	X	X	X
CONFIRM	X				X		
CTLDTA	X						
DETACH	X	X	X	X[1]	X		X
DFREVOKE	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	X
FMTNAME	X				X		
FRCDTA	X				X	X	X
INVITE	X	X	X	X	X	X	X
NEGRSP		X			X	X	X
PRPCMT	X						
RQSWRT	X	X	X		X		
RSPCONFIRM	X	X			X		
SECURITY	X	X	X	X	X		
SUBDEV			X		X		
SYNLVL	X				X		
TIMER	X	X	X	X	X	X	X
TNSSYNLVL	X						
VARBUFMGT[3]	X						
VARLEN	X	X	X	X	X	X	X

[1] Use of these keywords are restricted. Refer to the *Asynchronous Communications Programming* book for more details.

[2] This keyword is not valid for the 3694 controller. Refer to the *Finance Communications Programming* book for more details.

[3] Use of this keyword is restricted. Refer to the *APPC Programming* book.

Figure 6-18 lists the DDS keywords defined in this chapter that are supported by the various communications types for input operations.

Figure 6-18 (Page 1 of 2). Input DDS Processing Keyword Support

DDS Keyword	APPC	SNUF	BSCEL	Asyn-chronous	Intra-system	Finance	Retail
RCVCANCEL		X			X		X
RCVCONFIRM	X	X			X		
RCVCTLDTA	X						
RCVDETACH	X	X	X		X		X
RCVENDGRP		X	X		X	X	X
RCVFAIL	X			X	X		
RCVFMH		X			X	X	X
RCVNEGRSP		X			X	X	X

Figure 6-18 (Page 2 of 2). Input DDS Processing Keyword Support

DDS Keyword	APPC	SNUF	BSCEL	Asyn-chronous	Intra-system	Finance	Retail
RCVROLLB	X						
RCVTKCMT	X						
RCVTRNRND	X	X	X		X		
RECID	X	X	X	X	X	X	X

Figure 6-19 shows the priority sequence used by ICF in processing these DDS keywords and data during output operations.

```
1  Is EOS specified?
   | no       | yes
   |          |
   |          | Do EOS Function
   |
   |         14
   |
2  Is FAIL specified?
   | no       | yes
   |          |
   |          | Do FAIL Function
   |
   |         12
   |
3  Is NEGRSP specified?
   | no       | yes
   |          |
   |          | Do NEGRSP Function
   |          | (Send sense code if specified)
   |
   |         12
   |
4  Is CANCEL specified?
   | no       | yes
   |          |
   |          | Do CANCEL Function
   |
   |         12
   |
5  Is CNLINVITE specified?
   | no       | yes
   |          |
   |          | Do CNLINVITE Function
   |
   |         11
   |
6  Is RSPCONFIRM specified?
   | no       | yes
   |          |
   |          | Do RSPCONFIRM Function
   |
   |         14
   |
7  Is RQSWRT specified?
   | no       | yes
   |          |
   |         Is INVITE or READ specified?
   |          | yes                        | no
   |          |                            |
   |          | Do RQSWRT Function         | Do RQSWRT
   |          |  with Invite or Read       |  Function
   |
   |        14                            14
   |
```

Figure 6-19 (Part 1 of 2). Keyword Processing Chart

```
8  Is EVOKE specified?
   | no       | yes
   |          |
   |         Incorporate DFREVOKE, SECURITY, and SYNLVL keywords
   |          |
   |         Is user data specified? (Not program initialization
   |                 parameters)
   |          | yes                           | no
   |          |                               |
   |          | Do EVOKE Function             | Do EVOKE
   |          |  (Incorporate FMH keyword     |  Function
   |          |   and program initial-        |  (Incorporate
   |          |   ization parameters if       |   rest of
   |          |   specified)                  |   keywords and
   |          |                               |   program
   |          |                               |   initialization
   |         11                             14   parameters
   |                                             appropriately)
9  Is TIMER specified?
   | no       | yes
   |          |
   |          | Do TIMER Function
   |        14
10 Is PRPCMT specified?
   | no       | yes
   |          |
   |          | Perform PRPCMT function
   |          |
   |          |
   |
   |        14
   |
11 Process data (Incorporate VARLEN keyword)
   |            (Incorporate VARBUFMGT keyword)
   |
12 Is DETACH specified?
   | no       | yes
   |          |
   |         Is CONFIRM specified?
   |          | yes                           | no
   |          |                               |
   |          |                               |
   |          | Do DETACH Function            | Do DETACH function
   |          |  with CONFIRM                 |  without CONFIRM
   |          |  (Also process format name    |
   |          |  if FMTNAME keyword           |
   |          |  specified)                   |
   |        14                              14
   |
13 Is INVITE, ALWWRT, ENDGRP, FMH, SUBDEV, FMTNAME, FRCDTA,
   | no       | yes               CONFIRM or GET specified?
   |          |
   |          | Perform INVITE or GET function if specified
   |          | Perform ALWWRT function if specified
   |          | Perform ENDGRP function if specified
   |          | Perform FMH function if specified
   |          | Perform SUBDEV function if specified
   |          | Perform FMTNAME function if specified
   |          | Perform FRCDTA function if specified
   |          | Perform CONFIRM function if specified
   |          | Perform TNSSYNLVL function if specified
   |          | Perform CTLDTA function if specified
   |
   |        14
   |
14 END
```

Figure 6-19 (Part 2 of 2). 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 Figure 7-24 on page 7-21.

Programming examples are included to show you how these system-supplied formats are used. These examples are program segments only. You can find complete ILE C, COBOL/400., and RPG/400 programming 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. Figure 7-23 on page 7-20 summarizes the support provided by each communications type. For more detail, refer to the appropriate communications programming book 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 – Reference* book 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 7-17 for more information on the detach function.

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

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.

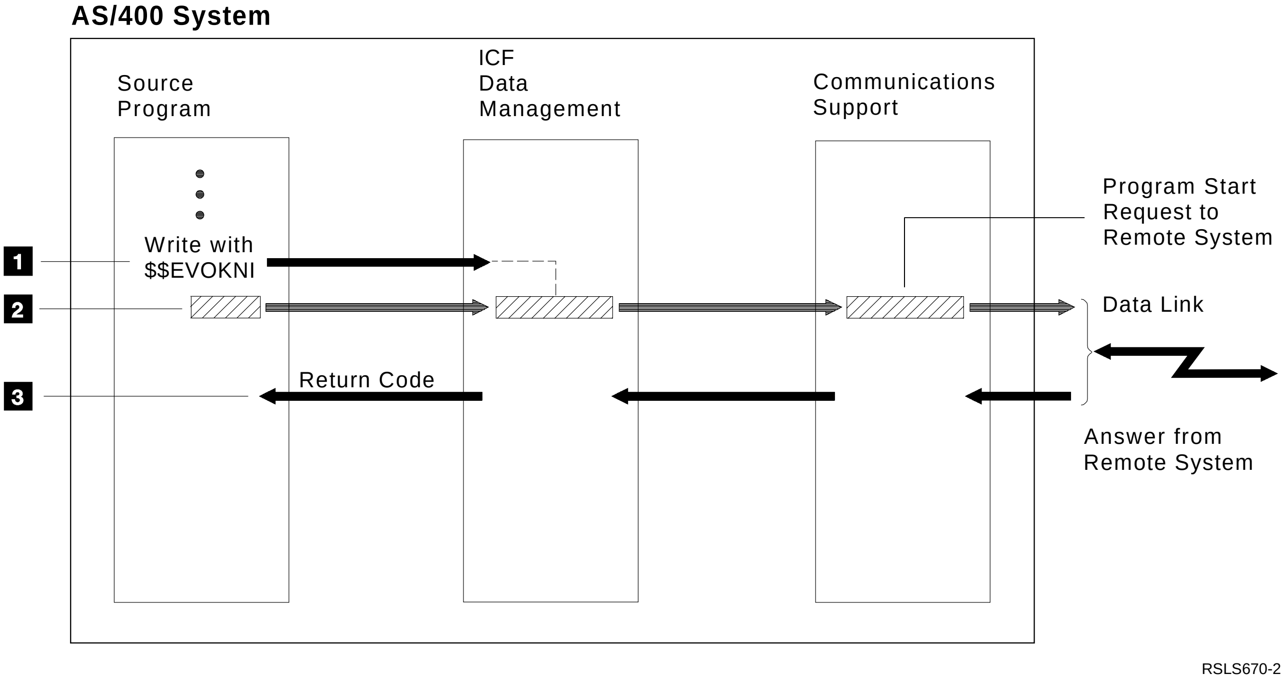


Figure 7-1. Starting a Target Program

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 Figure 7-2.

Figure 7-2. 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.

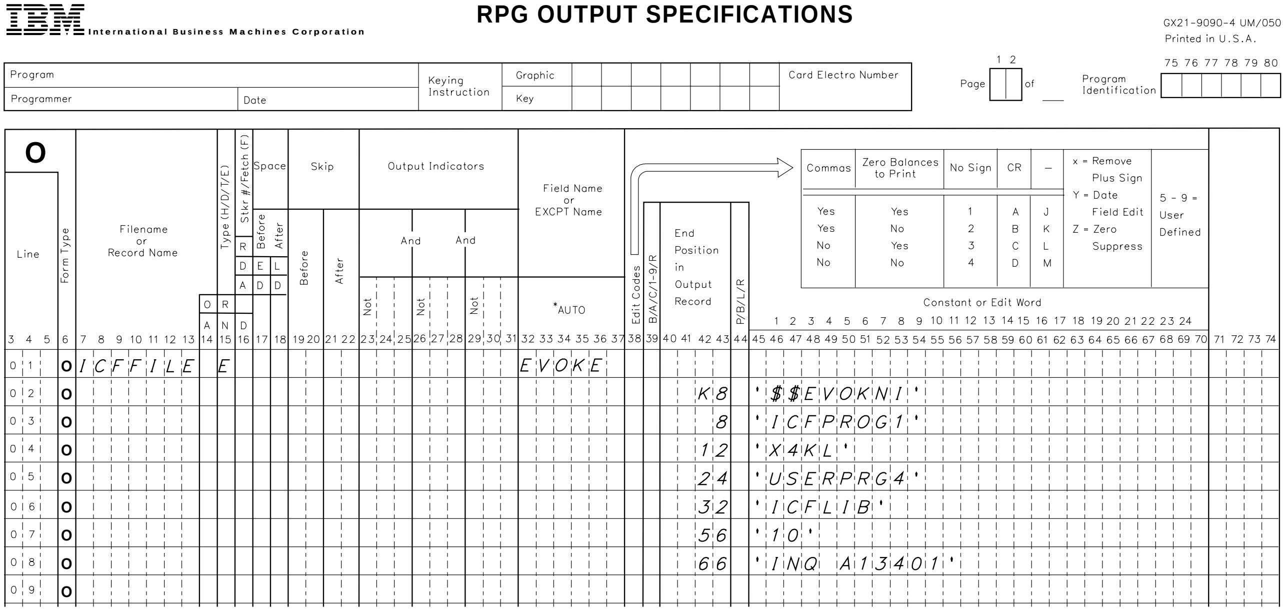
The following is an example of an ILE C write statement that can be used to issue an evoke.

```
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;

⋮
_RFILE *icffptr;         /* Pointer to the ICF file */

⋮
icffptr = _Ropen("ICFFILE","ab+ indicators=y riofb=y");

⋮
_Rformat(icffptr, "$$EVOKNI");
                    /* Set evoke w/no invite format */
_Rpgmdev(icffptr, "CM1");
                       /* Set default device to CM1 */

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

The following 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 on page 7-4 is an example of an RPG/400 output specification used to issue an evoke function.



RPG OUTPUT SPECIFICATIONS

GX21-9090-4 UM/050*
Printed in U.S.A.

Line	Form Type	Filename or Record Name	Type (H/D/T/E)	Field Name or EXCPT Name	End Position in Output Record	Constant or Edit Word
01	O	ICFFILE	E	EVOKE		
02	O				K8	'$$EVOKNI'
03	O				8	'ICFPROG1'
04	O				12	'X4KL'
05	O				24	'USERPRG4'
06	O				32	'ICFLIB'
07	O				56	'10'
08	O				66	'INQ A13401'
09	O					

RSLS184-1

Figure 7-3. 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 7-17 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 7-4 on page 7-5 shows how to use system-supplied formats to send data.

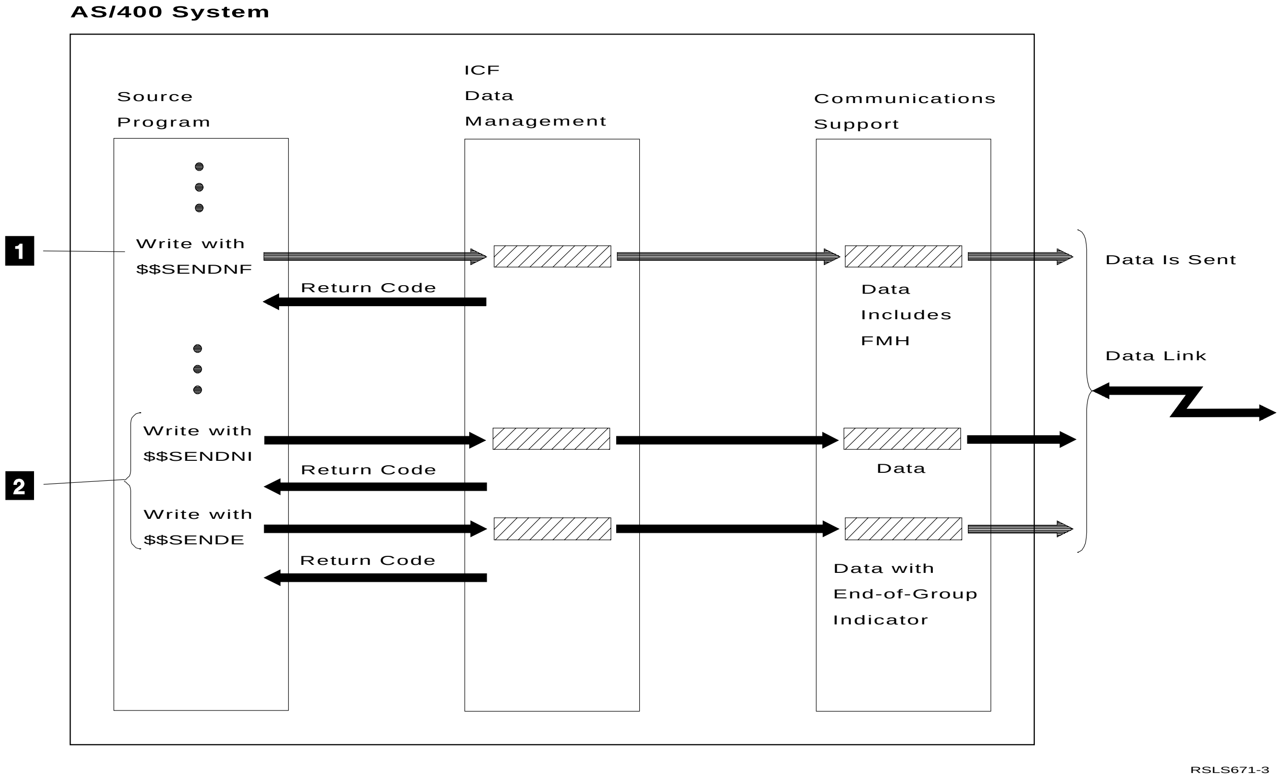


Figure 7-4. Using $$SENDNF, $$SENDNI, and $$SENDE to Send Data

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 Figure 7-5 in the output buffer.

Figure 7-5. Required Output Fields

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

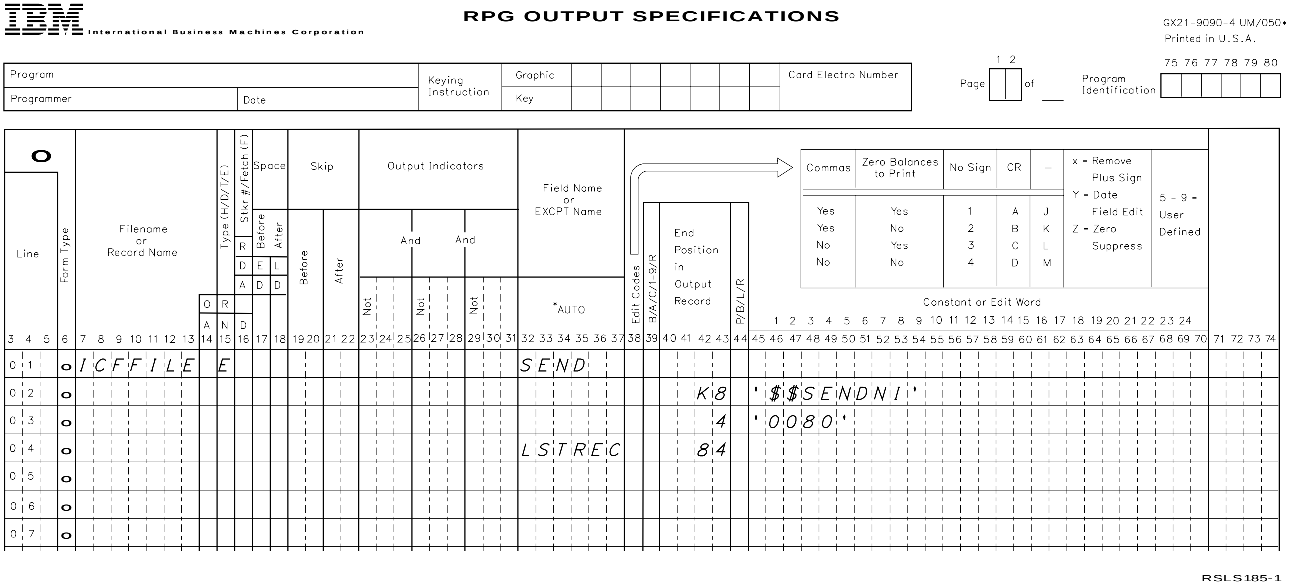
The following is an example of an ILE C write statement that sends one data record.

```
struct {
    char record_length??(4??);
    char data??(80??);
} data_rec;

⋮
_RFILE *icffptr;            /* Pointer to the ICF file */

⋮
icffptr = _Ropen("ICFFILE","ab+ indicators=y riofb=y");

⋮
_Rformat(icffptr, "$$SENDNI");
                      /* Set write w/no invite format */
_Rpgmdev(icffptr, "CM1"); /* Set default device to CM1 */

strncpy(data_rec.record_length, "0080", 4);/*Set record length*/
_Rwrite(icffptr, &data_rec, sizeof(data_rec));/* Do the write */
```

The following 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 7-6 on page 7-6 is an example of an RPG/400 output specification to send one data record.



IBM International Business Machines Corporation

RPG OUTPUT SPECIFICATIONS

GX21-9090-4 UM/050*
Printed in U.S.A.

```
  01 OICFFILE E                SEND
  02 O                                    K8 '$$SENDNI'
  03 O                                     4 '0080'
  04 O                         LSTREC     84
```

RSLS185-1

Figure 7-6. 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 ($$CANL)**. Cancels the current chain of data, then invites the remote program to send its own data. Refer to “Problem Notification” on page 7-8 for more information about the cancel function.
- **Negative Response with Invite ($$NRSP)**. Sends a negative-response indication to the remote program, followed by an invite. Refer to “Problem Notification” on page 7-8 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 7-13 for more information about the request-to-write function.

Refer to “Starting a Program on the Remote System” on page 7-1, “Problem Notification” on page 7-8, and “Additional System-Supplied Formats” on page 7-13 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.

Note: When a program device is invited, it is recommended that a read-from-invited-program-devices operation be performed rather than a read operation to receive data. Performance may be degraded if your program issues multiple read operations to invited program devices.

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.

The following is an ILE C example that shows how to use $$TIMER and set the timer to 30 seconds.

```
_RFILE *icffptr;              /* Pointer to the ICF file */

⋮
icffptr = _Ropen("ICFFILE","ab+ indicators=y riofb=y");

⋮
_Rformat(icffptr, "$$TIMER"); /* Set timer format */
_Rpgmdev(icffptr, "CM1");     /* Set default device to CM1 */

_Rwrite("000030", 6);         /* Issue timer function */

_Rreadindv(icffptr, &record, sizeof(record), __DFT);
                             /* Issue RFI */

      /* 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 */
```

The following 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.
```

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.

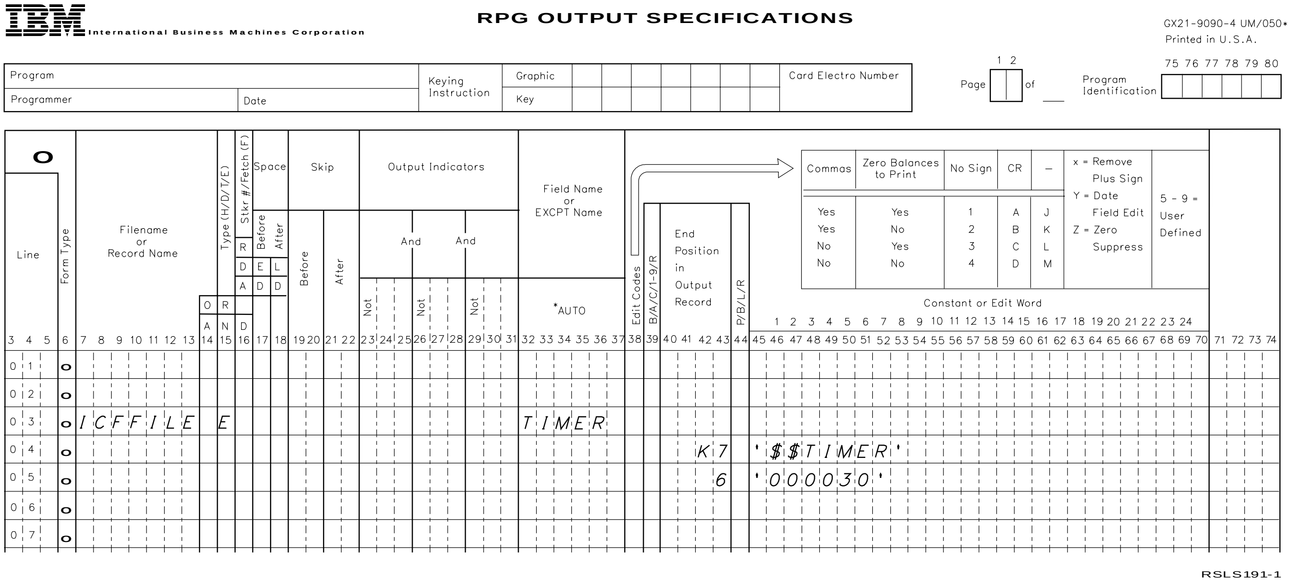
The timer function can be used to vary the maximum amount of time that a read-from-invited-program-devices operation will wait for a response. When the time interval set by the TIMER keyword is in effect, the value specified for the WAITRCD parameter on the CRTICFF command is ignored.

There is a minor difference between the functions of the $$TIMER format and the WAITRCD parameter. When a write operation is done using the $$TIMER format, the timer starts immediately. The time interval is no longer in effect when a subsequent read-from-invited-program-devices operation completes or when the end of the interval is reached. When the WAITRCD parameter is used, the timer starts when a read-from-invited-program-devices operation is performed.

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 7-7 on page 7-8 is an example of using RPG/400 programming language to enter the value on the output specifications to set the timer for 30 seconds.



IBM International Business Machines Corporation

RPG OUTPUT SPECIFICATIONS

GX21-9090-4 UM/050*
Printed in U.S.A.

Line	Form Type	Filename or Record Name	Type (H/D/T/E)	Field Name or EXCPT Name	End Position in Output Record	Constant or Edit Word
01	O					
02	O					
03	O	ICFFILE	E	TIMER		
04	O				K7	'$$TIMER'
05	O				6	'000030'
06	O					
07	O					

RSLS191-1

Figure 7-7. 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 7-8 on page 7-9 shows how to use the $$FAIL communications format when your program is receiving data and detects an error.

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.

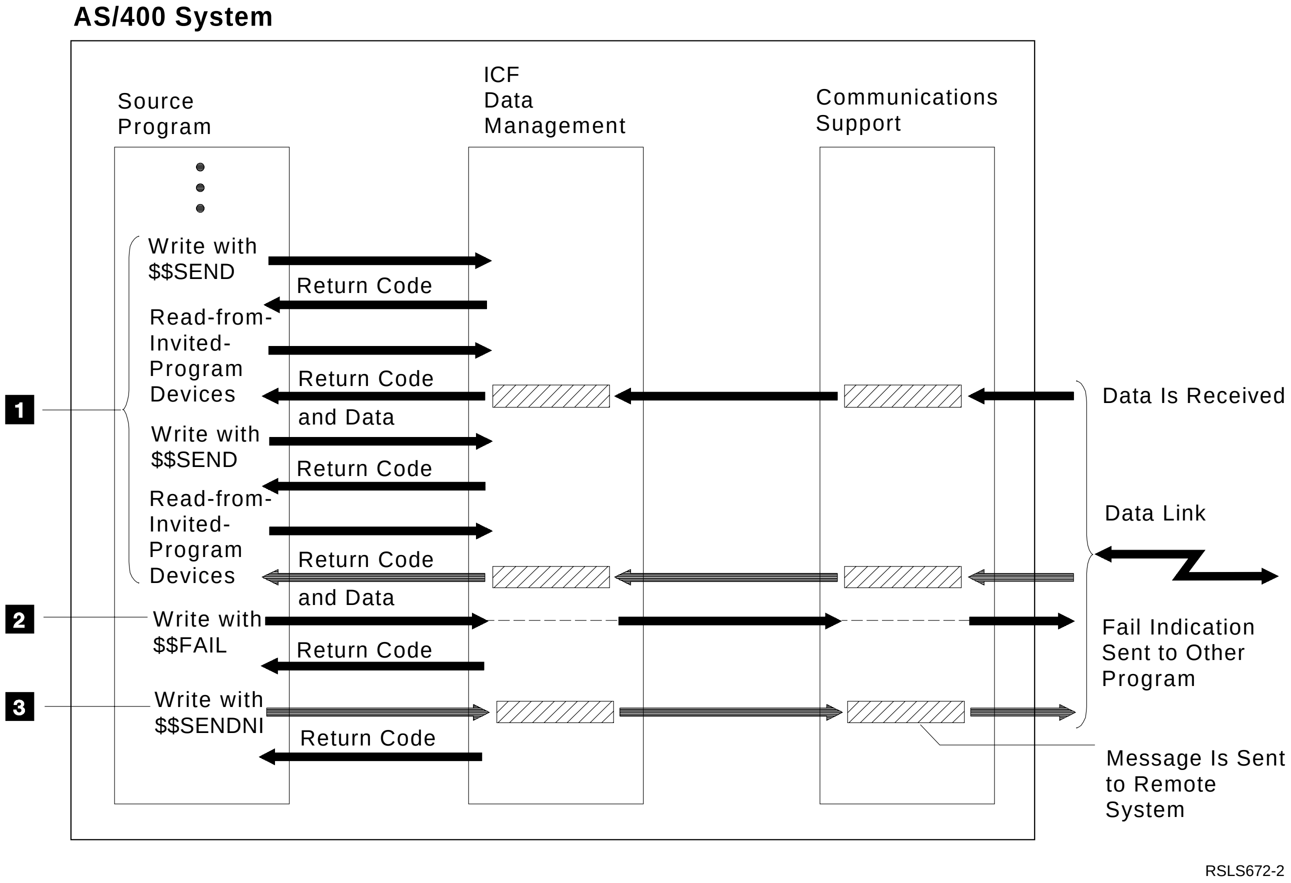


Figure 7-8. Using $$FAIL to Send an Error Signal

No output fields are associated with the $$FAIL communications format.

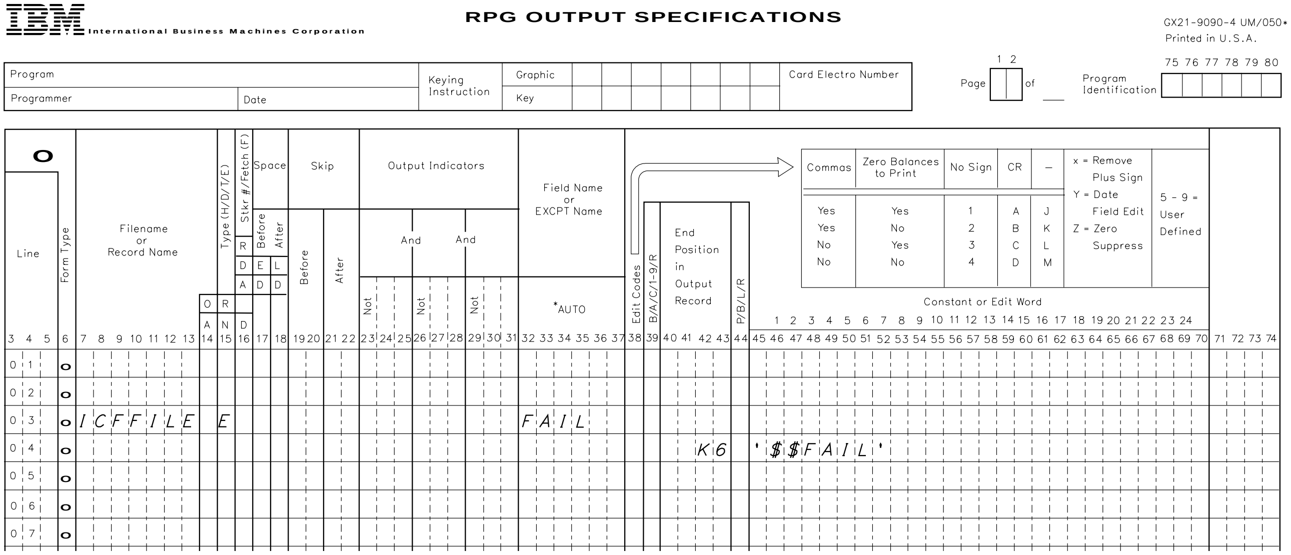
The following is an ILE C write statement example that sends a fail indication.

```
_RFILE *icffptr;             /* Pointer to the ICF file */

⋮
icffptr = _Ropen("ICFFILE","ab+ indicators=y riofb=y");

⋮
_Rformat(icffptr, "$$FAIL");  /* Set fail format */
_Rpgmdev(icffptr, "CM1");     /* Set default device to CM1 */

_Rwrite(icffptr, NULL, 0);    /* Send the fail */
```

The following is a COBOL/400 WRITE statement example that sends a fail indication.

```
WRITE TRANSACTION-RECORD,
   FORMAT IS '$$FAIL', TERMINAL IS ICF-PGMDEV.
```

Figure 7-9 on page 7-10 is an example of an RPG/400 output specification to send a fail indication.



IBM International Business Machines Corporation

RPG OUTPUT SPECIFICATIONS

GX21-9090-4 UM/050*
Printed in U.S.A.

Program		Keying Instruction	Graphic
Programmer	Date		Key

Card Electro Number

Page □ of ___

Program Identification □□□□□□

Line	Form Type	Filename or Record Name	Type (H/D/T/E)	Field Name or EXCPT Name	End Position in Output Record	Constant or Edit Word
01	O					
02	O					
03	O	ICFFILE	E	FAIL		
04	O				K6	'$$FAIL'
05	O					
06	O					
07	O					

RSLS186-1

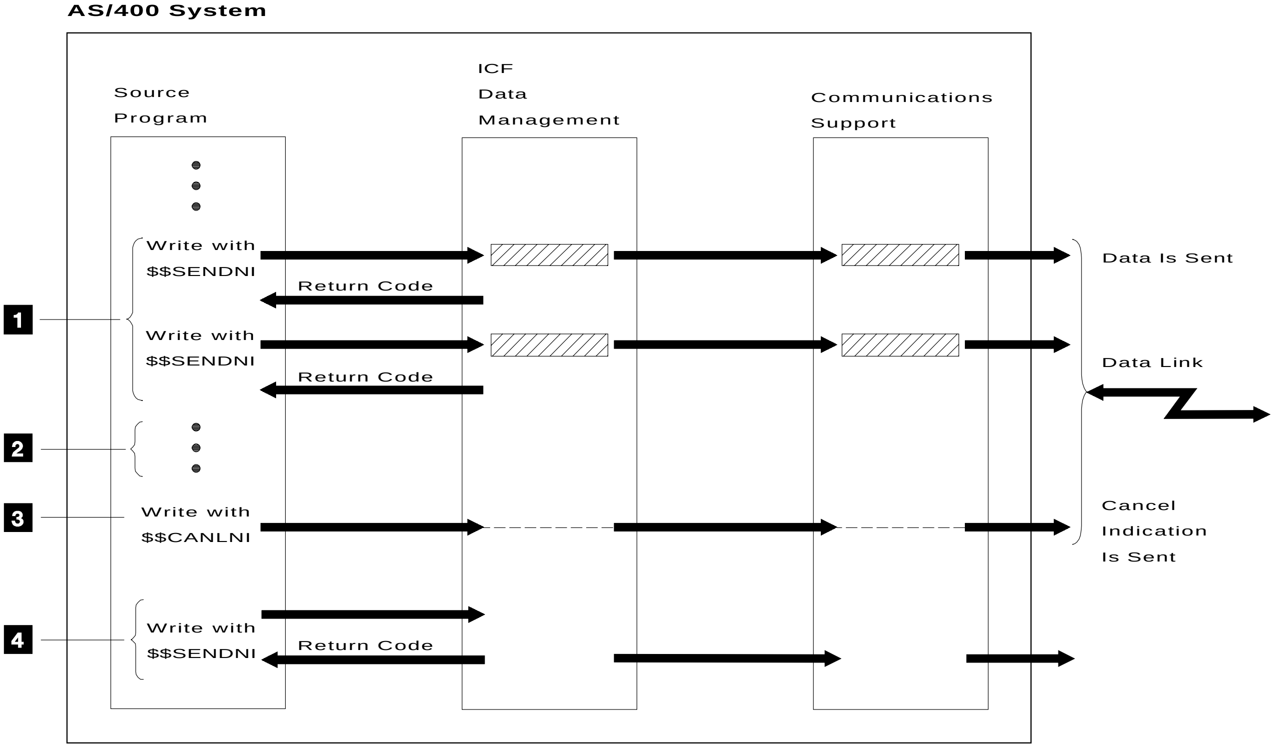
Figure 7-9. Fail RPG/400 Output Specification

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 7-10 shows how to use the $$CANLNI communications format when your program is sending data and detects an error.



RSLS673-3

Figure 7-10. 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.

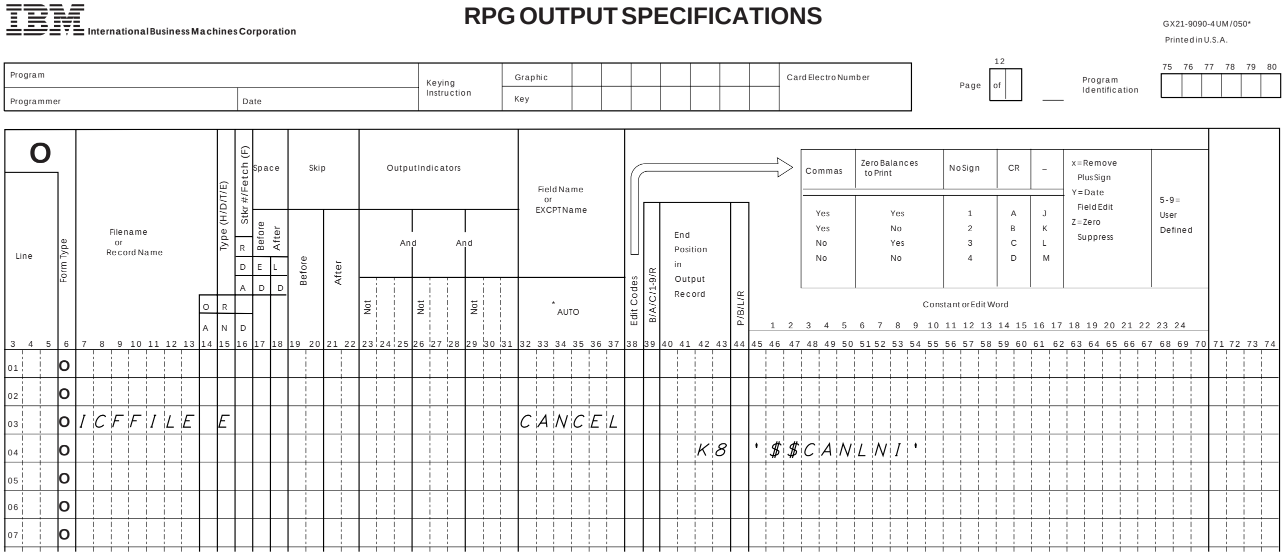
The following is an ILE C write statement example that cancels the current chain of records.

```
_RFILE *icffptr;              /* Pointer to the ICF file */

⋮
icffptr = _Ropen("ICFFILE","ab+ indicators=y riofb=y");

⋮
_Rformat(icffptr, "$$CANLNI");/* Set cancel w/no invite format */
_Rpgmdev(icffptr, "CM1");     /* Set default device to CM1 */

_Rwrite(icffptr, NULL, 0);    /* Send the cancel */
```

The following 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 7-11 is an example of an RPG/400 output specification to cancel the current chain of records.



IBM International Business Machines Corporation

RPG OUTPUT SPECIFICATIONS

GX21-9090-4 UM/050*
Printed in U.S.A.

```
03 OICFFILE E               CANCEL
04 O                                 K8 '$$CANLNI'
```

RSLS187-2

Figure 7-11. 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 7-12 on page 7-12 shows how to send a negative response with a sense code to the remote system.

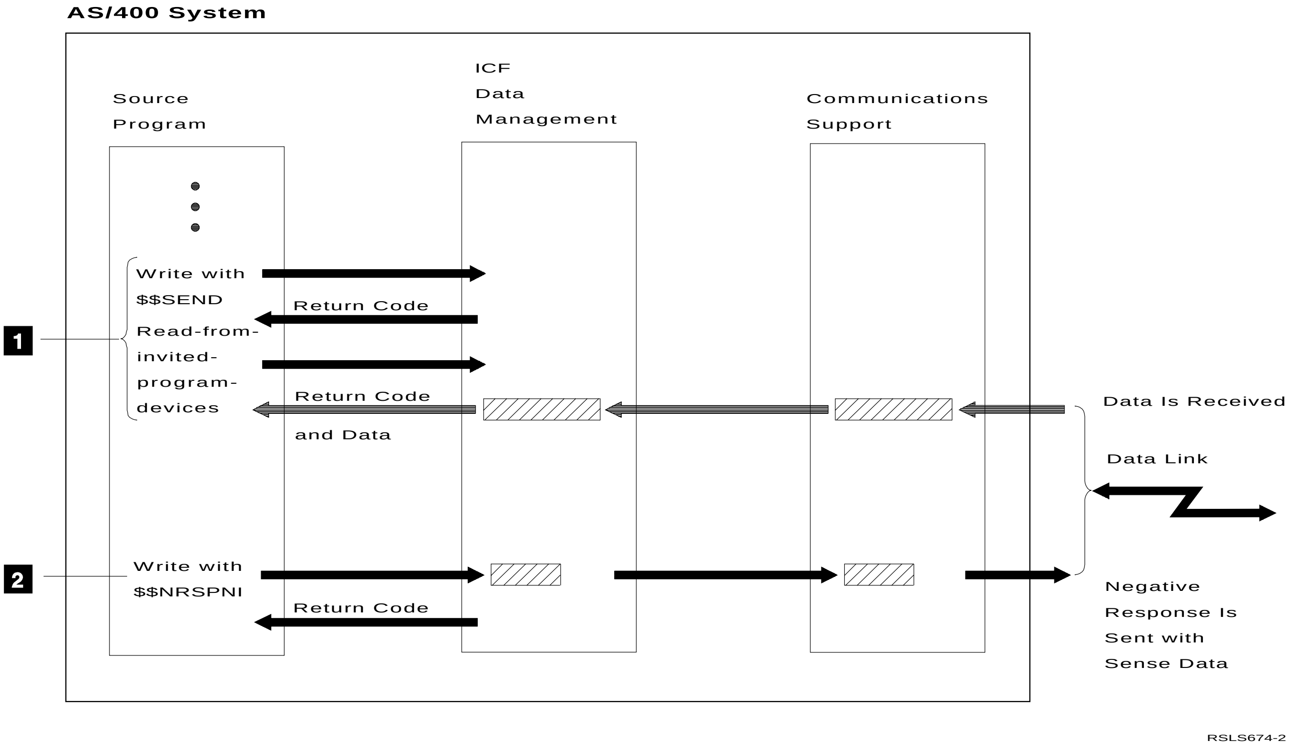


Figure 7-12. Using $$NRSP to Send an Error Condition

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 Figure 7-13 in the output buffer:

Figure 7-13. 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 book for the communications type you are using.

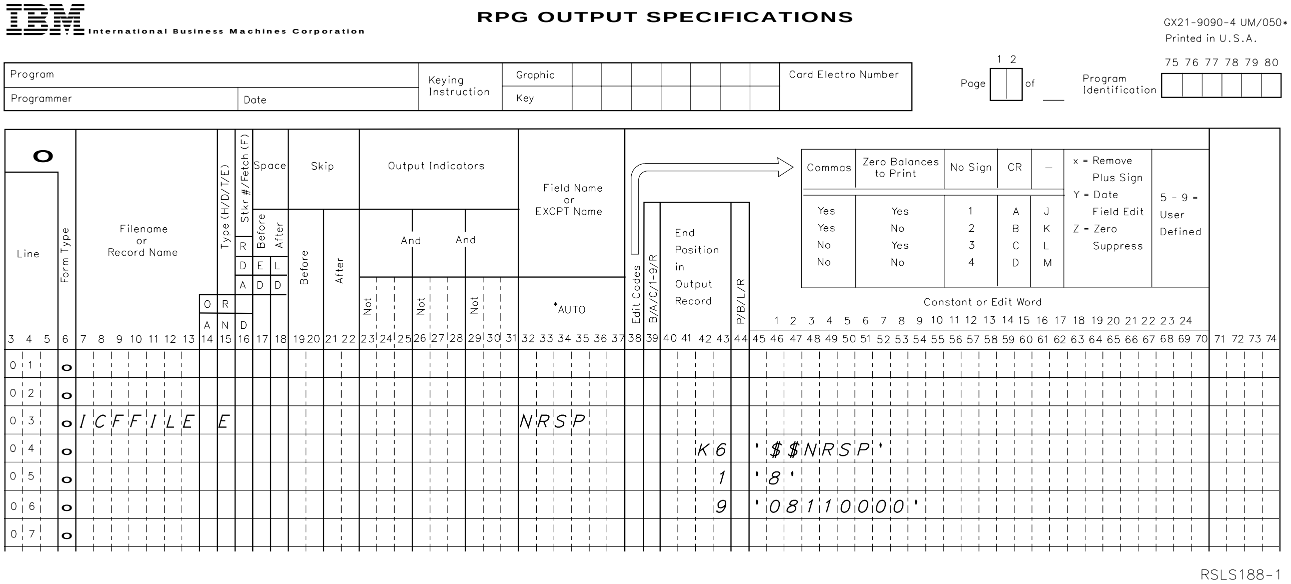
The following shows an ILE C write statement that sends a negative (–) response with invite function that includes the sense data 08110000.

```
struct {
    char length;
    char data??(8??);
} neg_resp_rec = {"8", "08110000"};

⋮
_RFILE *icffptr;            /* Pointer to the ICF file */

⋮
icffptr = _Ropen("ICFFILE","ab+ indicators=y riofb=y");

⋮
_RFORMAT(icffptr, "$$NRSP");/* Set negative response format */
_Rpgmdev(icffptr, "CM1");   /* Set default device to CM1 */

      /* Send negative response with sense data */
_Rwrite(icffptr, &neg_resp_rec, sizeof(neg_resp_rec));
```

The following 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 7-14 on page 7-13 shows the RPG/400 output specifications you can use to send a negative response that includes the sense data 08110000.


IBM International Business Machines Corporation
RPG OUTPUT SPECIFICATIONS
GX21-9090-4 UM/050*
Printed in U.S.A.
Program
Programmer
Date
Keying Instruction
Graphic
Key
Card Electro Number
Page
of
Program Identification
RSLS188-1

```
01 O
02 O
03 OICFFILE E                NRSP
04 O                                   K6 '$$NRSP'
05 O                                    1 '8'
06 O                                    9 '08110000'
07 O
```

Figure 7-14. 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.

Positive-Response

Use the $$POSRSP system-supplied format to send a positive response to the host system when you are using the SNUF protocol. Using this format can increase your application's performance when the host is waiting for a response. Refer to the *SNA Upline Facility Programming* book for more information.

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.

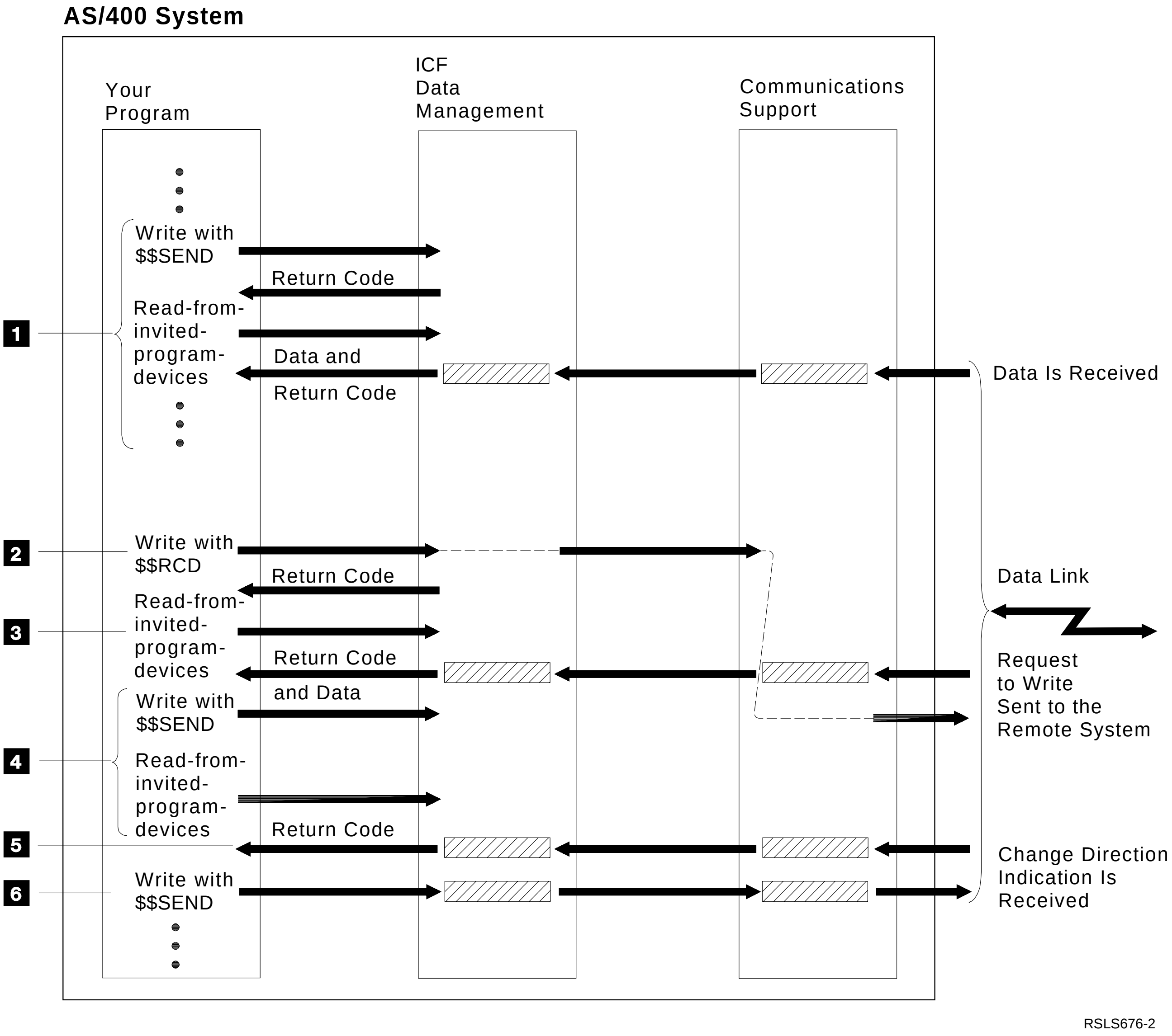


Figure 7-15. Using $$RCD to Request Write

Figure 7-15 shows how to use the $$RCD communications format to request permission to send.

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 Figure C-5 on page C-3 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 book.

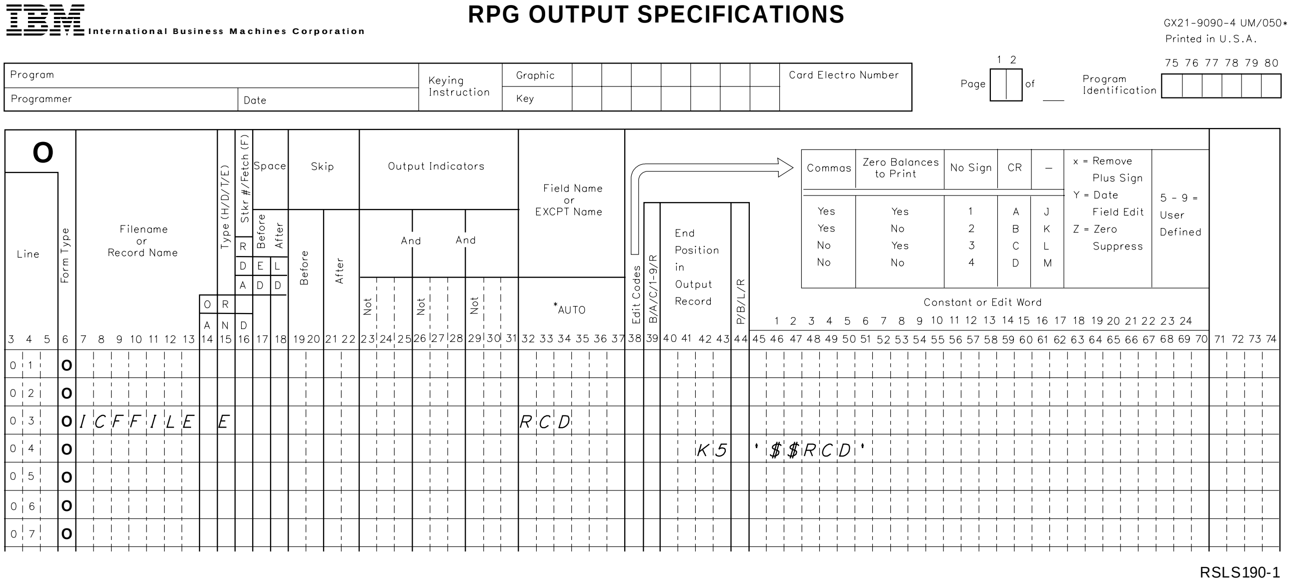
The following is an example of an ILE C write statement to request the remote system to stop sending data.

```
_RFILE *icffptr;              /* Pointer to the ICF file */

⋮
icffptr = _Ropen("ICFFILE","ab+ indicators=y riofb=y");

⋮
_Rformat(icffptr, "$$RCD");   /* Set request-to-write format */
_Rpgmdev(icffptr, "CM1");     /* Set default device to CM1 */

_Rwrite(icffptr, NULL, 0);    /* Send the request-to-write */
```

The following 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 7-16 is an example of an RPG/400 output specification to request that the remote system stop sending data.



IBM International Business Machines Corporation

RPG OUTPUT SPECIFICATIONS

GX21-9090-4 UM/050*
Printed in U.S.A.

Line	Form Type	Filename or Record Name (7–13)	Type (H/D/T/E) (15)	Field Name or EXCPT Name (32–37)	End Position in Output Record (40–43)	Constant or Edit Word (45–70)
01	O					
02	O					
03	O	ICFFILE	E	RCD		
04	O				K5	'$$RCD'
05	O					
06	O					
07	O					

RSLS190-1

Figure 7-16. Request-to-Write RPG/400 Output Specification

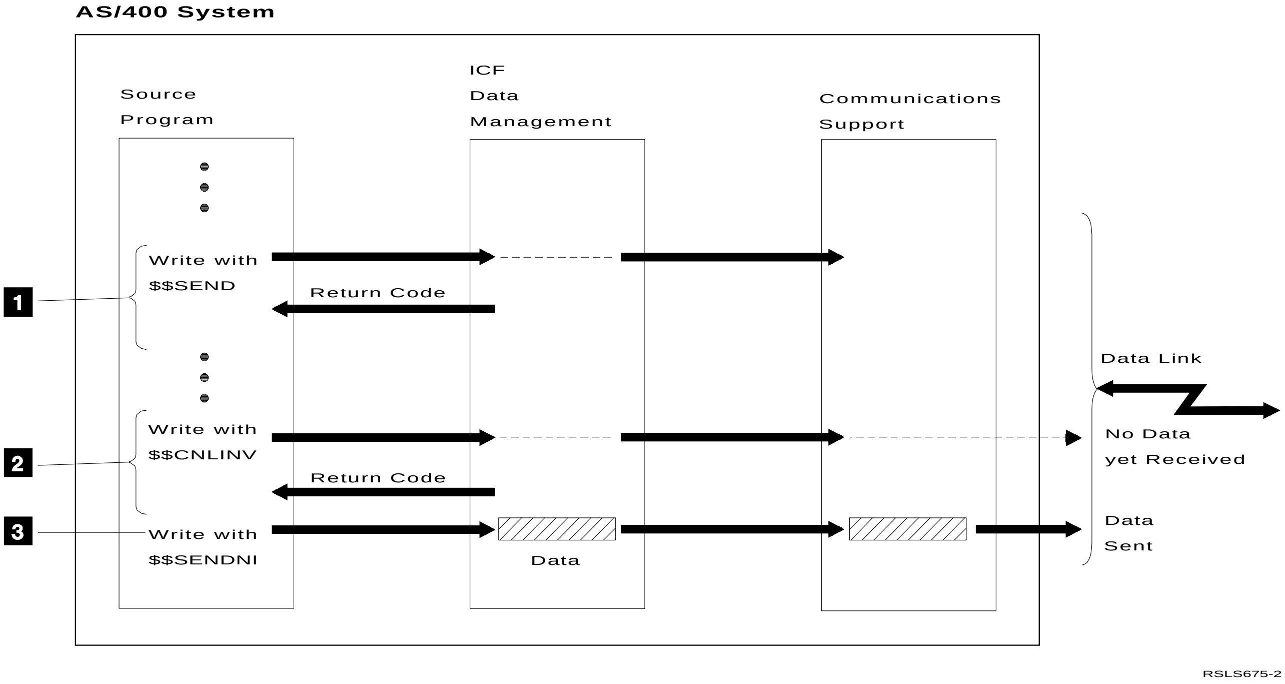


Figure 7-17. Using $$CNLINV to Cancel an Invite

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 7-17 shows how to use the $$CNLINV communications format to issue the cancel-invite function.

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 canceled.

3 If a successful return code is received, your program can send data.

No output fields are associated with the $$CNLINV communications format.

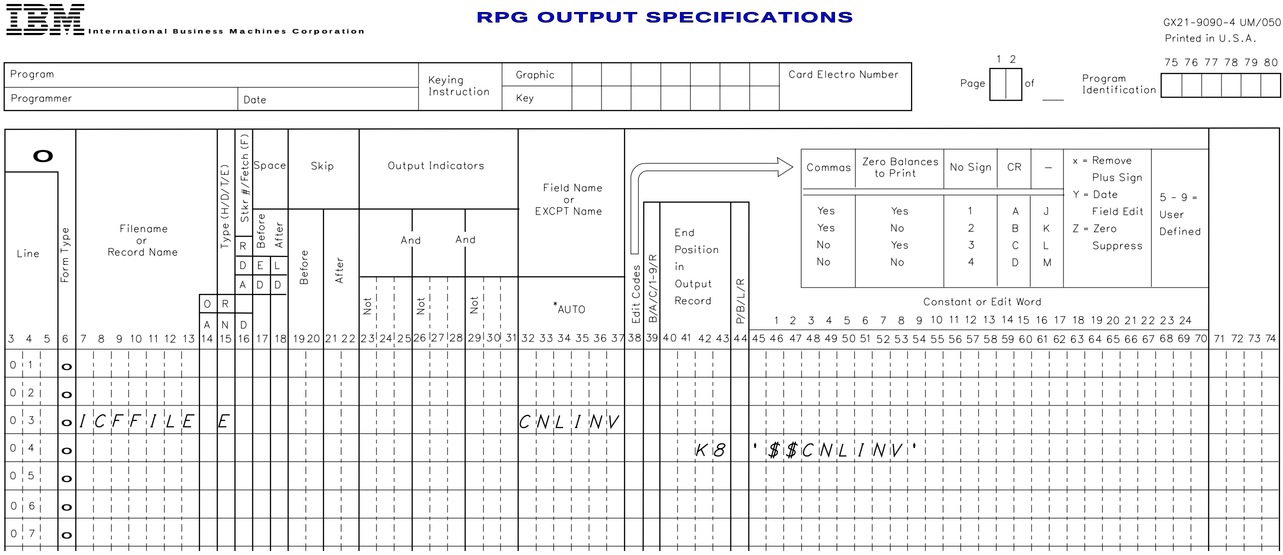
The following is an example of an ILE C write statement that issues a cancel-invite function to a program device that has not received input.

```
_RFILE *icffptr;               /* Pointer to the ICF file */

⋮
icffptr = _Ropen("ICFFILE","ab+ indicators=y riofb=y");

⋮
_Rformat(icffptr, "$$CNLINV");  /* Set cancel-invite format */
_Rpgmdev(icffptr, "CM1");       /* Set default device to CM1 */

_Rwrite(icffptr, NULL, 0);      /* Issue the cancel-invite */
```

The following 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 7-18 on page 7-17 is an example RPG/400 output specification to issue a cancel-invite to a program device that has not received input.



IBM International Business Machines Corporation

RPG OUTPUT SPECIFICATIONS

GX21-9090-4 UM/050* Printed in U.S.A.

```
Line  Form  Filename         Type   Field Name     End       Constant or Edit Word
      Type  or Record Name   (H/D/  or EXCPT Name  Position
                             T/E)                  in Output
                                                   Record
0 1   O
0 2   O
0 3   O     ICFFILE          E      CNLINV
0 4   O                                             K8       '$$CNLINV'
0 5   O
0 6   O
0 7   O
```

RSLS189-1

Figure 7-18. 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 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

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 book 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 7-19 on page 7-18 shows how to use the $$SENDET communications format to end the transaction.

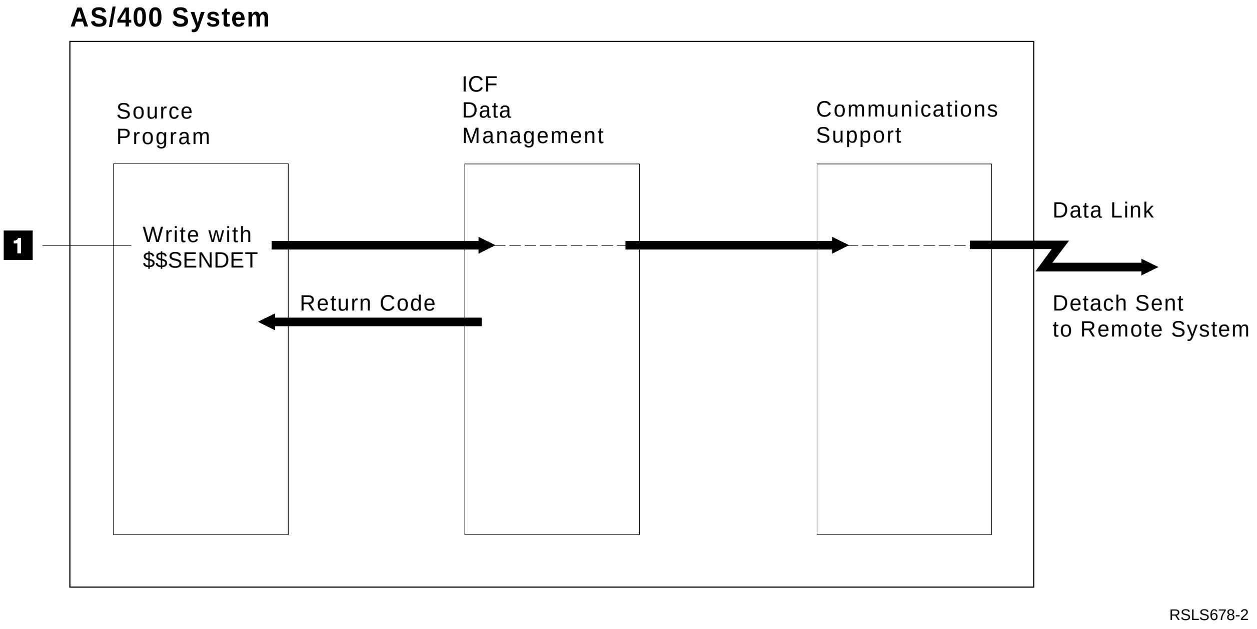


Figure 7-19. Ending the Communications Transaction

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 7-1 and “Sending Data” on page 7-4 for information about the output format of these functions.

The following is an example of an ILE C write statement to issue a detach.

```
struct {
    char data_length??(4??);
    char data??(80??);
} data_rec;

⋮
_RFILE *icffptr;               /* Pointer to the ICF file */

⋮
icffptr = _Ropen("ICFFILE","ab+ indicators=y riofb=y");

⋮
_Rformat(icffptr, "$$SENDET"); /* Set write-with-detach format */
_Rpgmdev(icffptr, "CM1");      /* Set default device to CM1 */

strncpy(data_rec.data_length, "0080", 4); /* Set record length */
_Rwrite(icffptr, &data_rec, sizeof(data_rec));/* Send detach */
```

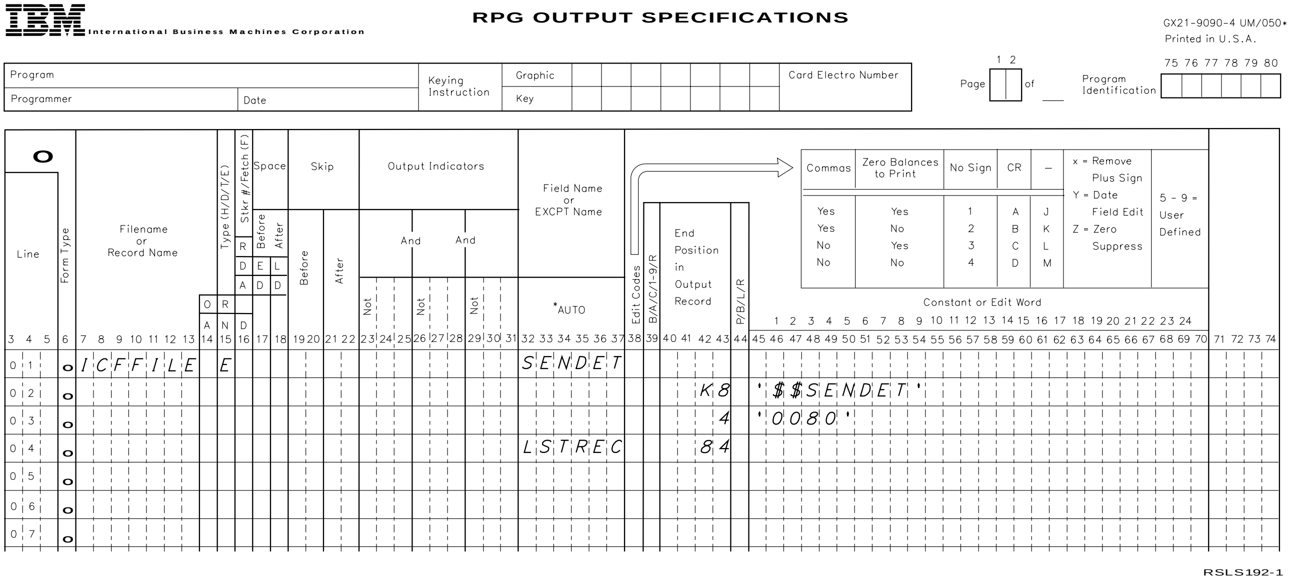
The following is an example of an COBOL/400 WRITE statement to issue a detach.

```
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.
```

If a detach function is issued by a target program, the EOS function is issued after the detach function has completed.

This should be done since neither the EOS, Detach, or any other ICF function can end the session (that is, cause an UNBIND to be sent).

Figure 7-20 on page 7-19 is an example of an RPG/400 output specification to issue a detach.



RPG OUTPUT SPECIFICATIONS

GX21-9090-4 UM/050*
Printed in U.S.A.

Line	Form Type	Filename or Record Name	Type (H/D/T/E)	Field Name or EXCPT Name	End Position in Output Record	Constant or Edit Word
01	O	ICFFILE	E	SENDET		
02	O				K8	'$$SENDET'
03	O				4	'0080'
04	O			LSTREC	84	
05	O					
06	O					
07	O					

RSLS192-1

Figure 7-20. Detach RPG/400 Output Specification

Ending the Communications Session

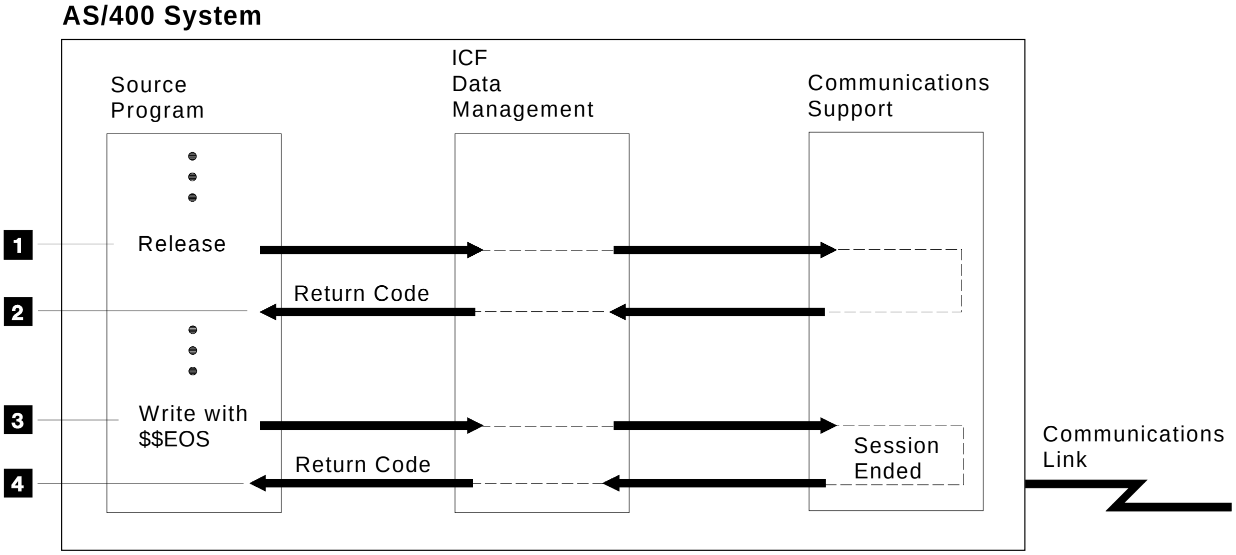
How the communications session is ended depends on whether your program or the remote system started the session.

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 7-21 shows how you can end the session by using the release operation and the end-of-session function.



RSLS677-2

Figure 7-21. 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 target program ends the transaction by the detach function, the session is ended implicitly. If the source program ends the transaction, the target program must issue an end-of-session function or go to the end of the job to end the session.

No output fields are associated with the $$EOS communications format.

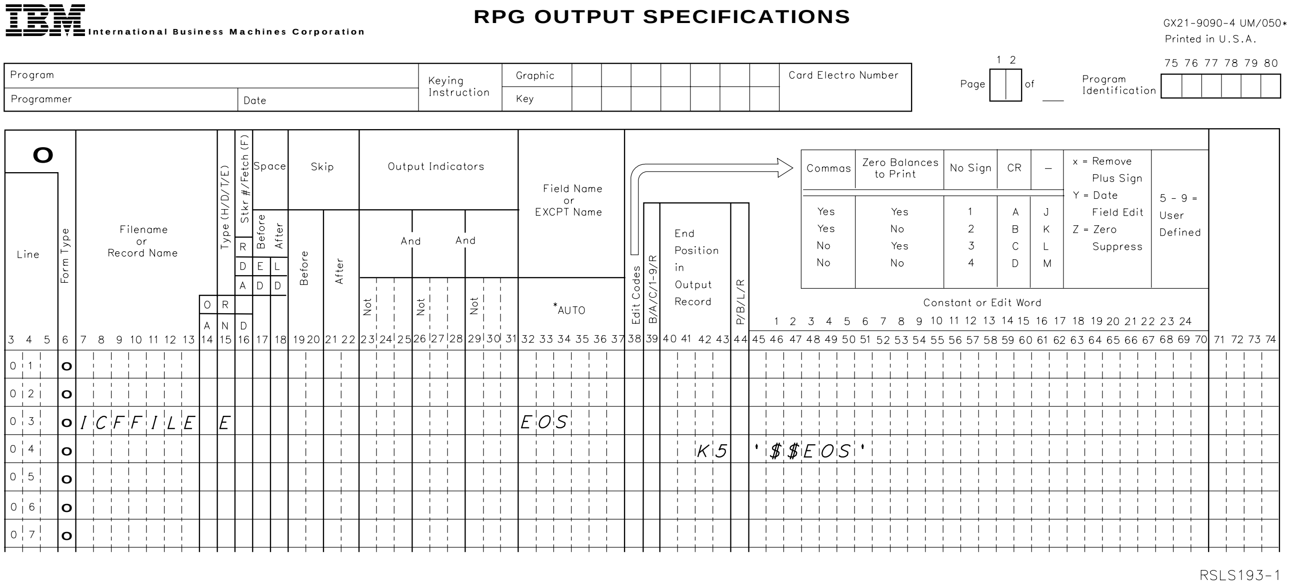
The following is an example of an ILE C write statement specifying the $$EOS format.

```
_RFILE *icffptr;              /* Pointer to the ICF file */

⋮
icffptr = _Ropen("ICFFILE","ab+ indicators=y riofb=y");

⋮
_Rformat(icffptr, "$$EOS");    /* Set end-of-session format */
_Rpgmdev(icffptr, "CM1");      /* Set default device to CM1 */

_Rwrite(icffptr, NULL, 0);     /* Send an end-of-session */
```

The following is an example of a COBOL/400 WRITE statement specifying the $$EOS format.

```
WRITE TRANSACTION-RECORD,
   FORMAT IS '$$EOS', TERMINAL IS ICF-PGMDEV.
```

Figure 7-22 is an example of an RPG/400 output specification to issue the end-of-session function using the $$EOS format.


IBM International Business Machines Corporation
RPG OUTPUT SPECIFICATIONS
GX21-9090-4 UM/050*
Printed in U.S.A.
Program
Programmer
Date
Keying Instruction
Graphic
Key
Card Electro Number
Page of
Program Identification
Line
Form Type
Filename or Record Name
Type (H/D/T/E)
Stkr #/Fetch (F)
Space
Skip
Output Indicators
Field Name or EXCPT Name
Edit Codes
B/A/C/1-9/R
End Position in Output Record
P/B/L/R
Constant or Edit Word
*AUTO
RSLS193-1

```
0 1 O
0 2 O
0 3 O ICFFILE  E          EOS
0 4 O                              K5  '$$EOS'
0 5 O
0 6 O
0 7 O
```

Figure 7-22. End-of-Session RPG/400 Output Specification

System-Supplied Format Support

Figure 7-23 shows which system-supplied formats are supported by each communications type.

Note: APPC support applies to APPC over TCP/IP, as well.

Figure 7-23 (Page 1 of 2). System-Supplied Format Support

System-Supplied Format	APPC	SNUF	BSCEL	Async	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

Figure 7-23 (Page 2 of 2). System-Supplied Format Support

System-Supplied Format	APPC	SNUF	BSCEL	Async	Intra- system	Finance	Retail
$$EVOKET	X	X	X	X	X		X
$$EVOKNI	X	X	X	X	X		X
$$FAIL	X	X	X	X	X	X	
$$NRSP		X			X	X	X
$$NRSPNI		X			X	X	X
$$POSRSP		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
$$SENDFM		X			X	X	X
$$SENDNF		X		X	X	X	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 Programming* book for more details.

Mapping System-Supplied Formats to DDS Keywords

Figure 7-24 maps system-supplied formats to DDS keywords.

Figure 7-24. 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
$$POSRSP	RSPCONFIRM
$$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 book 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).
- Irrecoverable error occurred. The session has been ended and the underlying communications support may no longer be active (80xx).
- Irrecoverable 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).

ICF Environment Considerations

You must understand the following considerations 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 communications:

- 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* book 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.

  Note: With a FORTRAN/400 program, a separate ICF file must be used for each ICF session. FORTRAN/400 programs can implicitly acquire a program device only when the ICF file is opened (ACQPGMDEV = program device).
- 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 communications:

- 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 an 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

The following input considerations apply when you write 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.
- For most communication types, 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

The following release, end-of-session, and close considerations apply when you write 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 an 832C return code.

  If the release operation is successful, the communications session ends.

  Note: The FORTRAN/400 language does not support a release operation. Instead, an end-of-session function or a close operation must be issued. Refer to “End-of-Session Considerations” and to “Close Considerations” for more information.

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.
- For conversations started using EVOKE with SYNLVL(*COMMIT) specified:
  - If EOS is issued after a TAKE_COMMIT_ indication has been received by the transaction program (TP), resynchronization processing is performed.
  - In all other cases, the EOS causes the logical unit of work (LUW) to be put into rollback required state. The TP must perform a rollback operation before working with any other resources involved in that LUW.

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* book 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.
- **Protected conversations** are conversations that use two-phase commit protocols. Protected conversations must commit the current logical unit of work before they can end normally. If an end-of-session function is issued before the logical unit of work is committed, APPC rolls back the logical unit of work.
- Application programs using protected conversations should detach the conversation before ending the session. The DETACH keyword issued with the TNSSYNLVL keyword and followed by a commit operation causes the system to commit all changes in the current logical unit of work and end the conversations. Ignoring this advice and issuing a write with EOS when there are still active protected conversations causes APPC to do the following.
  - Issue a DEALLOCATE_ABEND on those conversations.
  - Roll back the current logical unit of work.

Two-Phase Commit Considerations

The following should be considered when programming for two-phase commit.

Committing Resources

Your program requests that protected resources are committed by using the commit operation or the PRPCMT function.

Your program is notified that it has received a commit request from the remote program in the following ways:

- A major return code of 02 or 03 with minor return codes of 57, 58, or 59.
- The RCVTKCMT response indicator is set. The RCVTRNRND and RCVDETACH response indicators may also be set.

When your program receives a commit request, it must respond positively or negatively to the request as follows:

- To respond positively, do a commit operation.
- To respond negatively to the request, do one of the following:
  - Do a rollback operation.
  - Issue a fail function. This causes the logical unit of work to be rolled back if the partner issued a commit operation. Otherwise, if the partner issued a prepare-for-commit function, your program can attempt error recovery.
  - Abnormally end the transaction and session by issuing either an end-of-session function or a close operation.

Rolling Back Resources

Your program requests that protected resources are rolled back by using the rollback operation.

Your program is notified that it has received a rollback request from the remote program or that rollback is required because of an error on the conversation in the following ways.

- One of the following return codes is received.
  - 0054
  - 0254
  - 80F9, 80FA, 80FB
  - 81F0, 81F1, 81F2, 81F3, 81F4, 81F5
  - 83FB, 83FC, 83FD, 83FE, 83FF
- The RCVROLLB response indicator is set.

When your program receives a rollback request, it must respond with a rollback operation.

Exchanging Log Names

APPC uses a mechanism called exchange log name to negotiate the exact two-phase commit capabilities used for a protected conversation. The two systems at each end of the conversation exchange information about their level of two-phase commit support. Together, they decide which functions to use.

Exchange log name processing is performed when the system attempts to evoke its first protected conversation after communications have been established between the two systems. (The active session count between the two systems goes from 0 to 1.) The evoke is pended until the exchange log name processing has completed successfully.

Exchange log name processing brings up its own session to do the negotiation. You need to configure the mode description with one extra session that the system can use for exchange log name processing.

Performance

The following are performance considerations for two-phase commit processing.

- The first protected conversation evoke between two systems takes longer to complete because of the exchange log name processing that takes place between the systems.
- The user may experience slower response times due to the two-phase commit processing needed to process the commit and rollback operations. Commit and rollback operations are done for each transaction program in the two-phase commit transaction.
- The bigger the two-phase commit transaction and the greater the number of commits issued for each transaction, the slower the response time.
- If data integrity is critical to your application, you should use two-phase commit processing. The extra processing that is done to ensure data integrity slows the performance of applications that use two-phase commit processing.

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 change an IBM-supplied subsystem such as QBASE or QCMN, to receive and process program start requests.

Refer to the *Work Management* book 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* book.

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)
- Remove Communications Entry (RMVCMNE)
- Change Communications Entry (CHGCMNE)

You can specify a default user on the communications entry for a subsystem description on the DFTUSR parameter on the ADDCMNE command or CHGCMNE command. Refer to the *CL Reference* book 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 *Work Management* book 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 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 is not passed as part of the program start request (for example, *NONE specified on the SECURITY keyword), the system checks the communications entry in the subsystem handling the program start request to see if it 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.

If a program start request is received on the AS/400 system with an unqualified program name, the system uses the library list for the subsystem handling the program start request. The library list for the subsystem consists of the values from the QSYSLIBL and QUSRLIBL system values at the time the subsystem was started.

If a program start request is received on the AS/400 system with a qualified program name, the acceptable format varies by communication type. Most communication types (for example, asynchronous, BSCEL, and SNUF) separate the library and program as part of the program start request. Intrasystem communications allows the program name to be in the form *library/program*. APPC accepts either *library/program* or the architected form *program.library*.

Note: Program and library names on the AS/400 system are limited to 10 characters. If you are using BSCEL to start a remote program, the program and library name is limited to 8 characters.

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 8-7 shows a sample ICF communications environment.

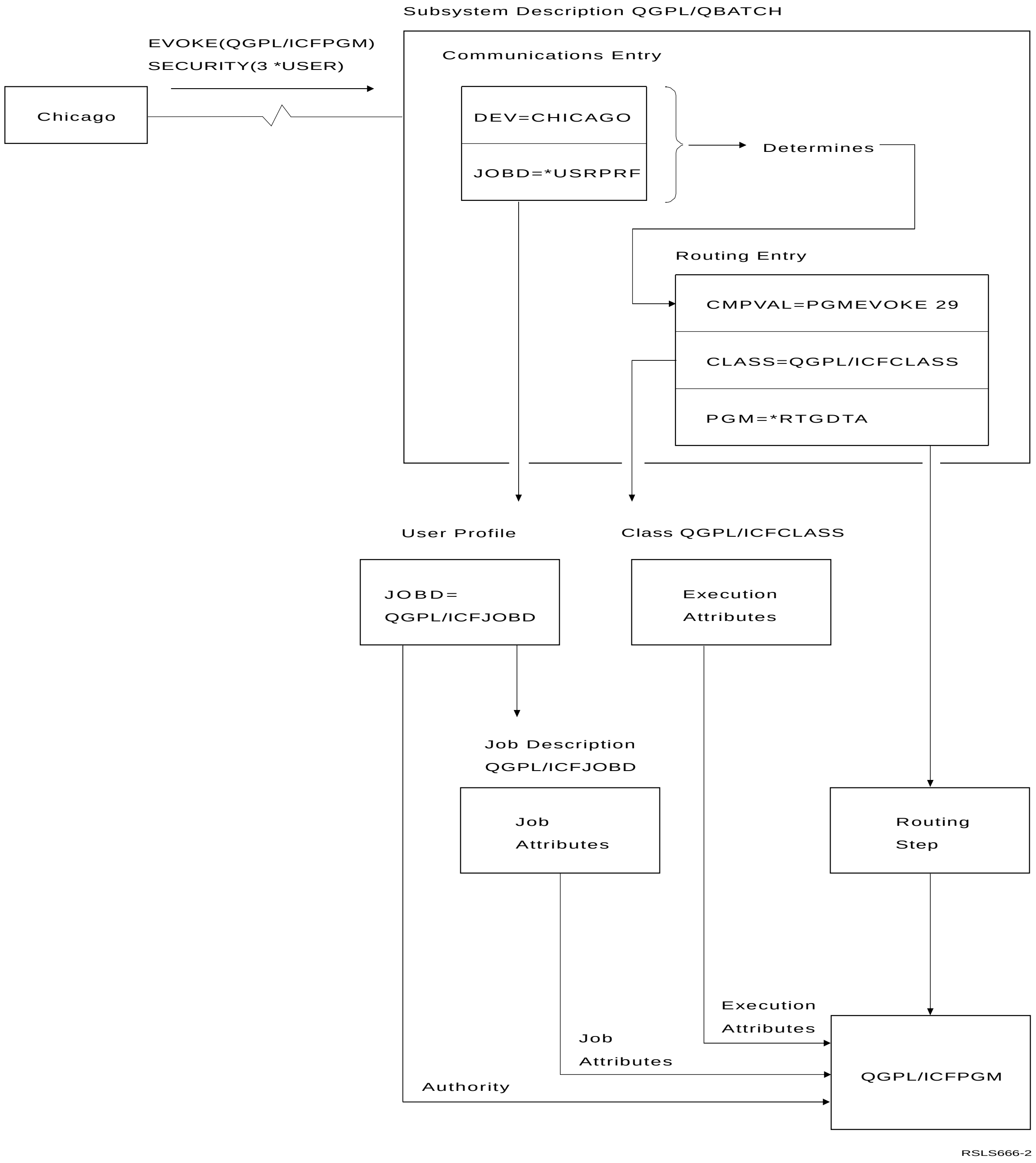


Figure 8-1. Sample ICF Communications Environment

Refer to the *Work Management* book 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 Management* book.

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.

Protected conversations must commit the current logical unit of work before they can end normally. If an end-of-session function is issued before the logical unit of work is committed, APPC rolls back the logical unit of work.

Application programs using protected conversations should detach the conversation before ending the session. The DETACH keyword issued with the TNSSYNLVL keyword and followed by a commit operation causes the system to commit all changes in the current logical unit of work and end the conversations. Ignoring this advice and issuing a write with EOS when there are still active protected conversations causes APPC to do the following.

- Issue a DEALLOCATE_ABEND on those conversations.
- Roll back the current logical unit of work.

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* book.

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) or the high level language Retrieve Data Area operation (for example, COBOL ACCEPT) 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.
- Your application should check for an 8209 return code after completion of the acquire to the requesting program device to determine if the prestarted job is being canceled.

See Figure 8-2 on page 8-10 for a sample prestart job program. For information about sharing database files in the same job and across jobs, see the *DB2 for AS/400 Database Programming* book.

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 when you create your prestart jobs program. In other words, for the ILE C, ILE COBOL, or ILE RPG programming language, specify USRPRF(*OWNER) on the CRTPGM, CRTSRVPGM, CRTBNDC, CRTBNDRPG, or CRTBNDCBL command. For the COBOL/400 or RPG/400 programming language, specify USRPRF(*OWNER) on the CRTRPGPGM or CRTCBLPGM command.
- 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 Jobs Program

Figure 8-2 on page 8-10 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 acquiring the requesting program device, an accept is done to retrieve the program initialization parameters, and a CL program is called 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.

```
Program  . . . . . . . . . . . . . . :   PJPGM
  Library  . . . . . . . . . . . . . :     QNETUSER
Source file  . . . . . . . . . . . . :   QCBLSRC
  Library  . . . . . . . . . . . . . :     QNETUSER
Source member  . . . . . . . . . . . :   PJPGM     03/08/90 13:04:36
Generation severity level  . . . . . :   29
Text 'description' . . . . . . . . . :   COBOL program for prestart job example in C.P.G.
Source listing options . . . . . . . :   *SOURCE
Generation options . . . . . . . . . :   *NONE
Message limit:
  Number of messages . . . . . . . . :   *NOMAX
  Message limit severity . . . . . . :   29
Print file . . . . . . . . . . . . . :   QSYSPRT
  Library  . . . . . . . . . . . . . :     *LIBL
FIPS flagging  . . . . . . . . . . . :   *NOFIPS *NOSEG *NODEB *NOOBSOLETE
SAA flagging . . . . . . . . . . . . :   *NOFLAG
Flagging severity  . . . . . . . . . :   0
Replace program  . . . . . . . . . . :   *YES
Target release . . . . . . . . . . . :   *CURRENT
User profile . . . . . . . . . . . . :   *USER
Authority  . . . . . . . . . . . . . :   *LIBCRTAUT
Compiler . . . . . . . . . . . . . . :   IBM AS/400 COBOL/400

    1  000010 IDENTIFICATION DIVISION.
    2  000030 PROGRAM-ID.               PJPGM.                                                        03/07/90
       000050********************************************************************                     03/07/90
       000060*  THIS IS A PRESTART JOB TARGET PROGRAM                           *                     03/07/90
       000070*  THIS PROGRAM WILL LOOP FOREVER                                  *                     03/07/90
       000180********************************************************************                     03/07/90
    3  000200 ENVIRONMENT DIVISION.
    4  000220 CONFIGURATION SECTION.
    5  000240 SOURCE-COMPUTER. IBM-AS400.                                                             03/07/90
    6  000250 OBJECT-COMPUTER. IBM-AS400.                                                             03/07/90
    7  000260 SPECIAL-NAMES.   I-O-FEEDBACK IS IO-FEEDBACK
    8  000270                  OPEN-FEEDBACK IS OPEN-FBA                                              03/08/90
    9  000271                  PIP-DATA IS PIP-PARM.                                                  03/08/90
       000280
   10  000290 INPUT-OUTPUT SECTION.
   11  000310 FILE-CONTROL.
       000311********************************************************************                     03/07/90
       000312*                 F I L E   S P E C I F I C A T I O N S            *                     03/07/90
       000313*    ICFFILE : ICF FILE                                            *                     03/07/90
       000314*    QSYSPRT : PRINTER FILE                                        *                     03/07/90
       000315********************************************************************                     03/07/90
   12  000330     SELECT ICFFILE ASSIGN TO WORKSTATION-PJDDS-SI                                       03/07/90
   13  000340         ORGANIZATION IS TRANSACTION
   14  000360         FILE STATUS IS STATUS-IND MAJ-MIN.
   15  000370     SELECT QPRINT ASSIGN TO PRINTER-QSYSPRT.                                            03/07/90
```

Figure 8-2 (Part 1 of 3). COBOL/400 Coding for a Prestart Job Program

```
16  000420 DATA DIVISION.
17  000440 FILE SECTION.
18  000500 FD  ICFFILE                                                                             03/07/90
19  000510     LABEL RECORDS ARE STANDARD.
20  000520 01  ICFREC.                                                                             03/07/90
21  000530     COPY DDS-ALL-FORMATS-I-O OF PJDDS.                                                  03/07/90
22 +000001       05  PJDDS-RECORD PIC X(4101).                                          <-ALL-FMTS
   +000002*    I-O FORMAT:INPFMT     FROM FILE PJDDS      OF LIBRARY QNETUSER           <-ALL-FMTS
   +000003*                                                                             <-ALL-FMTS
23 +000004       05  INPFMT        REDEFINES PJDDS-RECORD.                              <-ALL-FMTS
24 +000005           06 INPDATA               PIC X(4096).                              <-ALL-FMTS
   +000006*    I-O FORMAT:DETACH     FROM FILE PJDDS      OF LIBRARY QNETUSER           <-ALL-FMTS
   +000007*                                                                             <-ALL-FMTS
   +000008*      05  DETACH        REDEFINES PJDDS-RECORD.                              <-ALL-FMTS
   +000009*    I-O FORMAT:EOS        FROM FILE PJDDS      OF LIBRARY QNETUSER           <-ALL-FMTS
   +000010*                                                                             <-ALL-FMTS
   +000011*      05  EOS           REDEFINES PJDDS-RECORD.                              <-ALL-FMTS
   +000012*    I-O FORMAT:INVITE     FROM FILE PJDDS      OF LIBRARY QNETUSER           <-ALL-FMTS
   +000013*                                                                             <-ALL-FMTS
   +000014*      05  INVITE        REDEFINES PJDDS-RECORD.                              <-ALL-FMTS
   +000015*    I-O FORMAT:FAIL       FROM FILE PJDDS      OF LIBRARY QNETUSER           <-ALL-FMTS
   +000016*                                                                             <-ALL-FMTS
   +000017*      05  FAIL          REDEFINES PJDDS-RECORD.                              <-ALL-FMTS
   +000018*    I-O FORMAT:IOFMT      FROM FILE PJDDS      OF LIBRARY QNETUSER           <-ALL-FMTS
   +000019*                                                                             <-ALL-FMTS
25 +000020       05  IOFMT         REDEFINES PJDDS-RECORD.                              <-ALL-FMTS
26 +000021           06 IODATA                PIC X(4096).                              <-ALL-FMTS
27 +000022           06 OUTLEN                PIC S9(5).                                <-ALL-FMTS
28  000590 FD  QPRINT                                                                              03/07/90
29  000600     LABEL RECORDS ARE OMITTED.                                                          03/07/90
30  000610 01  PRINTREC.                                                                           03/07/90
31  000620     05 PRTNOTE                       PIC X(132).                                        03/07/90
32  000640 WORKING-STORAGE SECTION.
33  000660 77 STATUS-IND                        PIC X(2).                                          03/07/90
34  000661 77 PIP                               PIC X(6).                                          03/08/90
35  000662 01 MAJ-MIN.                                                                             03/07/90
36  000663    05  MAJ                           PIC X(2).                                          03/07/90
37  000664    05  MIN                           PIC X(2).                                          03/07/90
38  000840 01 IO-FEEDBACK.                                                                         03/07/90
39  000850    05 FILLER                         PIC X(149).                                        03/07/90
40  000860    05 ACTUAL-LEN                     PIC 9(5) COMP-4.                                   03/07/90
    000870*                                                                                        03/07/90
41  001090 PROCEDURE DIVISION.                                                                     03/08/90
    001091                                                                                         03/08/90
    001101*                                                                                        03/07/90
    001450 START-PROGRAM-PARAGRAPH.
    001451********************************************************************                     03/07/90
    001452* OPEN ICF FILE AND PRINTER FILE                                   *                     03/07/90
    001454********************************************************************                     03/07/90
42  001470     OPEN OUTPUT QPRINT.                                                                 03/07/90
43  001471     OPEN I-O    ICFFILE.                                                                03/07/90
    001480 MAIN-LOOP.                                                                              03/07/90
44  001481     MOVE "WAITING FOR TRANSACTION" TO PRTNOTE.                                          03/07/90
45  001482     WRITE PRINTREC.                                                                     03/07/90
    001483********************************************************************                     03/07/90
    001484* ACQUIRING THE REQUESTER PROGRAM DEVICE CAUSES THIS PRESTART      *                     03/07/90
    001485* JOB PROGRAM TO WAIT FOR A PROGRAM START REQUEST.                 *                     03/07/90
    001486********************************************************************                     03/07/90
46  001590     ACQUIRE "REQDEVICE " FOR ICFFILE.                                                   03/07/90
    001591********************************************************************                     03/07/90
    001592* IF THIS PRESTART JOB IS BEING ENDED WITH THE CONTROLLED OPTION,  *                     03/07/90
    001593* PERFORM END-OF-JOB PROCESSING AND EXIT.                          *                     03/07/90
    001594********************************************************************                     03/07/90
47  001600     IF MAJ-MIN = "8209" GO TO END-JOB.                                                  03/07/90
49  001610     MOVE "TRANSACTION ATTACHED" TO PRTNOTE.                                             03/07/90
50  001620     WRITE PRINTREC.                                                                     03/07/90
    001621********************************************************************                     03/07/90
    001622* CALL A CL PROGRAM TO USE SOME JOB ATTRIBUTES FROM THE PROGRAM    *                     03/07/90
    001624* START REQUEST USER PROFILE (CHGPJ).                              *                     03/07/90
    001625********************************************************************                     03/07/90
51  001630     CALL "CLPGM".                                                                       03/07/90
```

Figure 8-2 (Part 2 of 3). COBOL/400 Coding for a Prestart Job Program

```
        001631********************************************************************                     03/08/90
        001632* MOVE THE PIP DATA FROM THE PIP DATA AREA INTO IDENTIFIER PIP.   *                     03/08/90
        001634********************************************************************                     03/08/90
    52  001635     ACCEPT PIP FROM PIP-PARM END-ACCEPT.                                                03/08/90
        001636********************************************************************                     03/07/90
        001637* MAIN BODY OF THE PROGRAM, THAT DOES THE COMMUNICATIONS I/O       *                     03/07/90
        001638* WITH THE SOURCE PROGRAM, SHOULD BE PLACED HERE.                  *                     03/07/90
        001639********************************************************************                     03/07/90
        001640********************************************************************                     03/07/90
        001641* WHEN THE MAIN BODY OF THE PROGRAM IS DONE PROCESSING, END        *                     03/07/90
        001642* THE COMMUNICATIONS SESSION AND LOOP BACK TO WAIT FOR THE NEXT    *                     03/07/90
        001643* PROGRAM START REQUEST TO COME IN.                                *                     03/07/90
        001644********************************************************************                     03/07/90
    53  001645     WRITE ICFREC FORMAT IS "EOS".                                                       03/08/90
    54  001650     GO TO MAIN-LOOP.                                                                    03/07/90
        001660 END-JOB.                                                                                03/07/90
    55  001670     MOVE "PRESTART JOB BEING ENDED CONTROLLED" TO PRTNOTE.                              03/07/90
    56  001680     WRITE PRINTREC.                                                                     03/07/90
    57  001690     CLOSE ICFFILE.                                                                      03/07/90
    58  001691     CLOSE QPRINT.                                                                       03/07/90
    59  001692     STOP RUN.                                                                           03/07/90
        001693 MAIN-EXIT.                                                                              03/07/90
        001700     EXIT.                                                                               03/07/90
                            * * * * *   E N D   O F   S O U R C E   * * * * *

*   21  MSGID: LBL0600  SEVERITY: 10  SEQNBR:  000530
        Message . . . . :   No INPUT fields found for format DETACH.
*   21  MSGID: LBL0600  SEVERITY: 10  SEQNBR:  000530
        Message . . . . :   No INPUT fields found for format EOS.
*   21  MSGID: LBL0600  SEVERITY: 10  SEQNBR:  000530
        Message . . . . :   No INPUT fields found for format INVITE.
*   21  MSGID: LBL0600  SEVERITY: 10  SEQNBR:  000530
        Message . . . . :   No INPUT fields found for format FAIL.
                        * * * * *   E N D   O F   M E S S A G E S   * * * * *
                                          Message Summary
  Total    Info(0-4)    Warning(5-19)    Error(20-29)    Severe(30-39)    Terminal(40-99)
      4          0            4               0               0                 0
 Source records read . . . . . . . . :   97
 Copy records read . . . . . . . . . :   22
 Copy members processed  . . . . . . :   1
 Sequence errors . . . . . . . . . . :   0
 Highest severity message issued . . :   10
  LBL0901 00  Program PJPGM created in library QNETUSER.
                       * * * * *   E N D   O F   C O M P I L A T I O N   * * * * *
```

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* book.

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 book 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* book.

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 book for the communications type you are using.

Chapter 9. ILE C Communications Applications

Previous chapters in this book describe the functions provided by ICF. This chapter introduces you to the ILE C 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 book 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 ILE C Interface

Before you write an ILE C communications application, you must understand the high-level language interface provided by ILE C programming language.

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.

Figure 9-1 briefly introduces the ILE C functions you use in the communications portion of your program.

Note: ILE C functions are case sensitive.

Refer to the *ILE C/400 Programmer's Guide* for details on the syntax and function of each operation.

Figure 9-1. ILE C Function

ICF Operation	ILE C Function	Function
Open	fopen, _Ropen	Opens the ICF file
Acquire	_Racquire	Establishes a session
Get-Attributes	_Rdevatr	Gets the attributes of a session
Read	fread, _Rreadn	Receives data from a specific program device
Read-from-invited-program-devices	_Rreadindv	Receives data from any invited program device[1]
Write	fwrite, _Rwrite	Performs many of the ICF communications functions within a session
Write/Read	_Rwriterd	Performs the specified function and then receives data from the remote system.
Release	_Rrelease	Releases the session
Close	fclose, _Rclose	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).

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.

Error Handling

ILE C programming language provides an external variable, *Maj_Min_rc*, in the header file <stdio.h>, which contains the ICF major and minor codes after a read or write operation. The major and minor codes can also be obtained from the I/O feedback area.

A global variable, *ERRNO*, is defined in the header file <errno.h>. Two *ERRNO* values indicate that an exception has occurred:

EIOERROR
An I/O error has occurred that is not recoverable.

EIORECERR
A recoverable I/O error has occurred.

Accessing the Feedback Areas

Your program can obtain a copy of the open feedback area by using the _Ropnfbk routine. The _Riofbk routine can be used to obtain a copy of the I/O feedback area.

Transaction Flow of the Multiple-Session Inquiry: The program shown in Figure 9-2 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.

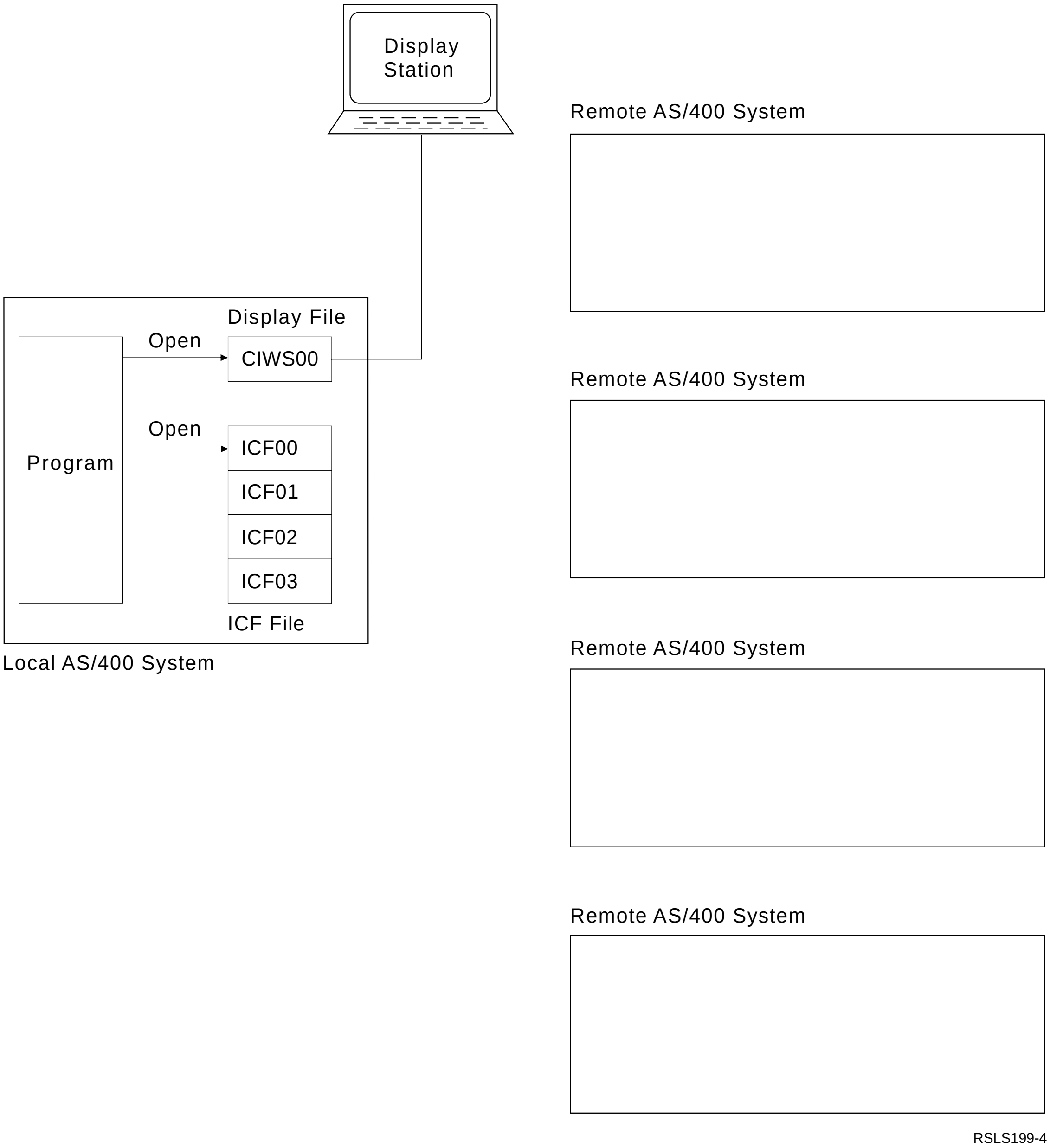


Figure 9-2. Program Starts at Display Station

All other program devices are explicitly acquired by the program, as shown in Figure 9-3.

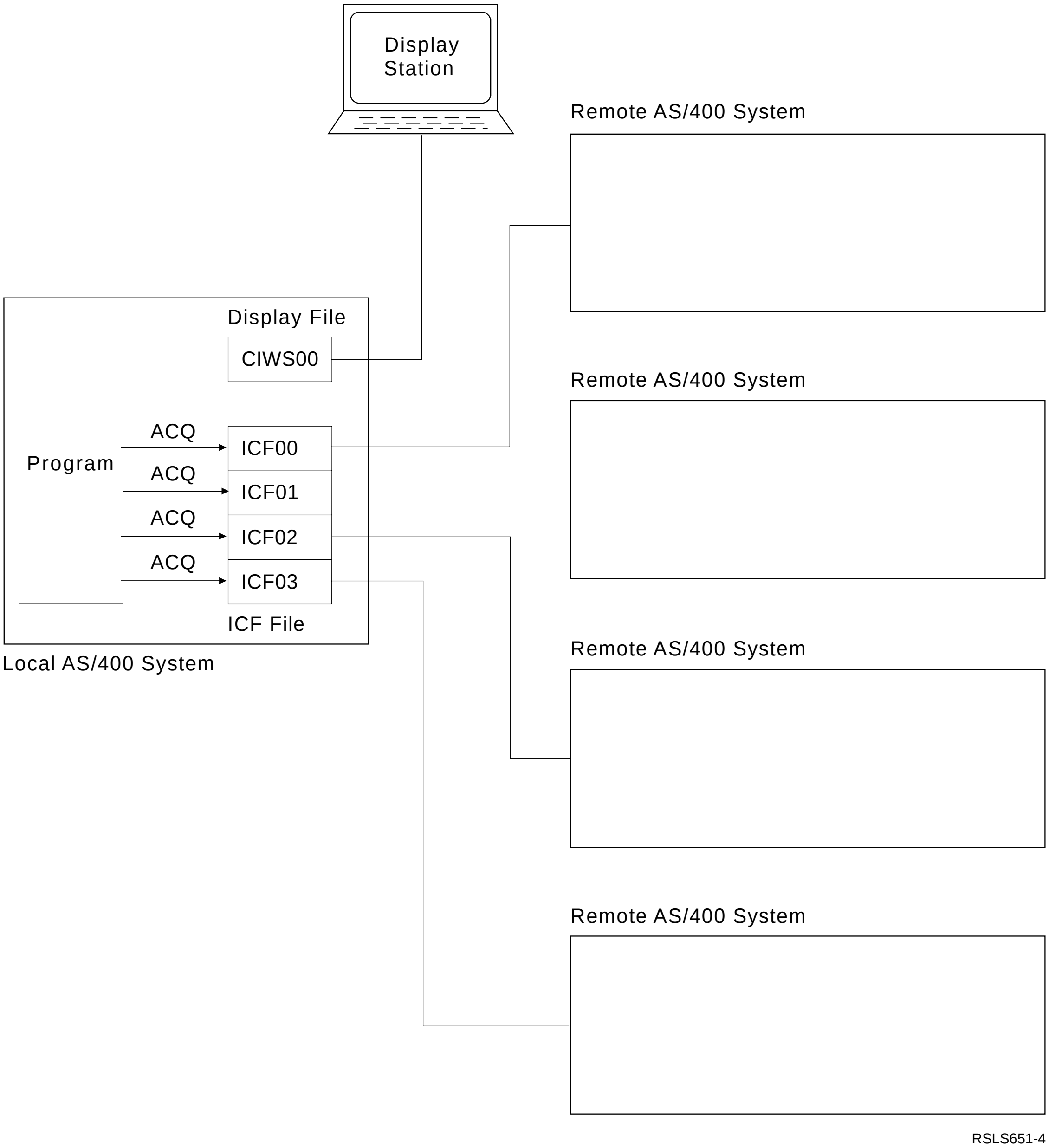


Figure 9-3. Program Devices Explicitly Acquired

All target programs are started with the evoke, as shown in Figure 9-4.

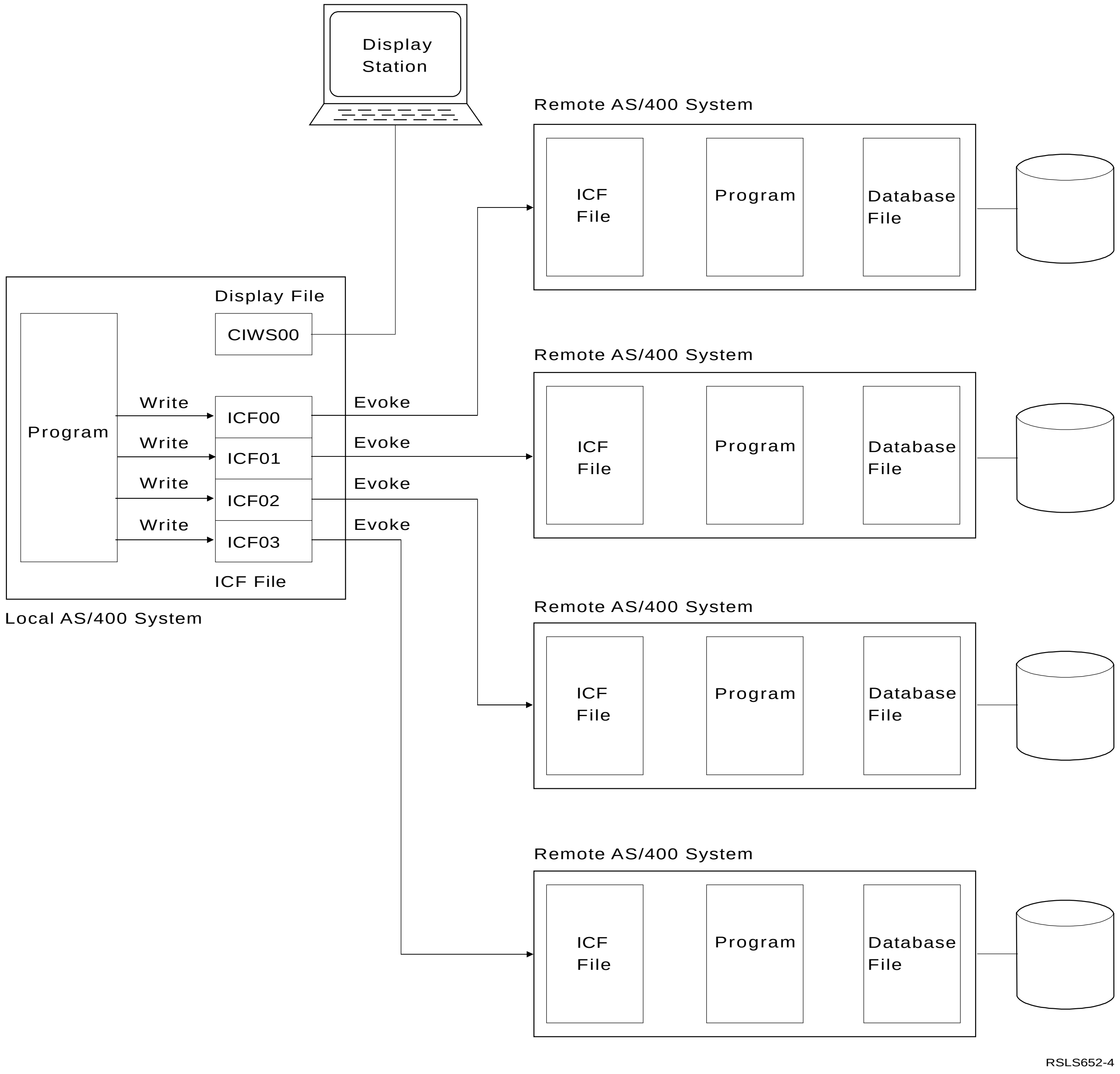


Figure 9-4. Evoke Starts Target Programs

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 ILE C 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 9-5.

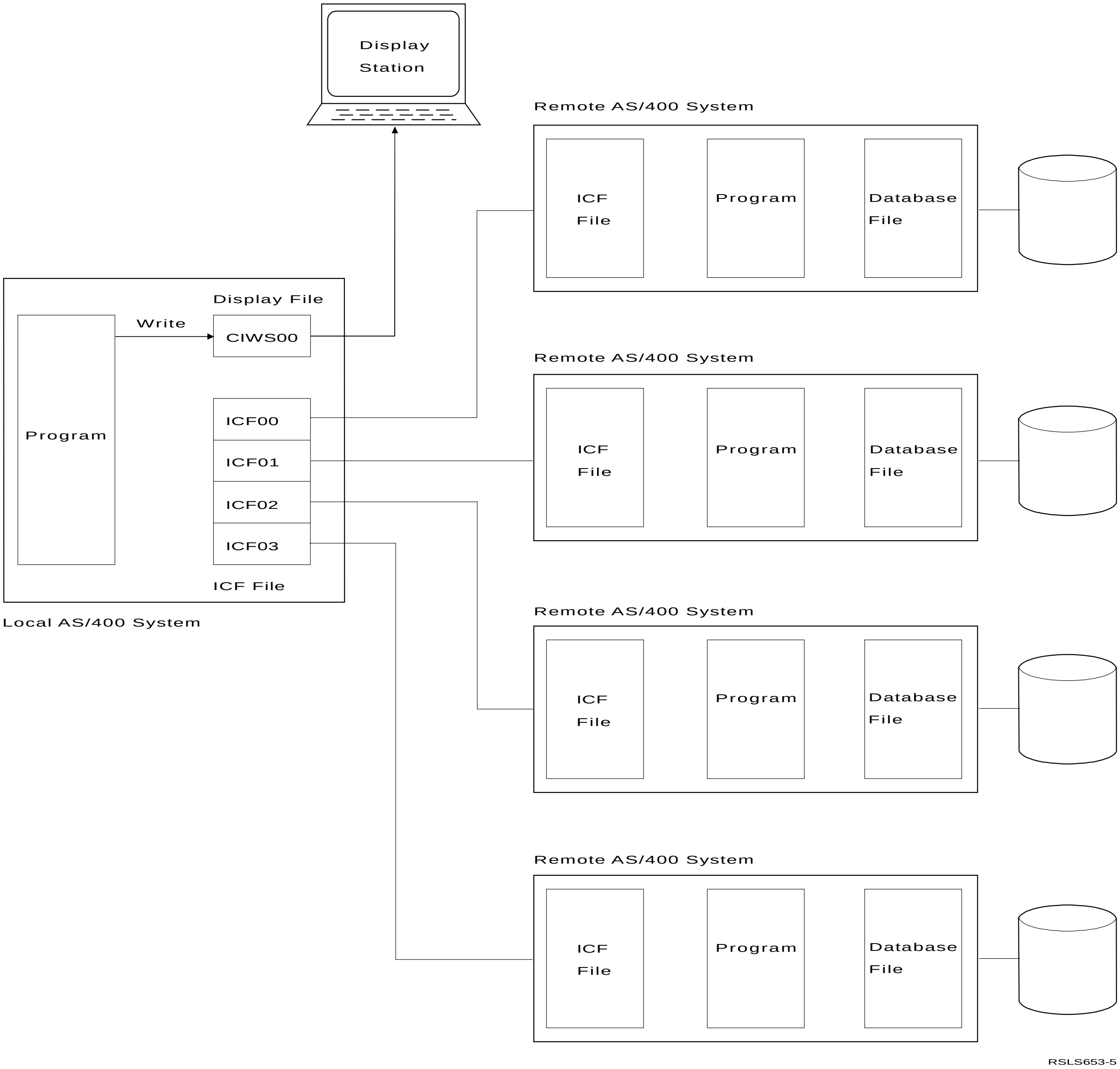


Figure 9-5. Main Menu Written to Display Station

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 9-6.

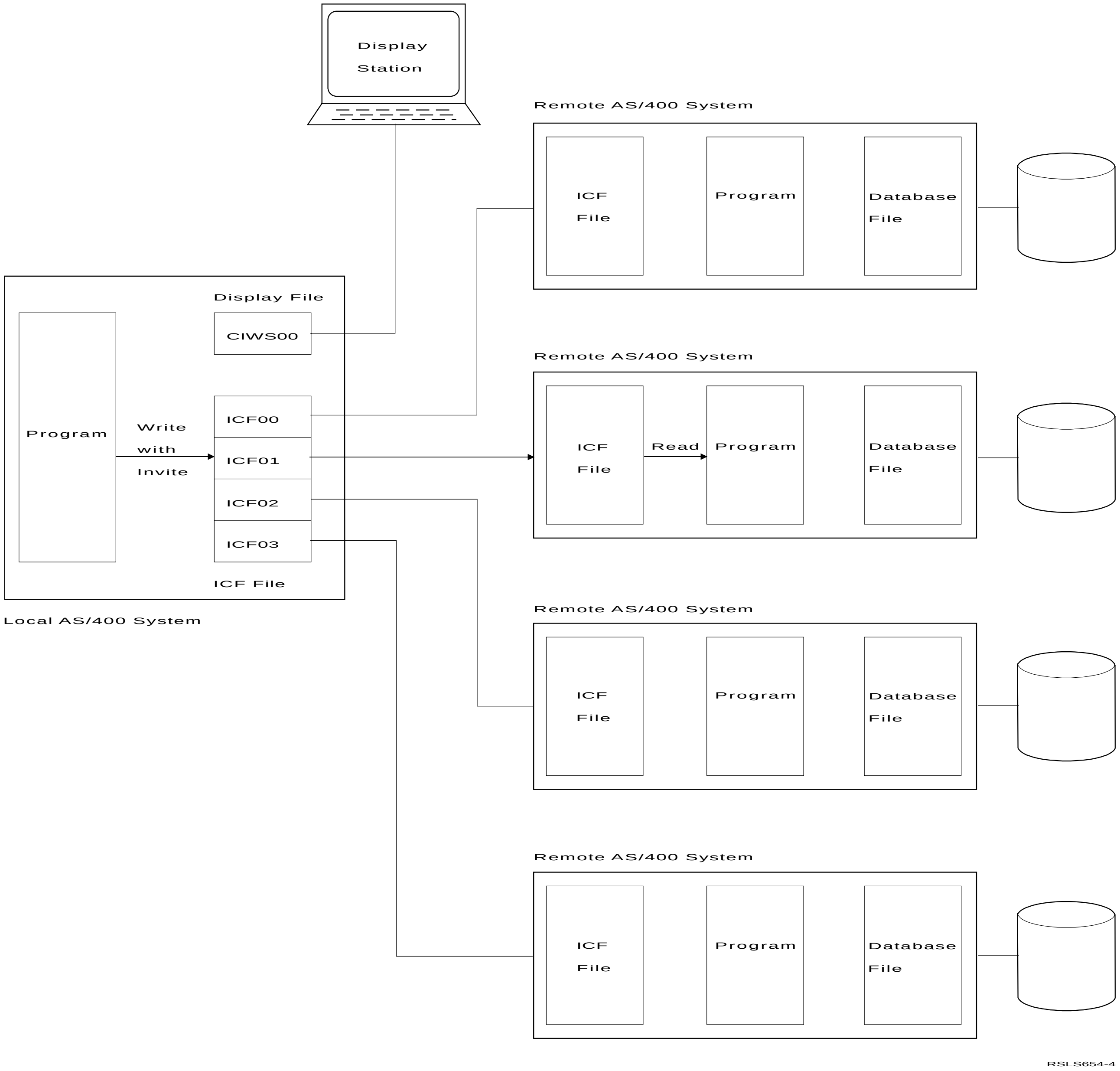


Figure 9-6. Program Sends Inquiry Request to Remote System

The target program responds to the inquiry by sending a reply, as shown in Figure 9-7.

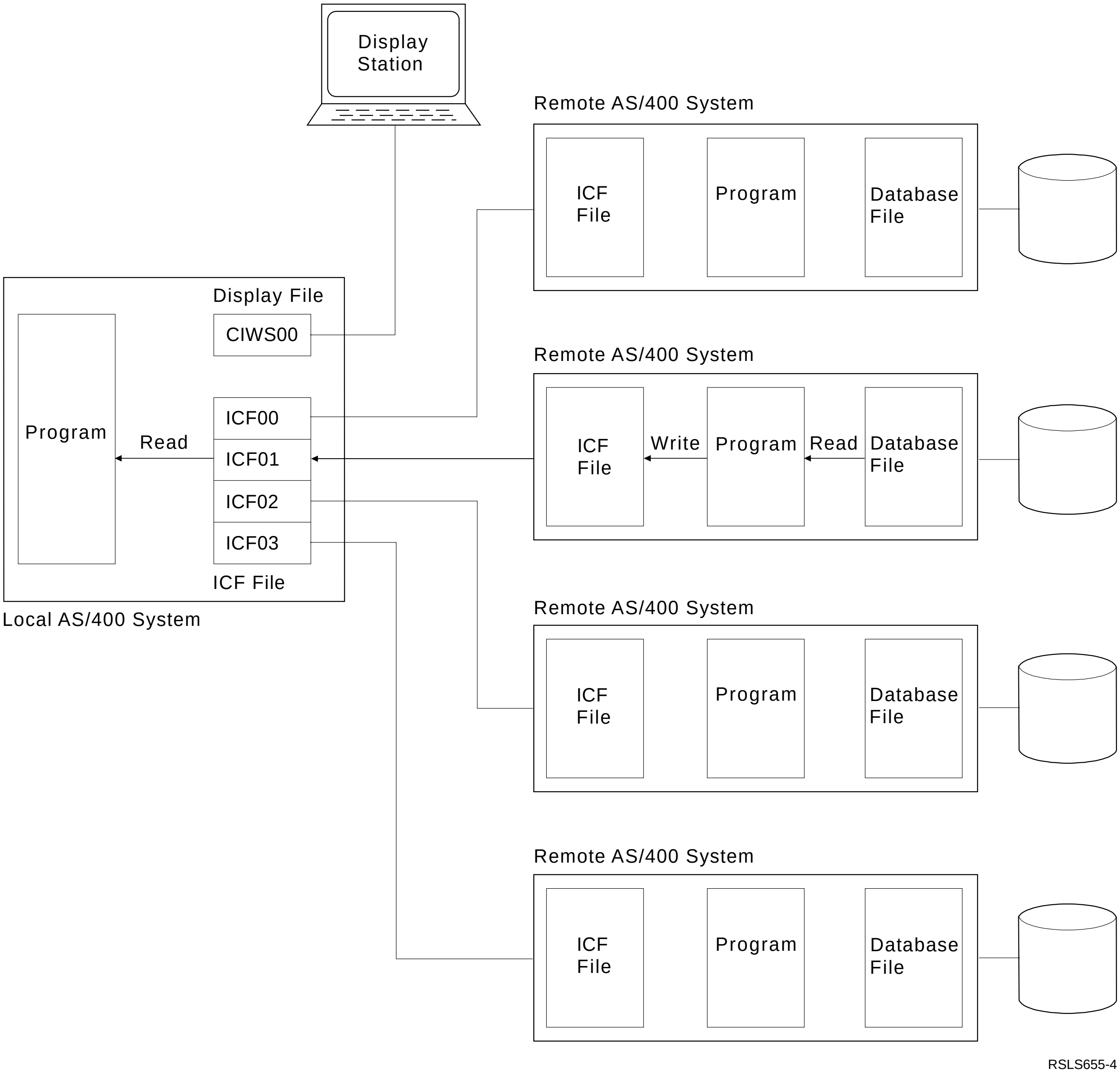


Figure 9-7. Target Program Sends a Reply

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 9-8.

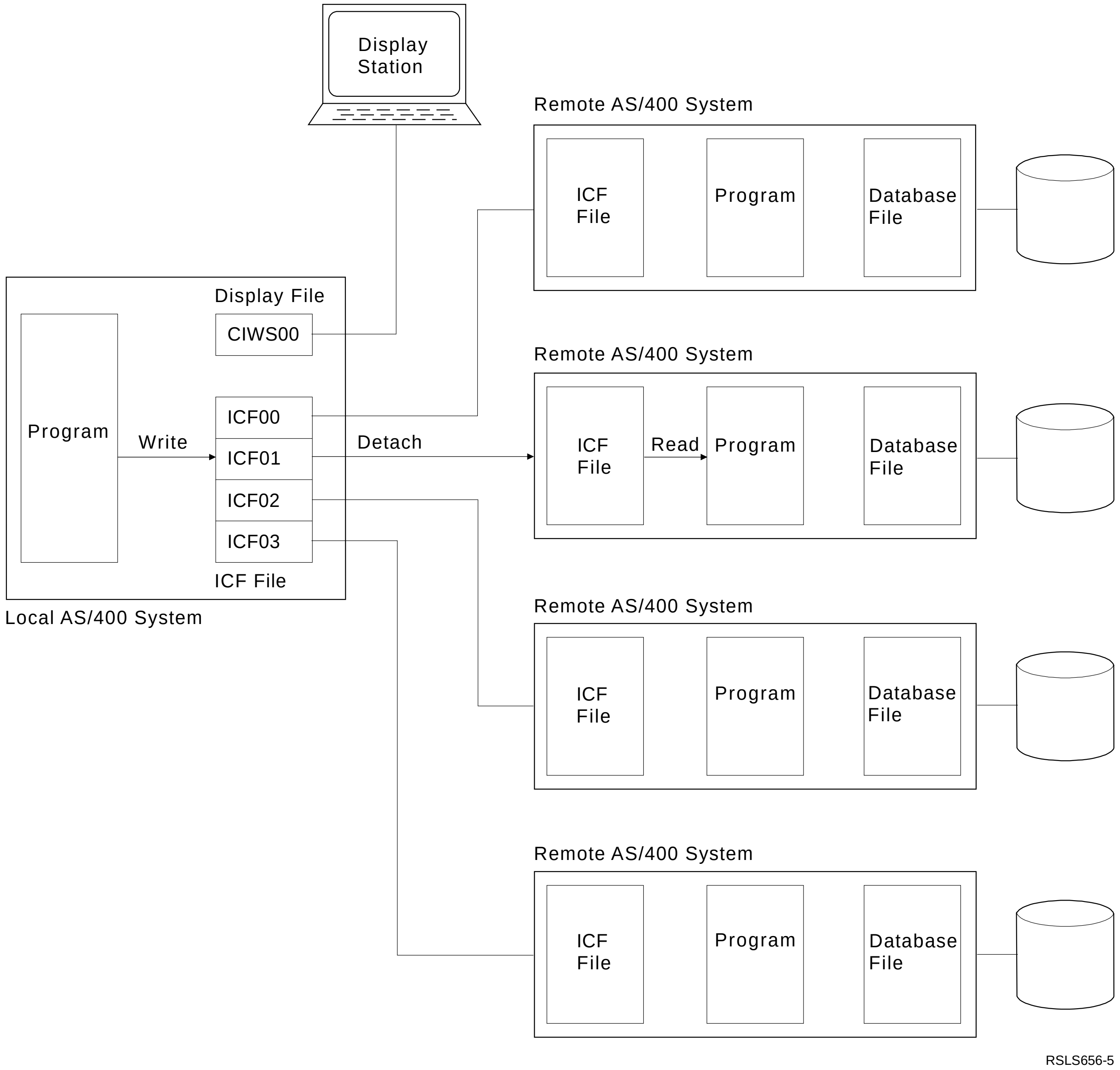


Figure 9-8. Program Ends the Session

Source Program Multiple-Session Inquiry

The following describes an ILE C source program multiple-session inquiry.

Program Files: The ILE C 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 below.

```
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
```

The DDS for the display file (DSPFIL) is illustrated below.

```
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')
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
```

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 9-9 on page 9-13. 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.

1 External file descriptions are included for the display file (DSPFIL).

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.

2 External file descriptions are included for the ICF file (CMNFIL).

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.

3 This structure is used to print the major or minor return code to the printer 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 printer file. See section **25** .

4 The variables to be used globally by the program are defined here. The common and display/ICF feedback area pointers, the file pointers, and the display file separate indicator area 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 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 initialized and defined. The variable `dsp_indic` 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, and the user is asked to type a `1` to process a request for item information, or a `2` to process a request for customer information.

If CMD 1 is pressed, indicator 99 is set by the system, a detach is sent to the remote program, the sessions are released, the files are closed, and the program ends. The separate indicator area 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 display, and reads input from the user. If CMD 1 is pressed, the end-program flag is set, and the program ends on 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 function is called to issue a read to the ICF file to get the item information.

13 This function selects the proper program device to use to send the item number to the target program and issues a write with an invite to the ICF file. 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 to the other program. The value returned from checking the return code (0 for 0000 return code, 1 for all other return codes) is returned to the caller.

14 This function gets called to receive item information from the target program after the user has specified a valid item number and that number has been sent to the target program.

The read is a read-from-invited-devices operation. The item number was previously sent with an invite.

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. If a time-out did occur, a message is written to the display asking the user to try again (by typing `1`) or to end the program (by typing `2`). This function gets called again if the user types `1`. If `2` is entered, control returns to **10** and the end-program flag is set.

If no data is received and the return code is 03XX, the request is 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** to read another number from the display.

The record received from the remote system is copied into the record used to print the information on the display, and then the information is written to the display using format ITMSC2.

If CMD 1 is pressed from this display, 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 display. If CMD 3 is pressed, the item inquiry menu is written to the display. By pressing the Enter key, 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 integers must then be converted back to character strings to conform with the character data types in the ICF file.

16 This procedure processes customer information requests, and is much like **12**. The customer inquiry display is displayed, indicators are checked, and functions are called to handle the sending and receiving of data.

17 This function sends the customer number to the remote system with an invite. Program device ICF00 is used for processing customer information.

18 This function receives information about a customer from the remote system and is structured like **14**. The major difference between this routine and the one in **14** is that if the record format is valid, a check for a 000000 customer number received from the remote system is made before any other checks. 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.

19 This function calls a procedure (section **26**) 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.

20 This function calls a procedure (section **26**) to access the display/ICF feedback area. It then checks for a 03 major return code, which indicates that data was not received.

21 This function calls a procedure (section **26**) to access the display/ICF feedback area. It then checks for a 0310 return code, which indicates that the timer interval has ended.

22 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.

23 This procedure releases the sessions that were acquired, and then calls another procedure to close the files.

24 This procedure closes the ICF, display, and printer files.

25 This procedure retrieves the return code to be printed from the display/ICF feedback area, and then prints the appropriate message to the printer file.

26 This procedure accesses 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.

```
                                    * * * * *   P R O L O G   * * * * *
Program name  . . . . . . . . . :   CSRCDMUL
  Library name  . . . . . . . . :      CICFLIB
Source file . . . . . . . . . . :   QICFPUB
  Library name  . . . . . . . . :      CICFLIB
Source member name  . . . . . . :   CSRCDMUL
Text Description  . . . . . . . :   Source C program for ICF Prog
Compiler options  . . . . . . . :   *SOURCE     *NOXREF     *NOSHOWUSR  *NOSHOWSYS  *NOSHOWSKP  *NOEXPMAC   *NOAGR
                                :   *NOPPONLY   *NODEBUG    *GEN        *NOSECLVL   *PRINT      *LOGMSG
Language level options  . . . . :   *EXTENDED
Source margins:
  Left margin . . . . . . . . . :   1
  Right margin  . . . . . . . . :   32767
Sequence columns:
  Left Column . . . . . . . . . :
  Right Column  . . . . . . . . :
Define name . . . . . . . . . . :
Generation options  . . . . . . :   *NOLIST     *NOXREF     *GEN        *NOATR      *NODUMP     *NOOPTIMIZE *NOALWBND
                                :   *NOANNO
Print file  . . . . . . . . . . :   QSYSPRT
  Library name  . . . . . . . . :      *LIBL
Message flagging level  . . . . :   0
Compiler message:
  Message limit . . . . . . . . :   *NOMAX
  Message limit severity  . . . :   30
Replace program object  . . . . :   *YES
User profile  . . . . . . . . . :   *USER
Authority . . . . . . . . . . . :   *CHANGE
Target Release  . . . . . . . . :   *CURRENT
Last change . . . . . . . . . . :   90/08/20 18:18:16
Source description  . . . . . . :   Source C program for ICF Prog
Compiler  . . . . . . . . . . . :   IBM ILE C/400 Compiler

                                    * * * * *   S O U R C E   * * * * *
Line  STMT                                                                                                    SEQNBR   INCNO
            *...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....+....9........
           |1                                                                                                |
   1       |#pragma mapinc("dspf", "cicflib/dspfil(*all)", "both", "p z")                                    |      1
   2       |#include "dspf"                                                                                  |      2
           |2                                                                                                |
   3       |#pragma mapinc("icff/itmrsp", "cicflib/cmnfil(itmrsp)", "input", "p z")                          |      3
   4       |#pragma mapinc("icff/dtlrsp", "cicflib/cmnfil(dtlrsp)", "input", "p z")                          |      4
   5       |#pragma mapinc("icff/itmreq", "cicflib/cmnfil(itmreq)", "output", "p z")                         |      5
   6       |#pragma mapinc("icff/dtlreq", "cicflib/cmnfil(dtlreq)", "output", "p z")                         |      6
   7       |#include "icff/itmrsp"                                                                           |      7
   8       |#include "icff/dtlrsp"                                                                           |      8
   9       |#include "icff/itmreq"                                                                           |      9
  10       |#include "icff/dtlreq"                                                                           |     10
  11       |/*-------------------------------------------------------------------*/                         |     11
  12       |/*  This program assigns four sessions as follows:                   */                         |     12
  13       |/*    'ICF00' to inquire about a customer account before an order is */                         |     13
  14       |/*            processed.                                             */                         |     14
  15       |/*    'ICF01' to inquire about the inventory status of an item being  */                         |     15
  16       |/*            ordered (item 000001 thru 399999).                     */                         |     16
  17       |/*    'ICF02' to inquire about the inventory status of an item being  */                         |     17
  18       |/*            ordered (item 400000 thru 699999).                     */                         |     18
  19       |/*    'ICF03' to inquire about the inventory status of an item being  */                         |     19
  20       |/*            ordered (item 700000 thru 999999).                     */                         |     20
  21       |/*  A display device is used to enter the request (using a customer   */                         |     21
  22       |/*  and an item menu) that is sent to the remote system.              */                         |     22
  23       |/*-------------------------------------------------------------------*/                         |     23
```

Figure 9-9 (Part 1 of 12). Source Program Example — CSRCDMUL (User-Defined Formats)

```

#define ON 1
#define OFF 0
#define ION '1'                             /* Indicator is set on */
#define IOFF '0'
#define ERROR 1                             /* Error occurred */
#define NOERROR 0
#define NORM_END 1                          /* Print normal end message */
#define RCD_ERR 2                           /* Print wrong record error msg */
#define ERR_END 3                           /* Print generic error message */
#include <stdio.h>                          /* Standard I/O header */
#include <recio.h>                          /* Record I/O header */
#include <stdlib.h>                         /* General utilities */
#include <stddef.h>                         /* Standard definitions */
#include <string.h>                         /* String handling utilities */
#include <xxfdbk.h>                         /* Feedback area structures */

#include <xxcvt.h>                          /* EPM conversion routines */

CICFLIB_DSPFIL_CIMENU_both_t  cimenu_dsp_i;
CICFLIB_DSPFIL_DTLMNU_both_t  dtlmnu_dsp_i;
CICFLIB_DSPFIL_DTLSCR_both_t  dtlscr_dsp_o;
CICFLIB_DSPFIL_ITMMNU_both_t  itmmnu_dsp_i;
CICFLIB_DSPFIL_ITMSC2_both_t  itmsc2_dsp_o;
CICFLIB_DSPFIL_ITMSC3_both_t  itmsc3_dsp_o;
CICFLIB_DSPFIL_TIMOUT_both_t  timout_dsp_i_o;

CICFLIB_CMNFIL_ITMRSP_i_t  itmrsp_icf_i;
CICFLIB_CMNFIL_DTLRSP_i_t  dtlrsp_icf_i;
CICFLIB_CMNFIL_ITMREQ_o_t  itmreq_icf_o;
CICFLIB_CMNFIL_DTLREQ_o_t  dtlreq_icf_o;
3
/*----------------------------------------------------------------------*/
/* Define structure used to write to the print file.                    */
/*----------------------------------------------------------------------*/

struct {
    char major??(2??);
    char minor??(2??);
    char filler??(32??);                    /* Used for padding with blanks */
} print_rec;
4
/*----------------------------------------------------------------------*/
/* Declare global variables.                                            */
/*----------------------------------------------------------------------*/

_RFILE  *icffptr;                           /* Ptr to ICF file */
_RFILE  *dspfptr;                           /* Ptr to display file */
_RFILE  *prtfptr;                           /* Ptr to print file */
_XXIOFB_T  *comm_fdbk;                      /* Ptr to common I/O feedback */
_XXIOFB_DSP_ICF_T  *dsp_icf_fdbk;           /* Ptr to dsp/ICF I/O feedback */
char dsp_indic??(99??);                     /* Display separate indic area */
5
int evoke_target(void);
void process_item_req(int *);
int send_item_req(void);
int rec_item_info(int *);
void calc_profit_loss(void);
void process_cust_req(int *);
int send_cust_req(void);
int rec_cust_info(int *);
int pos_ret_code(void);
int pos_ret_code(void);
int check_no_data(void);
int check_timeout(void);
void detach(void);
void end_job(void);
void close_files(void);
void print_msg(int);
void get_access_to_fb(void);
```

Figure 9-9 (Part 2 of 12). Source Program Example — CSRCDMUL (User-Defined Formats)

```

main()
{
    int  end_pgm_flag = OFF;                /* Signals program end request */
 6
    /*-----------------------------------------------------------*/
    /* Open ICF, display, and printer files.  If an error occurs,*/
    /* the program will end.                                     */
    /*-----------------------------------------------------------*/

    if ((icffptr = _Ropen("CICFLIB/CMNFIL", "ar+ indicators=y riofb=y"))
                                                       == NULL)
         exit(ERROR);
    if ((dspfptr = _Ropen("CICFLIB/DSPFIL", "ar+ indicators=y riofb=y"))
                                                     == NULL) {
         _Rclose(icffptr);
         exit(ERROR);
    }
    if ((prtfptr = _Ropen("CICFLIB/QSYSPRT", "wr")) == NULL) {
         _Rclose(icffptr);
         _Rclose(dspfptr);
         exit(ERROR);
    }
 7
    /*-----------------------------------------------------------*/
    /* Set up separate indicator area for the display file.      */
    /*-----------------------------------------------------------*/

     memset(dsp_indic, IOFF, 99);
     _Rindara(dspfptr, dsp_indic);
 8
    /*-----------------------------------------------------------*/
    /* Explicitly acquire four sessions. If an error occurs on   */
    /* any of the acquire operations then the ICF and display    */
    /* files will be closed, an error message will be printed,   */
    /* and the program will end.                                 */
    /*-----------------------------------------------------------*/

    _Racquire(icffptr, "ICF00");
    if (pos_ret_code() == ERROR) {
        close_files();
        exit(ERROR);
    }
    _Racquire(icffptr, "ICF01");
    if (pos_ret_code() == ERROR) {
        close_files();
        exit(ERROR);
    }
    _Racquire(icffptr, "ICF02");
    if (pos_ret_code() == ERROR) {
        close_files();
        exit(ERROR);
    }
    _Racquire(icffptr, "ICF03");
    if (pos_ret_code() == ERROR) {
        close_files();
        exit(ERROR);
    }
 9
```

Figure 9-9 (Part 3 of 12). Source Program Example — CSRCDMUL (User-Defined Formats)

```
     |    /*----------------------------------------------------------*/
     |    /* Evoke the target four times. If an error occurs on any of */
     |    /* the evoke operations then detach, release, and close      */
     |    /* operations will be issued, an error message printed, and  */
     |    /* the program will end.                                     */
     |    /*----------------------------------------------------------*/
     |
  28 |    _Rpgmdev(icffptr, "ICF00");
  29 |    if (evoke_target() == ERROR)
  30 |        exit(ERROR);
  31 |    _Rpgmdev(icffptr, "ICF01");
  32 |    if (evoke_target() == ERROR)
  33 |        exit(ERROR);
  34 |    _Rpgmdev(icffptr, "ICF02");
  35 |    if (evoke_target() == ERROR)
  36 |        exit(ERROR);
  37 |    _Rpgmdev(icffptr, "ICF03");
  38 |    if (evoke_target() == ERROR)
  39 |        exit(ERROR);
     |10
     |    /*----------------------------------------------------------*/
     |    /* Put out the main menu to the display, and depending on    */
     |    /* the input, either request item information or customer    */
     |    /* information from remote system.  If CMD 1 (indicator 99)  */
     |    /* is pressed on any screen, the program ends. If the user   */
     |    /* picks option 1, an item inquiry is processed, and if the  */
     |    /* user picks option 2 a customer inquiry is processed.  If  */
     |    /* CMD 1 wasn't pressed, nor option 1 or 2, then the input   */
     |    /* is invalid.                                               */
     |    /*----------------------------------------------------------*/
     |
  40 |    while (end_pgm_flag == OFF) {
  41 |        _Rformat(dspfptr, "CIMENU");
  42 |        _Rwrite(dspfptr, NULL, 0);
  43 |        memset(dsp_indic, IOFF, 99);
  44 |        _Rreadn(dspfptr, &cimenu_dsp_i, sizeof(cimenu_dsp_i), __DFT);
  45 |        if (dsp_indic??(98??) == IOFF) {
  46 |            if (cimenu_dsp_i.OPTION == '1')
  47 |                process_item_req(&end_pgm_flag);
     |            else
  48 |                if (cimenu_dsp_i.OPTION == '2')
  49 |                    process_cust_req(&end_pgm_flag);
     |        }
     |        else {
  50 |            print_msg(NORM_END);
  51 |            end_pgm_flag = ON;
     |        }
     |    }
  52 |    detach();
  53 |    end_job();
     |}
     |
     |
```

Figure 9-9 (Part 4 of 12). Source Program Example — CSRCDMUL (User-Defined Formats)

```
/*-------------------------------------------------------------------*/
/*                       Evoke the Target Program                    */
/* Evoke the target program. If an error occurs then a detach will be   */
/* sent, a release will be issued, and the ICF, display, and printer    */
/* files will be closed.                                             */
/*-------------------------------------------------------------------*/
 11
evoke_target()
{
    _Rformat(icffptr, "EVKREQ");
    _Rwrite(icffptr, NULL, 0);
    if (pos_ret_code() == ERROR) {
        print_msg(ERR_END);
        detach();
        end_job();
        return(ERROR);
    }
    return(NOERROR);
}


/*-------------------------------------------------------------------*/
/*                    Process Item Request Screen                    */
/* This routine puts out the item request screen and reads the input    */
/* from the user.  CMD 1 (end job), CMD 2 (go to main menu), and CMD 3  */
/* (reenter item number) are checked.  Routines to send a valid item    */
/* number and to receive item information are called.                */
/*-------------------------------------------------------------------*/
 12
void process_item_req(int *end_pgm_flag)
{
    _Rformat(dspfptr, "ITMMNU");
    _Rwrite(dspfptr, NULL, 0);
    memset(dsp_indic, IOFF, 99);
    _Rreadn(dspfptr, &itmmnu_dsp_i, sizeof(itmmnu_dsp_i), __DFT);
    if (dsp_indic??(98??) == ION) {         /* CMD 1, indic 99 */
        print_msg(NORM_END);
        *end_pgm_flag = ON;
    }
    else
        if (dsp_indic??(97??) == IOFF)      /* CMD 2, indic 98 */
            if (send_item_req() == NOERROR) {
                if (rec_item_info(end_pgm_flag) == ERROR) {
                    print_msg(ERR_END);
                    *end_pgm_flag = ON;
                }
            }
            else {
                print_msg(ERR_END);
                *end_pgm_flag = ON;
            }
}


```

Figure 9-9 (Part 5 of 12). Source Program Example — CSRCDMUL (User-Defined Formats)

```
/*--------------------------------------------------------------------*/
/*                      Send Item Request                             */
/* This routine sends the item number entered to the appropriate target */
/* program based on the range of the number.                          */
/*--------------------------------------------------------------------*/
13
send_item_req()
{
    _Rformat(icffptr, "ITMREQ");
    if (strncmp(itmmnu_dsp_i.ITEMNO, "399999", 6) != 1)
        _Rpgmdev(icffptr, "ICF01");
    else
        if (strncmp(itmmnu_dsp_i.ITEMNO, "699999", 6) != 1)
            _Rpgmdev(icffptr, "ICF02");
        else
            _Rpgmdev(icffptr, "ICF03");
    strncpy(itmreq_icf_o.ITEMNO, itmmnu_dsp_i.ITEMNO, 6);
    _Rwrite(icffptr, &itmreq_icf_o, sizeof(itmreq_icf_o));
    return(pos_ret_code());
}


/*--------------------------------------------------------------------*/
/*                   Receive Item Information                         */
/* The item information is read from the remote system.  A check is   */
/* made for three conditions following the read:  1) The remote system */
/* timed out, 2) no data was received, and 3) data was returned in an  */
/* unexpected format.                                                 */
/* If the remote system times out (maj/min 0310), a message is written */
/* to the screen asking to try again (enter 1) or to end the program   */
/* (enter 2).                                                         */
/* If no data is received (major 03) the request is sent again to the  */
/* remote system.                                                     */
/* If the record returns with the wrong record format, the program will */
/* end on error.                                                      */
/* If CMD 1 (in99) is pressed, the end program flag is set causing the  */
/* program to end upon return to main.  If CMD 2 (in98) is pressed,    */
/* the main menu is written to the screen.  If CMD 3 (in97) is pressed, */
/* the item inquiry menu is written to the screen.                    */
/* If the remote program didn't find the item, and the item number     */
/* "000000" was returned, then the item request screen is displayed.   */
/*--------------------------------------------------------------------*/
14
rec_item_info(int *end_pgm_flag)
{
    _Rreadindv(icffptr, &itmrsp_icf_i, sizeof(itmrsp_icf_i), __DFT);
    if (check_timeout() == 0) {
        _Rformat(dspfptr, "TIMOUT");
        _Rwrite(dspfptr, NULL, 0);
        _Rreadn(dspfptr, &timout_dsp_i_o, sizeof(timout_dsp_i_o), __DFT);
        if (timout_dsp_i_o.TIMRSP == '1')
            return(rec_item_info(end_pgm_flag));
        else
            if (timout_dsp_i_o.TIMRSP == '2') {
                print_msg(NORM_END);
                *end_pgm_flag = ON;
                return(NOERROR);
            }
    }
```

Figure 9-9 (Part 6 of 12). Source Program Example — CSRCDMUL (User-Defined Formats)

```
    else
        if (check_no_data() == 0) {         /* No data received */
            _Rformat(icffptr, "ITMREQ");
            _Rwrite(icffptr, &itmreq_icf_o, sizeof(itmreq_icf_o));
            return(pos_ret_code());
        }
        else {                              /* Check for valid record fmt */
            comm_fdbk = _Riofbk(icffptr);
            if (strncmp(comm_fdbk->rec_format, "ITMRSP    ", 10) != 0) {
                print_msg(RCD_ERR);
                return(ERROR);
            }
        }
    if (strncmp(itmrsp_icf_i.ITEMNO, "000000", 6) != 1)  /* Item not found */
        process_item_req(end_pgm_flag);
    else {
        strncpy(itmsc2_dsp_o.DSC, itmrsp_icf_i.DESC, 30);
        strncpy(itmsc2_dsp_o.QAVAIL, itmrsp_icf_i.QTYLST, 7);
        strncpy(itmsc2_dsp_o.QTYO, itmrsp_icf_i.QTYOO, 7);
        strncpy(itmsc2_dsp_o.QTYH, itmrsp_icf_i.QTYOH, 7);
        strncpy(itmsc2_dsp_o.QTYB, itmrsp_icf_i.QTYBO, 7);
        strncpy(itmsc2_dsp_o.UNT, itmrsp_icf_i.UNITQ, 2);
        strncpy(itmsc2_dsp_o.PR1, itmrsp_icf_i.PR01, 7);
        strncpy(itmsc2_dsp_o.PR5, itmrsp_icf_i.PR05, 7);
        strncpy(itmsc2_dsp_o.UFR, itmrsp_icf_i.UFRT, 5);
        _Rformat(dspfptr, "ITMSC2");        /* Display item screen 2 */
        _Rwrite(dspfptr, &itmsc2_dsp_o, sizeof(itmsc2_dsp_o));
        memset(dsp_indic, IOFF, 99);
        _Readn(dspfptr, NULL, 0, __DFT);
        if (dsp_indic??(98??) == ION) {     /* CMD 1, indic 99 */
            print_msg(NORM_END);
            *end_pgm_flag = ON;
        }
        else
            if (dsp_indic??(96??) == ION)   /* CMD 3, indic 97 */
                process_item_req(end_pgm_flag);
            else
                if (dsp_indic??(97??) == IOFF) {
                    calc_profit_loss();
                    _Rformat(dspfptr, "ITMSC3");  /* Display screen 3 */
                    _Rwrite(dspfptr, &itmsc3_dsp_o, sizeof(itmsc3_dsp_o));
                    memset(dsp_indic, IOFF, 99);
                    _Readn(dspfptr, NULL, 0, __DFT);
                }
    }
    return(NOERROR);
}
```

Figure 9-9 (Part 7 of 12). Source Program Example — CSRCDMUL (User-Defined Formats)

```
/*------------------------------------------------------------------*/
/*                     Calculate Profit/Loss                        */
/* This routine calculates profit and loss figures and displays them   */
/* on screen two of the item.                                       */
/*------------------------------------------------------------------*/
15
void calc_profit_loss()
{
    long qtylst_l;
    double slstm_d, csttm_d;
    double pr01_d, losts_d, profit_d;

    csttm_d = QXXZTOD(itmrsp_icf_i.CSTTM, 9, 2);
    qtylst_l = QXXZTOI(itmrsp_icf_i.QTYLST, 7, 0);
    pr01_d = QXXZTOD(itmrsp_icf_i.PR01, 7, 2);
    slstm_d = QXXZTOD(itmrsp_icf_i.SLSTM, 9, 2);
    profit_d = csttm_d - slstm_d;
    profit_d *= 100;
    if (slstm_d > 0)
        profit_d /= slstm_d;
    losts_d = qtylst_l * pr01_d;
    strncpy(itmsc3_dsp_o.SLSM, itmrsp_icf_i.SLSTM, 9);
    strncpy(itmsc3_dsp_o.SLSY, itmrsp_icf_i.SLSTY, 11);
    strncpy(itmsc3_dsp_o.CSTM, itmrsp_icf_i.CSTTM, 9);
    strncpy(itmsc3_dsp_o.CSTY, itmrsp_icf_i.CSTTY, 11);
    QXXDTOZ(itmsc3_dsp_o.LOSTS, 11, 2, losts_d);
    QXXDTOZ(itmsc3_dsp_o.PROFIT, 5, 2, profit_d);
}


/*------------------------------------------------------------------*/
/*                    Process Customer Request                      */
/* This routine puts out the customer request screen and reads the     */
/* input from the user.  CMD 1 (end job) and CMD 2 (go to main menu)    */
/* are checked.  Routines to send the customer number and to receive    */
/* customer information are called.                                 */
/*------------------------------------------------------------------*/
16
void process_cust_req(int *end_pgm_flag)
{
    _Rformat(dspfptr, "DTLMNU");
    _Rwrite(dspfptr, NULL, 0);
    memset(dsp_indic, IOFF, 99);
    _Rreadn(dspfptr, &dtlmnu_dsp_i, sizeof(dtlmnu_dsp_i), __DFT);
    if (dsp_indic??(98??) == ION) {         /* CMD 1, indic 99 */
        print_msg(NORM_END);
        *end_pgm_flag = ON;
    }
    else
        if (dsp_indic??(97??) == IOFF)      /* CMD 2, indic 98 */
            if (send_cust_req() == NOERROR) {
                if (rec_cust_info(end_pgm_flag) == ERROR) {
                    print_msg(ERR_END);
                    *end_pgm_flag = ON;
                }
            }
            else {
                print_msg(ERR_END);
                *end_pgm_flag = ON;
            }
}


```

Figure 9-9 (Part 8 of 12). Source Program Example — CSRCDMUL (User-Defined Formats)

```
/*----------------------------------------------------------------------*/
/*                        Send Customer Request                         */
/* This routine sends the customer number entered by the user to the    */
/* remote program.  The number has already been read in.                */
/*----------------------------------------------------------------------*/
 17
send_cust_req()
{
        strncpy(dtlreq_icf_o.CUSTNO, dtlmnu_dsp_i.CUSTNO, 6);
        _Rformat(icffptr, "DTLREQ");
        _Rpgmdev(icffptr, "ICF00");
        _Rwrite(icffptr, &dtlreq_icf_o, sizeof(dtlreq_icf_o));
        return(pos_ret_code());
}


/*----------------------------------------------------------------------*/
/*                     Receive Customer Information                     */
/* This routine attempts to receive customer information from the       */
/* target program.  A check is made for the following three conditions  */
/* following the read operation: 1) The remote system timed out, 2) no  */
/* data was received, and 3) data returned in an unexpected format.     */
/* If the remote system times out (maj/min 0310), a message is written  */
/* to the screen asking user to try again (enter 1) or to end the       */
/* program (enter 2).                                                   */
/* If no data is received (major 03) the request is sent again to the   */
/* remote system.                                                       */
/* If the record returns with the wrong record format, the program will */
/* end on error.                                                        */
/* If the remote program didn't find the customer, and the item number  */
/* "000000" was returned, the main menu is displayed.                   */
/*----------------------------------------------------------------------*/
 18
rec_cust_info(int *end_pgm_flag)
{
    _Rreadindv(icffptr, &dtlrsp_icf_i, sizeof(dtlrsp_icf_i), __DFT);
    if (check_timeout() == 0) {
        _Rformat(dspfptr, "TIMOUT");
        _Rwrite(dspfptr, NULL, 0);
        _Rreadn(dspfptr, &timout_dsp_i_o, sizeof(timout_dsp_i_o), __DFT);
        if (timout_dsp_i_o.TIMRSP == '1')
            return(rec_cust_info(end_pgm_flag));
        else
            if (timout_dsp_i_o.TIMRSP == '2') {
                print_msg(NORM_END);
                *end_pgm_flag = ON;
                return(NOERROR);
            }
    }
    else
        if (check_no_data() == 0) {         /* No data received */
            _Rformat(icffptr, "DTLREQ");
            _Rwrite(icffptr, &dtlreq_icf_o, sizeof(dtlreq_icf_o));
            return(pos_ret_code());
        }
        else {                              /* Was item found ? */
            if (strncmp(dtlrsp_icf_i.CUSTNO, "000000", 6) != 1)
                return(NOERROR);
            else {                          /* Check record format */
                comm_fdbk = _Riofbk(icffptr);
                if (strncmp(comm_fdbk->rec_format, "DTLRSP    ", 10) != 0) {
                    print_msg(RCD_ERR);
                    return(ERROR);
                }
            }
        }
```

Figure 9-9 (Part 9 of 12). Source Program Example — CSRCDMUL (User-Defined Formats)

```
    strncpy(dtlscr_dsp_o.CUSTN, dtlrsp_icf_i.CUSTNO, 6);
    strncpy(dtlscr_dsp_o.CNAME, dtlrsp_icf_i.DNAME, 30);
    strncpy(dtlscr_dsp_o.DLSTR, dtlrsp_icf_i.DLSTOR, 6);
    strncpy(dtlscr_dsp_o.DSLSM, dtlrsp_icf_i.DSLSTM, 9);
    strncpy(dtlscr_dsp_o.DSPM1, dtlrsp_icf_i.DSPM01, 9);
    strncpy(dtlscr_dsp_o.DSPM2, dtlrsp_icf_i.DSPM02, 9);
    strncpy(dtlscr_dsp_o.DSTYD, dtlrsp_icf_i.DSTTYD, 11);
    strncpy(dtlscr_dsp_o.DEPT, dtlrsp_icf_i.IDEPT, 3);
    _Rformat(dspfptr, "DTLSCR");           /* Display customer info */
    _Rwrite(dspfptr, &dtlscr_dsp_o, sizeof(dtlscr_dsp_o));
    memset(dsp_indic, IOFF, 99);
    _Rreadn(dspfptr, NULL, 0, __DFT);
    if (dsp_indic??(98??) == ION) {
        print_msg(NORM_END);
        *end_pgm_flag = ON;
    }
    return(NOERROR);
}


/*--------------------------------------------------------------------*/
/*                    Check For Successful Operation                  */
/* If the major return code is 00 the last operation attempted was    */
/* successful, otherwise an error occurred.                           */
/*--------------------------------------------------------------------*/
 19
pos_ret_code()
{
    get_access_to_fb();
    if (strncmp(dsp_icf_fdbk->major_ret_code, "00", 2) == 0)
        return(NOERROR);
    else
        return(ERROR);
}


/*--------------------------------------------------------------------*/
/*                    Check for No Data Received                      */
/* If the major return code is 03 then no data was received on an input */
/* operation.                                                         */
/*--------------------------------------------------------------------*/
 20
check_no_data()
{
    get_access_to_fb();
    if (strncmp(dsp_icf_fdbk->major_ret_code, "03", 2) == 0)
        return(0);
    else
        return(1);
}


/*--------------------------------------------------------------------*/
/*                      Check for Timeout                             */
/* If the major/minor return code is 0310, then a timeout occurred when */
/* reading from an invited program device.  Return 1 if timeout occurred*/
/* otherwise return 0.                                                */
/*--------------------------------------------------------------------*/
 21
check_timeout()
{
    get_access_to_fb();
    if ((strncmp(dsp_icf_fdbk->major_ret_code, "03", 2) == 0) &&
        (strncmp(dsp_icf_fdbk->minor_ret_code, "10", 2) == 0))
        return(0);
    else
        return(1);
}


```

Figure 9-9 (Part 10 of 12). Source Program Example — CSRCDMUL (User-Defined Formats)

```
/*----------------------------------------------------------------------*/
/*                         Issue Detach                                 */
/* A detach is sent to each program device to end the transaction with  */
/* the remote system.                                                   */
/*----------------------------------------------------------------------*/
 22
void detach()
{
    _Rformat(icffptr, "DETACH");
    _Rpgmdev(icffptr, "ICF00");
    _Rwrite(icffptr, NULL, 0);
    _Rpgmdev(icffptr, "ICF01");
    _Rwrite(icffptr, NULL, 0);
    _Rpgmdev(icffptr, "ICF02");
    _Rwrite(icffptr, NULL, 0);
    _Rpgmdev(icffptr, "ICF03");
    _Rwrite(icffptr, NULL, 0);
}


/*----------------------------------------------------------------------*/
/*                       End the Job                                    */
/* A release is sent to each program device to end the sessions, and    */
/* the ICF, display, and printer files are closed.                      */
/*----------------------------------------------------------------------*/
 23
void end_job()
{
    _Rrelease(icffptr, "ICF00");
    _Rrelease(icffptr, "ICF01");
    _Rrelease(icffptr, "ICF02");
    _Rrelease(icffptr, "ICF03");
    close_files();
}


/*----------------------------------------------------------------------*/
/*                        Close the Files                               */
/* Close the ICF, display, and print files.                             */
/*----------------------------------------------------------------------*/
 24
void close_files()
{
        _Rclose(icffptr);
        _Rclose(dspfptr);
        _Rclose(prtfptr);
}


/*----------------------------------------------------------------------*/
/*                        Print Message                                 */
/* Write message and return code to print file.                         */
/*----------------------------------------------------------------------*/
 25
void print_msg(int mtype)
{
    get_access_to_fb();
    strncpy(print_rec.major, dsp_icf_fdbk->major_ret_code, 2);
    strncpy(print_rec.minor, dsp_icf_fdbk->minor_ret_code, 2);
    strncpy(print_rec.filler, "                                ", 32);
    _Rwrite(prtfptr, "RETURN CODE:                        ", 36);
    _Rwrite(prtfptr, &print_rec, sizeof(print_rec));
    if (mtype == NORM_END)
        _Rwrite(prtfptr, "PROGRAM CSRCDMUL COMPLETED NORMALLY ", 36);
    else
        if (mtype == RCD_ERR)
            _Rwrite(prtfptr, "RECORD FORMAT IS INCORRECT ON READ  ", 36);
        else
            _Rwrite(prtfptr, "PROGRAM ENDED DUE TO ERROR IN CMNFIL", 36);
}

```

Figure 9-9 (Part 11 of 12). Source Program Example — CSRCDMUL (User-Defined Formats)

```
 638       |                                                                                                |    638
 639       |/*-------------------------------------------------------------------*/                         |    639
 640       |/*                Get Access to DSP/ICF Feedback                      */                         |    640
 641       |/* The feedback areas are updated after each display or ICF file I/O    */                       |    641
 642       |/* operation, and so the pointers must be updated to point to the "new" */                       |    642
 643       |/* feedback areas to get the return code.  The offset to the display/   */                       |    643
 644       |/* ICF feedback area is contained in the common I/O feedback and is     */                       |    644
 645       |/* added to the pointer to the common feedback area to get access to    */                       |    645
 646       |/* display/ICF feedback area.                                           */                       |    646
 647       |/*-------------------------------------------------------------------*/                         |    647
 648       | 26                                                                                             |    648
 649       |void get_access_to_fb()                                                                         |    649
 650       |{                                                                                               |    650
 651     1 |    comm_fdbk = _Riofbk(icffptr);                                                               |    651
 652       |    dsp_icf_fdbk = (_XXIOFB_DSP_ICF_T *)((char *)comm_fdbk +                                    |    652
 653     2 |                                        comm_fdbk->file_dep_fb_offset);                         |    653
 654       |}                                                                                               |    654
                                 * * * * *   E N D   O F   S O U R C E   * * * * *

                                      * * * * *   I N C L U D E S   * * * * *
                 INCNO   Include Name        Actual Include Name
                    1    dspf                            CICFLIB/dspfil(*all)
                    2    icff/itmrsp                     CICFLIB/cmnfil(itmrsp)
                    3    icff/dtlrsp                     CICFLIB/cmnfil(dtlrsp)
                    4    icff/itmreq                     CICFLIB/cmnfil(itmreq)
                    5    icff/dtlreq                     CICFLIB/cmnfil(dtlreq)
                    6    stdio.h                         QCC/H/STDIO
                    7    stddef.h                        QCC/H/STDDEF
                    8    errno.h                         QCC/H/ERRNO
                    9    signal.h                        QCC/H/SIGNAL
                   10    ctype.h                         QCC/H/CTYPE
                   11    stdarg.h                        QCC/H/STDARG
                   12    recio.h                         QCC/H/RECIO
                   13    xxfdbk.h                        QCC/H/XXFDBK
                   14    stdlib.h                        QCC/H/STDLIB
                   15    stddef.h                        QCC/H/STDDEF
                   16    string.h                        QCC/H/STRING
                   17    xxfdbk.h                        QCC/H/XXFDBK
                   18    xxcvt.h                         QCC/H/XXCVT
                              * * * * *   E N D   O F   I N C L U D E S   * * * * *

                               * * * * *   M E S S A G E   S U M M A R Y   * * * * *
    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   M E S S A G E   S U M M A R Y   * * * * *
ROUTINE          BLOCK NUMBER  SCOPE  TYPE
<MAIN>                 2       LOCAL  MAIN-PROGRAM
__reads              115       LOCAL  PROCEDURE
__rwrite             118       LOCAL  PROCEDURE
__rfmt               125       LOCAL  PROCEDURE
__memset             172       LOCAL  PROCEDURE
__strncmp            190       LOCAL  PROCEDURE
__strncpy            192       LOCAL  PROCEDURE
QXXDTOZ              204       LOCAL  PROCEDURE
QXXZTOD              209       LOCAL  PROCEDURE
QXXZTOI              210       LOCAL  PROCEDURE
evoke_target         229       ENTRY   PROCEDURE
process_item_req     230       ENTRY   PROCEDURE
send_item_req        231       ENTRY   PROCEDURE
rec_item_info        232       ENTRY   PROCEDURE
calc_profit_loss     233       ENTRY   PROCEDURE
process_cust_req     234       ENTRY   PROCEDURE
send_cust_req        235       ENTRY   PROCEDURE
rec_cust_info        236       ENTRY   PROCEDURE
pos_ret_code         237       ENTRY   PROCEDURE
check_no_data        238       ENTRY   PROCEDURE
check_timeout        239       ENTRY   PROCEDURE
detach               240       ENTRY   PROCEDURE
end_job              241       ENTRY   PROCEDURE
close_files          242       ENTRY   PROCEDURE
print_msg            243       ENTRY   PROCEDURE
get_access_to_fb     244       ENTRY   PROCEDURE
main                 245       ENTRY   PROCEDURE
```

Figure 9-9 (Part 12 of 12). Source Program Example — CSRCDMUL (User-Defined Formats)

Target Program Multiple-Session Inquiry: The following describes the ILE C target program for multiple-session inquiry program example.

Program Files: The ILE C 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 below.

```
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
```

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)
```

The DDS source for the database file (PFILE) is illustrated below:

```
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  * * * *
```

Program Explanation: The following explains the structure of the program example illustrated in Figure 9-10 on page 9-27. 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.

1 External file descriptions are included for the ICF file (CFILE).

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 printer 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 printer file. See 15 and 16.

3 The variables to be used globally by the program are defined here. The common and display/ICF feedback area pointers, the file pointers, and the ICF file separate indicator area are defined.

4 The routines, except the main routine, are prototyped so the compiler knows the type of value returned and the type of parameters passed, if any.

5 The ICF, printer, and database files are opened for record input, output, or both. The ICF file is opened with the separate indicator area option specified.

6 The separate indicator area is initialized and defined. The variable `icf_indic` is a character array of size 99.

7 The program device ICF00 used by the program is explicitly acquired.

8 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 13) to close the files is made before the program ends.

9 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 returns to 8 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 value of 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 8. If the turnaround is not received, the program ends.

10 The number received from the remote system is used to find the corresponding item or customer record by key in PFILE. If the record is not found, 000000 is sent to the source program. The record ID is set to ‘I’.

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.

11 This function calls a procedure (section 16) to access the display/ICF feedback area. It then checks for a 3431 return code or a 00 major return code.

12 This function calls a procedure (section 16) to access the display/ICF feedback area. It then checks for a 00 major return code, which indicates that the last operation was successful.

13 This procedure closes the ICF, printer, and database files.

14 This procedure retrieves the return code to be printed from the display/ICF feedback area and prints a normal end message to the printer file.

15 This procedure retrieves the return code to be printed from the display/ICF feedback area and prints an abnormal end message to the printer file.

16 This procedure accesses 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.

```
                                    * * * * *   P R O L O G   * * * * *
Program name . . . . . . . . . :   CTGTDMUL
  Library name . . . . . . . . :     CICFLIB
Source file . . . . . . . . . . :   QICFPUB
  Library name . . . . . . . . :     CICFLIB
Source member name  . . . . . . :   CTGTDMUL
Text Description  . . . . . . . :   Target C program for ICF Programming
Compiler options  . . . . . . . :   *SOURCE     *NOXREF     *NOSHOWUSR  *NOSHOWSYS  *NOSHOWSKP  *NOEXPMAC   *NOAGR
                                :   *NOPPONLY   *NODEBUG    *GEN        *NOSECLVL   *PRINT      *LOGMSG
Language level options  . . . . :   *EXTENDED
Source margins:
  Left margin . . . . . . . . . :   1
  Right margin  . . . . . . . . :   32767
Sequence columns:
  Left Column . . . . . . . . . :
  Right Column  . . . . . . . . :
Define name . . . . . . . . . . :
Generation options  . . . . . . :   *NOLIST     *NOXREF     *GEN        *NOATR      *NODUMP     *NOOPTIMIZE *NOALWBND
                                :   *NOANNO
Print file  . . . . . . . . . . :   QSYSPRT
  Library name  . . . . . . . . :     *LIBL
Message flagging level  . . . . :   0
Compiler message:
  Message limit . . . . . . . . :   *NOMAX
  Message limit severity  . . . :   30
Replace program object  . . . . :   *YES
User profile  . . . . . . . . . :   *USER
Authority . . . . . . . . . . . :   *LIBCRTAUT
Target Release  . . . . . . . . :   *CURRENT
Last change . . . . . . . . . . :   90/09/20 14:46:14
Source description  . . . . . . :   Target C program for ICF Programming
Compiler  . . . . . . . . . . . :   IBM ILE C/400 Compiler

                                    * * * * *   S O U R C E   * * * * *
Line  STMT                                                                                                        SEQNBR   INCNO
           *...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....+....9.......
          |1                                                                                                |
   1      |#pragma mapinc("icff/snd", "cicflib/cfile(sndpart)", "output", "p z")                            |      1
   2      |#include "icff/snd"                                                                              |      2
   3      |#pragma mapinc("icff/rcv", "cicflib/cfile(rcvpart)", "input", "p z")                             |      3
   4      |#include "icff/rcv"                                                                              |      4
   5      |/*----------------------------------------------------------------------*/                       |      5
   6      |/* This program will handle the request for either a customer number or */                       |      6
   7      |/* an item number.  This is accomplished by making the data base file   */                       |      7
   8      |/* structure (key length, key position, record length, record size,     */                       |      8
   9      |/* etc.) the same for both files with only the record contents          */                       |      9
  10      |/* different.  The database file is searched with a customer number or  */                       |     10
  11      |/* item number as a key.                                                */                       |     11
  12      |/* This program ends when a detach request is received from the source  */                       |     12
  13      |/* program.                                                             */                       |     13
  14      |/* Indicators associated with the ICF file are defined and are          */                       |     14
  15      |/* referenced for every I/O operation issued.                           */                       |     15
  16      |/*----------------------------------------------------------------------*/                       |     16
  17      |                                                                                                 |     17
  18      |#define ION '1'                             /* Indicator set on */                               |     18
  19      |#define IOFF '0'                                                                                 |     19
  20      |#define ERROR 1                             /* Error occurred */                                 |     20
  21      |#define NOERROR 0                                                                                |     21
  22      |#define TRUE 1                                                                                   |     22
  23      |#define FALSE 0                                                                                  |     23
  24      |#include <stdio.h>                          /* Standard I/O header */                            |     24
  25      |#include <recio.h>                          /* Record I/O header */                              |     25
  26      |#include <stdlib.h>                         /* General utilities */                              |     26
  27      |#include <stddef.h>                         /* Standard definitions */                           |     27
  28      |#include <string.h>                         /* String handling utilities */                      |     28
  29      |#include <xxfdbk.h>                         /* Feedback area structures */                       |     29
  30      |                                                                                                 |     30
```

Figure 9-10 (Part 1 of 6). Target Program Example — CTGTDMUL (User-Defined Formats)

```
2
/*---------------------------------------------------------------------*/
/* Define the structure to be used to write to the print file.         */
/*---------------------------------------------------------------------*/

struct {
    char major??(2??);
    char minor??(2??);
    char filler??(29??);                    /* Used for padding with blanks */
} print_rec;
3
/*---------------------------------------------------------------------*/
/* Declare global variables.                                           */
/*---------------------------------------------------------------------*/

_RFILE *icffptr;                            /* Ptr to ICF file */
_RFILE *prtfptr;                            /* Ptr to print file */
_RFILE *dbfptr;                             /* Ptr to database file */
_RIOFB_T  *rio_fdbk;                        /* Ptr to partial I/O feedback */
_XXIOFB_T *comm_fdbk;                       /* Ptr to common I/O feedback */
_XXIOFB_DSP_ICF_T *dsp_icf_fdbk;            /* Ptr to dsp/ICF I/O feedback */
char icf_indic??(99??);                     /* ICF file separate indic area */

CICFLIB_CFILE_SNDPART_o_t  sndpart_icf_o;
CICFLIB_CFILE_RCVPART_i_t  rcvpart_icf_i;
4
int read_cfile(void);
int send_data(void);
int check_ret_code(void);
int pos_ret_code(void);
void close_files(void);
void print_norm_end(void);
void print_error_end(void);
void get_access_to_fb(void);

main()
{
    int error_or_end = FALSE;               /* Set if I/O error occured */
5
    /*----------------------------------------------------------*/
    /* Open the ICF and printer files.  If an error occurs the   */
    /* program ends.                                             */
    /*----------------------------------------------------------*/

    if ((icffptr = _Ropen("CICFLIB/CFILE", "ar+ indicators=y riofb=y"))
                                                       == NULL)
         exit(ERROR);
    if ((prtfptr = _Ropen("CICFLIB/QSYSPRT", "wr")) == NULL)            {
         _Rclose(icffptr);
         exit(ERROR);
    }

    if ((dbfptr = _Ropen("CICFLIB/PFILE", "rr riofb=y")) == NULL)       {
         _Rclose(icffptr);
         _Rclose(prtfptr);
         exit(ERROR);
    }
6
    /*----------------------------------------------------------*/
    /* Set up separate indicator area for the ICF file.         */
    /*----------------------------------------------------------*/

    memset(icf_indic, IOFF, 99);
    _Rindara(icffptr, icf_indic);
7
    /*----------------------------------------------------------*/
    /* Explicitly acquire a session.  If an error occurs the     */
    /* files will be closed and the program will end.            */
    /*----------------------------------------------------------*/

```

Figure 9-10 (Part 2 of 6). Target Program Example — CTGTDMUL (User-Defined Formats)

```
    _Racquire(icffptr, "ICF00");
    if (pos_ret_code() == ERROR) {
        print_error_end();
        close_files();
        exit(ERROR);
    }
 8
    /*-----------------------------------------------------------*/
    /* Process requests from the source program until a detach   */
    /* is received or an error occurs.                           */
    /*-----------------------------------------------------------*/

    _Rpgmdev(icffptr, "ICF00");
    while (error_or_end == FALSE) {
        if (read_cfile() == NOERROR) {
            if (send_data() == ERROR)
                error_or_end = TRUE;
        }
        else
            error_or_end = TRUE;
    }
    close_files();
}


/*----------------------------------------------------------------------*/
/*                        Read ICF File                                 */
/* This routine issues a read operation to the program device.  The     */
/* detach indication is checked, and if it was signaled main is         */
/* notified.  If the receive turnaround indicator isn't set, or an      */
/* unexpected return code is received, main is notified of an error.    */
/*----------------------------------------------------------------------*/
 9
read_cfile()
{
    _Rformat(icffptr, "RCVPART");
    memset(icf_indic, IOFF, 99);
    _Rreadn(icffptr, &rcvpart_icf_i, sizeof(rcvpart_icf_i), __DFT);
    if (icf_indic??(43??) == ION) {        /* Detach indicator (44) set */
        print_norm_end();
        return(ERROR);
    }
    else
        if (icf_indic??(39??) == ION)      /* Rec. turnaround (40) set */
            if (check_ret_code() == NOERROR)
                return(NOERROR);
            else {
                print_error_end();
                return(ERROR);
            }
        else {
            print_error_end();
            return(ERROR);
        }
}


/*----------------------------------------------------------------------*/
/*                     Write to ICF File                                */
/* A request from the source program results in reading a single record */
/* containing the requested customer or order number.  The response will*/
/* be returned in a single record containing either the item or customer*/
/* information, depending on the data base content.                     */
/* The response is sent to the source program by writing to the program */
/* device file using format sndpart.                                    */
/* The database file is searched by key.  If the number of bytes        */
/* returned is 0, then the record was not found on the read, in which   */
/* case "000000" is sent back to the source program.                    */
/*----------------------------------------------------------------------*/
 10
```

Figure 9-10 (Part 3 of 6). Target Program Example — CTGTDMUL (User-Defined Formats)

```
   |send_data()
   |{
   |    rio_fdbk = _Rreadk(dbfptr, &sndpart_icf_o, sizeof(sndpart_icf_o),
 1 |                       __KEY_EQ, rcvpart_icf_i.RECID2, 6);
 2 |    if (rio_fdbk->num_bytes == 0) {
 3 |        sndpart_icf_o.RECTYP = 'I';
 4 |        strncpy(sndpart_icf_o.ITEMNO, "000000", 6);
   |    }
 5 |    _Rformat(icffptr, "SNDPART");
 6 |    _Rwrite(icffptr, &sndpart_icf_o, sizeof(sndpart_icf_o));
 7 |    if (pos_ret_code() == NOERROR)
 8 |        return(NOERROR);
   |    else {
 9 |        print_error_end();
10 |        return(ERROR);
   |    }
   |}
   |
   |
   |/*----------------------------------------------------------------------*/
   |/*                         Check Return Code                            */
   |/* This routine checks the return code after a receive operation for   */
   |/* 0000 and 3431.  Anything else is considered an error.                */
   |/*----------------------------------------------------------------------*/
   | 11
   |check_ret_code()
   |{
 1 |    get_access_to_fb();
   |    if (strncmp(dsp_icf_fdbk->major_ret_code, "34", 2) == 0 &&
 2 |        strncmp(dsp_icf_fdbk->minor_ret_code, "31", 2) == 0)
 3 |        return(NOERROR);
   |    else
 4 |        return(pos_ret_code());
   |}
   |
   |
   |/*----------------------------------------------------------------------*/
   |/*                     Check for Successful Operation                   */
   |/* This routine checks the major return code of 00 to see if the last   */
   |/* operation was successful.                                            */
   |/*----------------------------------------------------------------------*/
   | 12
   |pos_ret_code()
   |{
 1 |    get_access_to_fb();
 2 |    if (strncmp(dsp_icf_fdbk->major_ret_code, "00", 2) == 0)
 3 |        return(NOERROR);
   |    else
 4 |        return(ERROR);
   |}
   |
   |
   |/*----------------------------------------------------------------------*/
   |/*                         Close the Files                              */
   |/* Close the ICF, print, and database files.                            */
   |/*----------------------------------------------------------------------*/
   | 13
   |void close_files()
   |{
 1 |        _Rclose(icffptr);
 2 |        _Rclose(prtfptr);
 3 |        _Rclose(dbfptr);
   |}
   |
   |
```

Figure 9-10 (Part 4 of 6). Target Program Example — CTGTDMUL (User-Defined Formats)

```
/*----------------------------------------------------------------------*/
/*                     Print Normal End Message                         */
/* Write normal end message and return code to print file.              */
/*----------------------------------------------------------------------*/
 14
void print_norm_end()
{
    get_access_to_fb();
    strncpy(print_rec.major, dsp_icf_fdbk->major_ret_code, 2);
    strncpy(print_rec.minor, dsp_icf_fdbk->minor_ret_code, 2);
    strncpy(print_rec.filler, "                             ", 29);
    _Rwrite(prtfptr, "RETURN CODE:                     ", 33);
    _Rwrite(prtfptr, &print_rec, sizeof(print_rec));
    _Rwrite(prtfptr, "CTGTDMUL HAS COMPLETED NORMALLY  ", 33);
}


/*----------------------------------------------------------------------*/
/*                   Print Abnormal End Message                         */
/* Write abnormal end message and return code to print file.            */
/*----------------------------------------------------------------------*/
 15
void print_error_end()
{
    get_access_to_fb();
    strncpy(print_rec.major, dsp_icf_fdbk->major_ret_code, 2);
    strncpy(print_rec.minor, dsp_icf_fdbk->minor_ret_code, 2);
    strncpy(print_rec.filler, "                             ", 29);
    _Rwrite(prtfptr, "RETURN CODE:                     ", 33);
    _Rwrite(prtfptr, &print_rec, sizeof(print_rec));
    _Rwrite(prtfptr, "CTGTDMUL HAS COMPLETED ABNORMALLY", 33);
}


/*----------------------------------------------------------------------*/
/*                     Get Access to DSP/ICF Feedback                   */
/* The feedback areas are updated after each display or ICF file I/O    */
/* operation, and so the pointers must be updated to point to the "new" */
/* feedback areas to get the return code.  The offset to the display/   */
/* ICF feedback area is contained in the common I/O feedback and is     */
/* added to the pointer to the common feedback area to get access to    */
/* display/ICF feedback area.                                           */
/*----------------------------------------------------------------------*/
 16
void get_access_to_fb()
{
    comm_fdbk = _Riofbk(icffptr);
    dsp_icf_fdbk = (_XXIOFB_DSP_ICF_T *)((char *)comm_fdbk +
                                        comm_fdbk->file_dep_fb_offset);
}
                      * * * * *   E N D   O F   S O U R C E   * * * * *
```

Figure 9-10 (Part 5 of 6). Target Program Example — CTGTDMUL (User-Defined Formats)

```
                                              * * * * *   I N C L U D E S   * * * * *
INCNO   Include Name                     Last change         Actual Include Name
    1   icff/snd                                             CICFLIB/cfile(sndpart)
    2   icff/rcv                                             CICFLIB/cfile(rcvpart)
    3   stdio.h                          90/09/12 14:42:23  QCC/H/STDIO
    4   stddef.h                         90/09/12 14:42:22  QCC/H/STDDEF
    5   errno.h                          90/09/12 14:42:18  QCC/H/ERRNO
    6   signal.h                         90/09/12 14:42:22  QCC/H/SIGNAL
    7   ctype.h                          90/09/12 14:42:18  QCC/H/CTYPE
    8   stdarg.h                         90/09/12 14:42:22  QCC/H/STDARG
    9   recio.h                          90/09/12 14:42:21  QCC/H/RECIO
   10   xxfdbk.h                         90/09/12 14:42:28  QCC/H/XXFDBK
   11   stdlib.h                         90/09/12 14:42:24  QCC/H/STDLIB
   12   stddef.h                         90/09/12 14:42:22  QCC/H/STDDEF
   13   string.h                         90/09/12 14:42:24  QCC/H/STRING
   14   xxfdbk.h                         90/09/12 14:42:28  QCC/H/XXFDBK
                                      * * * * *   E N D   O F   I N C L U D E S   * * * * *

                                       * * * * *   M E S S A G E   S U M M A R Y   * * * * *
     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   M E S S A G E   S U M M A R Y   * * * * *
 ROUTINE          BLOCK NUMBER  SCOPE  TYPE
 <MAIN>                 2       LOCAL  MAIN-PROGRAM
 __reads              115       LOCAL  PROCEDURE
 __readk              117       LOCAL  PROCEDURE
 __rwrite             118       LOCAL  PROCEDURE
 __rfmt               125       LOCAL  PROCEDURE
 __memset             172       LOCAL  PROCEDURE
 __strncmp            190       LOCAL  PROCEDURE
 __strncpy            192       LOCAL  PROCEDURE
 read_cfile           213       ENTRY   PROCEDURE
 send_data            214       ENTRY   PROCEDURE
 check_ret_code       215       ENTRY   PROCEDURE
 pos_ret_code         216       ENTRY   PROCEDURE
 close_files          217       ENTRY   PROCEDURE
 print_norm_end       218       ENTRY   PROCEDURE
 print_error_end      219       ENTRY   PROCEDURE
 get_access_to_fb     220       ENTRY   PROCEDURE
 main                 221       ENTRY   PROCEDURE
                                   * * * * *   E N D   O F   C O M P I L A T I O N   * * * * *
```

Figure 9-10 (Part 6 of 6). Target Program Example — CTGTDMUL (User-Defined Formats)

Chapter 10. COBOL/400 Communications Applications

Previous chapters in this book 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 book 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 the COBOL/400 programming language.

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 the COBOL/400 programming language. 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.

Figure 10-1 briefly introduces the COBOL/400 statements you use in the communications portion of your program.

Figure 10-1. 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
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 book 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 the *ILE COBOL/400 Reference* for details on the syntax and function of each operation.

Error Handling: The FILE STATUS clause of the FILE-CONTROL paragraph is used in COBOL/400 programming to specify the variables for the COBOL file status and the ICF major and minor return codes.

Figure 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.

Figure 10-2 (Page 1 of 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

Figure 10-2 (Page 2 of 2). File Status Values for Major and Minor Return Codes

ICF Return Code	COBOL/400 Return Code
83E0	9K

Accessing the Feedback Areas: Use the COBOL/400 ACCEPT statement to obtain the open or I/O feedback information for an ICF file.

The COBOL/400 offset values for the open and I/O feedback areas are the same as those listed in Appendix C.

Example Programs

The programs presented in this section are:

- Example I (Batch Data Transfer)

  Figure 10-3 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.

- Example II (Multiple-Session Inquiry)

  Figure 10-4 on page 10-3 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 10-4 on page 10-3 contains a display device and four ICF communication program devices.

The remainder of this chapter discusses the details of the two application examples. The DDS file, program listings, and an explanation of the programs are included.

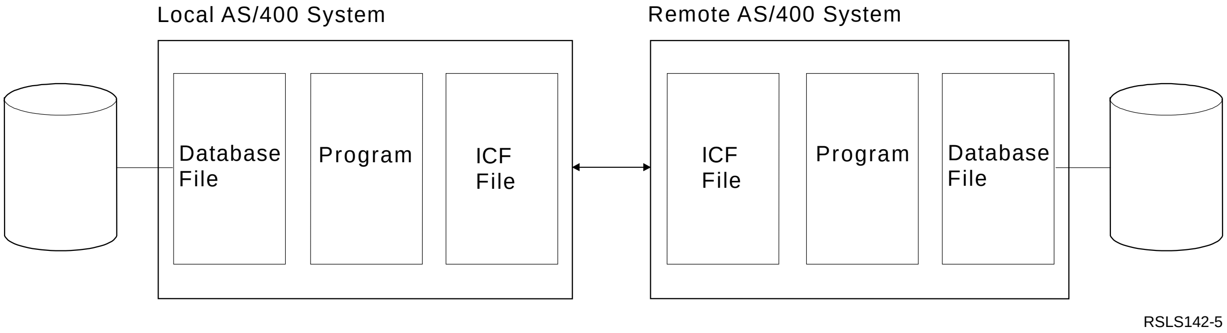


Figure 10-3. Batch Data Transfer

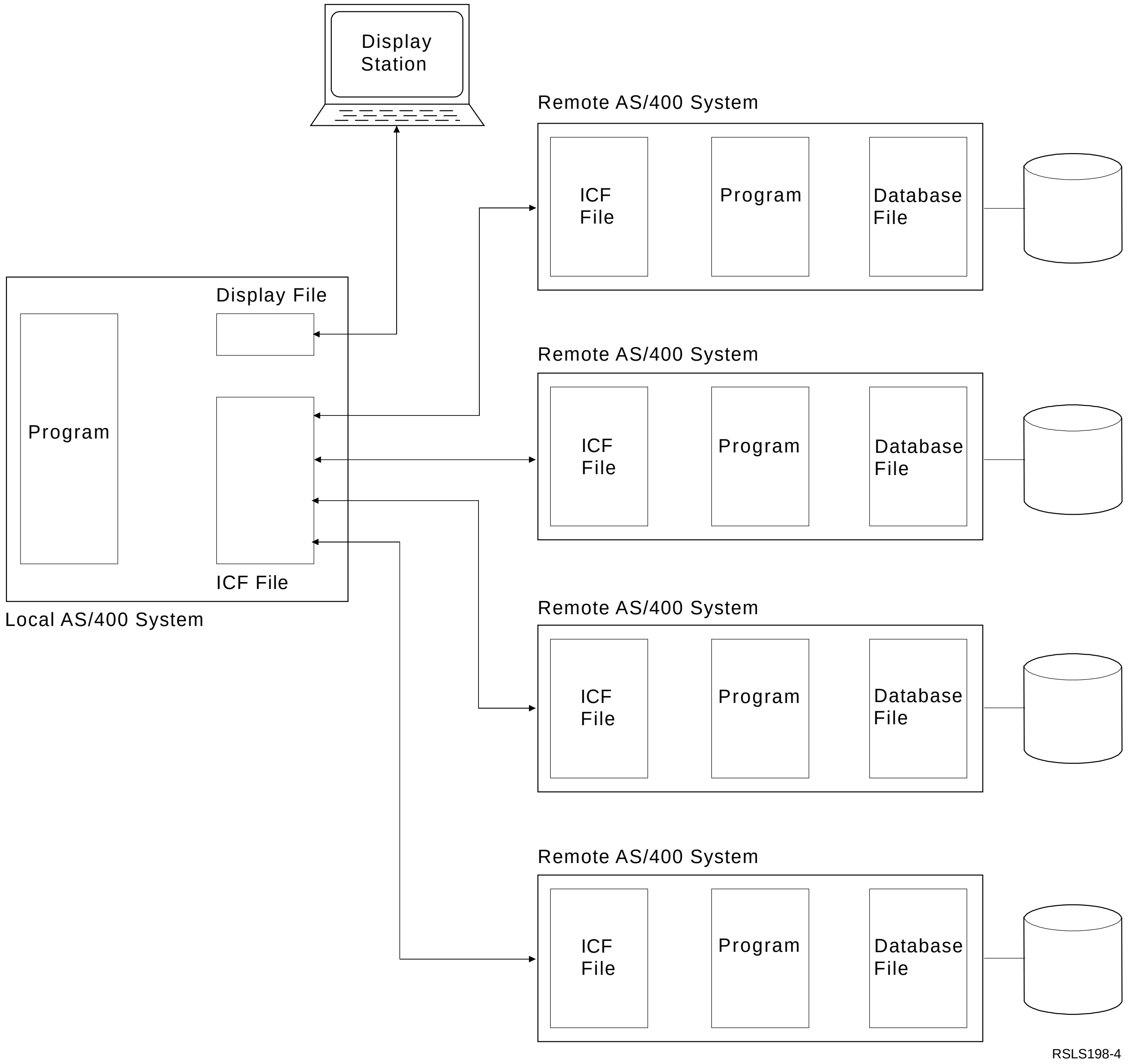


Figure 10-4. Multiple-Session Inquiry

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 10-5, 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.

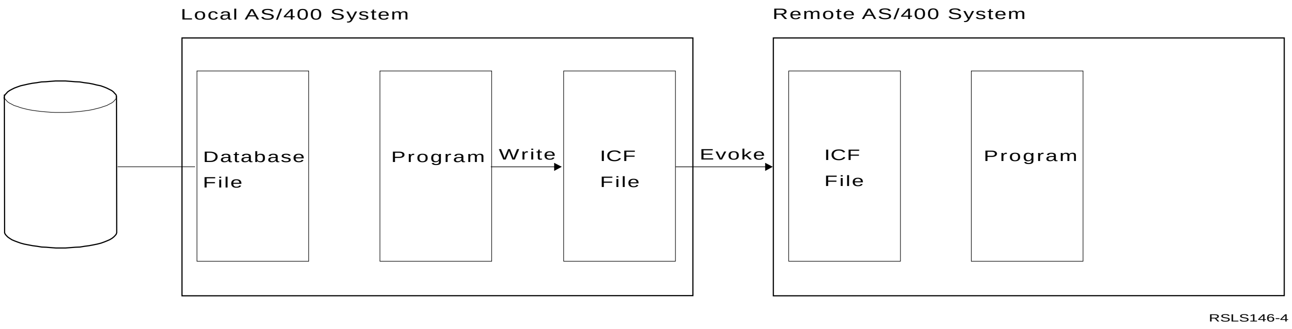


Figure 10-5. Evoke Request Starts a Target Program

After issuing the evoke request, the source program sends a database file to the target program, which prints the records as shown in Figure 10-6.

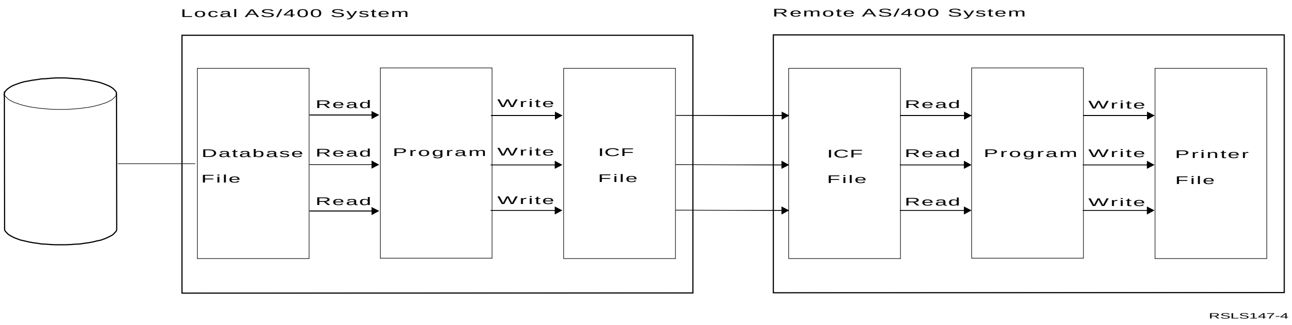


Figure 10-6. Target Program Prints Records

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 10-7.

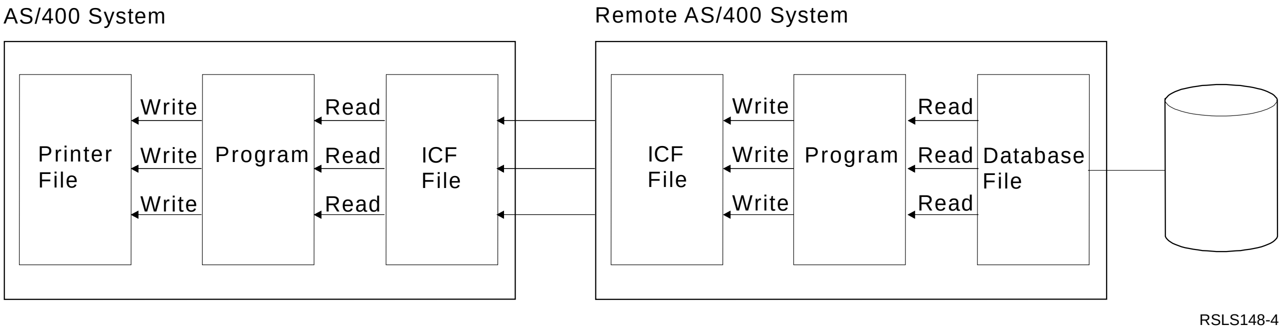


Figure 10-7. Source Program Prints the Received Records

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 10-8 on page 10-5.

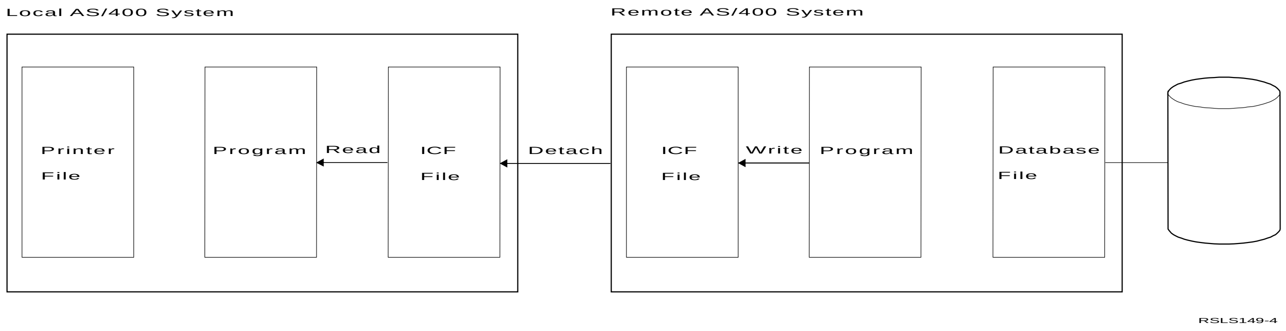


Figure 10-8. Target Program Ends the Transaction

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 the following example. 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
```

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 10-9 on page 10-7 and Figure 10-10 on page 10-13. 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 printer 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 printer 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.

```
Program . . . . . . . . . . . . . . :   CSDBAT
  Library . . . . . . . . . . . . . :     ICFLIB
Source file . . . . . . . . . . . . :   QICFPUB
  Library . . . . . . . . . . . . . :     ICFLIB
Source member . . . . . . . . . . . :   CSDBAT     02/28/89 13:51:38
Generation severity level . . . . . :   29
Text 'description' . . . . . . . . . :   cobol batch file transfer using dds (source)
Source listing options . . . . . . . :   *SOURCE
Generation options . . . . . . . . . :   *NONE
Message limit:
  Number of messages . . . . . . . . :   *NOMAX
  Message limit severity . . . . . . :   29
Print file . . . . . . . . . . . . . :   QSYSPRT
  Library . . . . . . . . . . . . . :     *LIBL
FIPS flagging . . . . . . . . . . . :   *NOFIPS *NOSEG *NODEB *NOOBSOLETE
SAA flagging . . . . . . . . . . . . :   *NOFLAG
Flagging severity . . . . . . . . . :   0
Replace program . . . . . . . . . . :   *YES
Target release . . . . . . . . . . . :   *CURRENT
User profile . . . . . . . . . . . . :   *USER
Authority . . . . . . . . . . . . . :   *LIBCRTAUT
Compiler . . . . . . . . . . . . . . :   IBM AS/400 COBOL/400

    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       *                   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
```

Figure 10-9 (Part 1 of 6). Source Program Example — CSDBAT (User-Defined Formats)

```
  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
   +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
   +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
```

Figure 10-9 (Part 2 of 6). Source Program Example — CSDBAT (User-Defined Formats)

```
 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
     006500/                                                                            10/16/87
 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 XPF-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 DATABASE 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 "QCMDEXC" USING CMD2 LEN.                                          10/16/87
 81  009700     MOVE "PROGRAM TERMINATED DUE TO ERROR IN XPF-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
```

Figure 10-9 (Part 3 of 6). Source Program Example — CSDBAT (User-Defined Formats)

```
     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
  90 012500              PERFORM ERROR-RECOVERY-RTN                                                 10/16/87
  91 012600              GO TO START-PROGRAM-PARAGRAPH                                              10/16/87
     012700         ELSE                                                                            10/16/87
  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
 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
```

Figure 10-9 (Part 4 of 6). Source Program Example — CSDBAT (User-Defined Formats)

```
     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
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
     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   * * * * *
```

Figure 10-9 (Part 5 of 6). Source Program Example — CSDBAT (User-Defined Formats)

```
*   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.
                        * * * * *   E N D   O F   M E S S A G E S   * * * * *
                                           Message Summary
  Total    Info(0-4)    Warning(5-19)    Error(20-29)    Severe(30-39)    Terminal(40-99)
      7         2            5               0               0                 0
 Source records read . . . . . . . . :   236
 Copy records read . . . . . . . . . :   37
 Copy members processed  . . . . . . :   1
 Sequence errors . . . . . . . . . . :   0
 Highest severity message issued . . :   10
  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 10-9 (Part 6 of 6). Source Program Example — CSDBAT (User-Defined Formats)

```
Program . . . . . . . . . . . . . . . :   CSFBAT
  Library . . . . . . . . . . . . . . :     ICFLIB
Source file . . . . . . . . . . . . . :   QICFPUB
  Library . . . . . . . . . . . . . . :     ICFLIB
Source member . . . . . . . . . . . . :   CSFBAT     10/03/90 14:27:28
Generation severity level . . . . . . :   29
Text 'description' . . . . . . . . . :   cobol batch file transfer using $$FORMAT (source)
Source listing options . . . . . . . :   *SOURCE
Generation options . . . . . . . . . :   *NONE
Message limit:
  Number of messages . . . . . . . . :   *NOMAX
  Message limit severity . . . . . . :   29
Print file . . . . . . . . . . . . . :   QSYSPRT
  Library . . . . . . . . . . . . . . :     *LIBL
FIPS flagging . . . . . . . . . . . . :   *NOFIPS *NOSEG *NODEB *NOOBSOLETE
SAA flagging . . . . . . . . . . . . :   *NOFLAG
Flagging severity . . . . . . . . . . :   0
Replace program . . . . . . . . . . . :   *YES
Target release . . . . . . . . . . . :   *CURRENT
User profile . . . . . . . . . . . . :   *USER
Authority . . . . . . . . . . . . . . :   *LIBCRTAUT
Compiler . . . . . . . . . . . . . . :   IBM AS/400 COBOL/400

    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       *                   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 10-10 (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
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
    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.                                            10/03/90
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 DATABASE 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 "QCMDEXC" USING CMD2 LEN.                                                      10/16/87
75  010300     MOVE "PROGRAM TERMINATED DUE TO ERROR IN ICF FILE"                                  10/03/90
    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
    011000*                                                                                        10/16/87
78  011100 END DECLARATIVES.                                                                       10/16/87
    011200/                                                                                        10/16/87
```

Figure 10-10 (Part 2 of 5). Source Program Example — CSFBAT (System-Supplied Formats)

```
      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
  84  013100              PERFORM ERROR-RECOVERY-RTN                                10/16/87
  85  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
 101  016700         PERFORM WRITE-SRCICF-RTN                                       10/16/87
 102  016800         GO TO SEND-DATA.                                               10/16/87
      016900*                                                                       10/16/87
```

Figure 10-10 (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      *                     10/03/90
     022100*   THE ERROR IS PRINTED.                                          *                     02/21/89
     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 10-10 (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   * * * * *
*   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.
                        * * * * *   E N D   O F   M E S S A G E S   * * * * *
                                         Message Summary
  Total    Info(0-4)    Warning(5-19)    Error(20-29)    Severe(30-39)    Terminal(40-99)
      2         2            0               0               0                 0
 Source records read . . . . . . . . :   241
 Copy records read . . . . . . . . . :   0
 Copy members processed  . . . . . . :   0
 Sequence errors . . . . . . . . . . :   0
 Highest severity message issued . . :   0
  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 10-10 (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
```

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 10-19 and Figure 10-12 on page 10-23. 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 printer 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 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 printer 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.

```
Program . . . . . . . . . . . . . . . :   CTDBAT
  Library . . . . . . . . . . . . . . :     ICFLIB
Source file . . . . . . . . . . . . . :   QICFPUB
  Library . . . . . . . . . . . . . . :     ICFLIB
Source member . . . . . . . . . . . . :   CTDBAT     02/21/89 17:51:20
Generation severity level . . . . . . :   29
Text 'description' . . . . . . . . . :   cobol batch file transfer using dds (target)
Source listing options . . . . . . . :   *SOURCE
Generation options . . . . . . . . . :   *NONE
Message limit:
  Number of messages . . . . . . . . :   *NOMAX
  Message limit severity . . . . . . :   29
Print file . . . . . . . . . . . . . :   QSYSPRT
  Library . . . . . . . . . . . . . . :     *LIBL
FIPS flagging . . . . . . . . . . . . :   *NOFIPS *NOSEG *NODEB *NOOBSOLETE
SAA flagging . . . . . . . . . . . . :   *NOFLAG
Flagging severity . . . . . . . . . . :   0
Replace program . . . . . . . . . . . :   *YES
Target release . . . . . . . . . . . :   *CURRENT
User profile . . . . . . . . . . . . :   *USER
Authority . . . . . . . . . . . . . . :   *LIBCRTAUT
Compiler . . . . . . . . . . . . . . :   IBM AS/400 COBOL/400

    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
      +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
```

Figure 10-11 (Part 1 of 5). Target Program Example — CTDBAT (User-Defined Formats)

```
      +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
      +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
       006200/                                                                                      10/16/87
   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
```

Figure 10-11 (Part 2 of 5). Target Program Example — CTDBAT (User-Defined Formats)

```
    007100********************************************************************                     02/21/89
    007200* GET INFORMATION FROM THE MAJOR-MINOR CODE AND PLACE IT INTO      *                     02/21/89
    007300* A DATABASE 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 "QCMDEXC" 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
    009200/                                                                                        10/16/87
```

Figure 10-11 (Part 3 of 5). Target Program Example — CTDBAT (User-Defined Formats)

```
    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
    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
```

Figure 10-11 (Part 4 of 5). Target Program Example — CTDBAT (User-Defined Formats)

```
        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
        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  * * * * *
*   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.
                        * * * * *  E N D   O F   M E S S A G E S  * * * * *
                                          Message Summary
  Total    Info(0-4)    Warning(5--19)    Error(20-29)    Severe(30-39)    Terminal(40-99)
      8          3            5                0                0                 0
 Source records read . . . . . . . . :   149
 Copy records read . . . . . . . . . :   37
 Copy members processed  . . . . . . :   1
 Sequence errors . . . . . . . . . . :   0
 Highest severity message issued . . :   10
  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 5 of 5). Target Program Example — CTDBAT (User-Defined Formats)

```
Program . . . . . . . . . . . . . . . :   CTFBAT
  Library . . . . . . . . . . . . . . :     ICFLIB
Source file . . . . . . . . . . . . . :   QICFPUB
  Library . . . . . . . . . . . . . . :     ICFLIB
Source member . . . . . . . . . . . . :   CTFBAT     02/27/89 09:38:46
Generation severity level . . . . . . :   29
Text 'description' . . . . . . . . . . :   cobol batch file transfer using $$FORMAT (target)
Source listing options . . . . . . . . :   *SOURCE
Generation options . . . . . . . . . . :   *NONE
Message limit:
  Number of messages . . . . . . . . . :   *NOMAX
  Message limit severity . . . . . . . :   29
Print file . . . . . . . . . . . . . . :   QSYSPRT
  Library . . . . . . . . . . . . . . :     *LIBL
FIPS flagging . . . . . . . . . . . . :   *NOFIPS *NOSEG *NODEB *NOOBSOLETE
SAA flagging . . . . . . . . . . . . . :   *NOFLAG
Flagging severity . . . . . . . . . . :   0
Replace program . . . . . . . . . . . :   *YES
Target release . . . . . . . . . . . . :   *CURRENT
User profile . . . . . . . . . . . . . :   *USER
Authority . . . . . . . . . . . . . . :   *LIBCRTAUT
Compiler . . . . . . . . . . . . . . . :   IBM AS/400 COBOL/400

     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
    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
```

Figure 10-12 (Part 1 of 3). Target Program Example — CTFBAT (System-Supplied Formats)

```
  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
  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
      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 DATABASE 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 "QCMDEXC" 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
      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
      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
```

Figure 10-12 (Part 2 of 3). Target Program Example — CTFBAT (System-Supplied Formats)

```
        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* 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
    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   * * * * *

*   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.
                        * * * * *   E N D   O F   M E S S A G E S   * * * * *
                                          Message Summary
  Total    Info(0-4)    Warning(5-19)    Error(20-29)    Severe(30-39)    Terminal(40-99)
      3         3            0                0               0                 0
 Source records read . . . . . . . . :   152
 Copy records read . . . . . . . . . :   0
 Copy members processed  . . . . . . :   0
 Sequence errors . . . . . . . . . . :   0
 Highest severity message issued . . :   0
  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 3 of 3). 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.

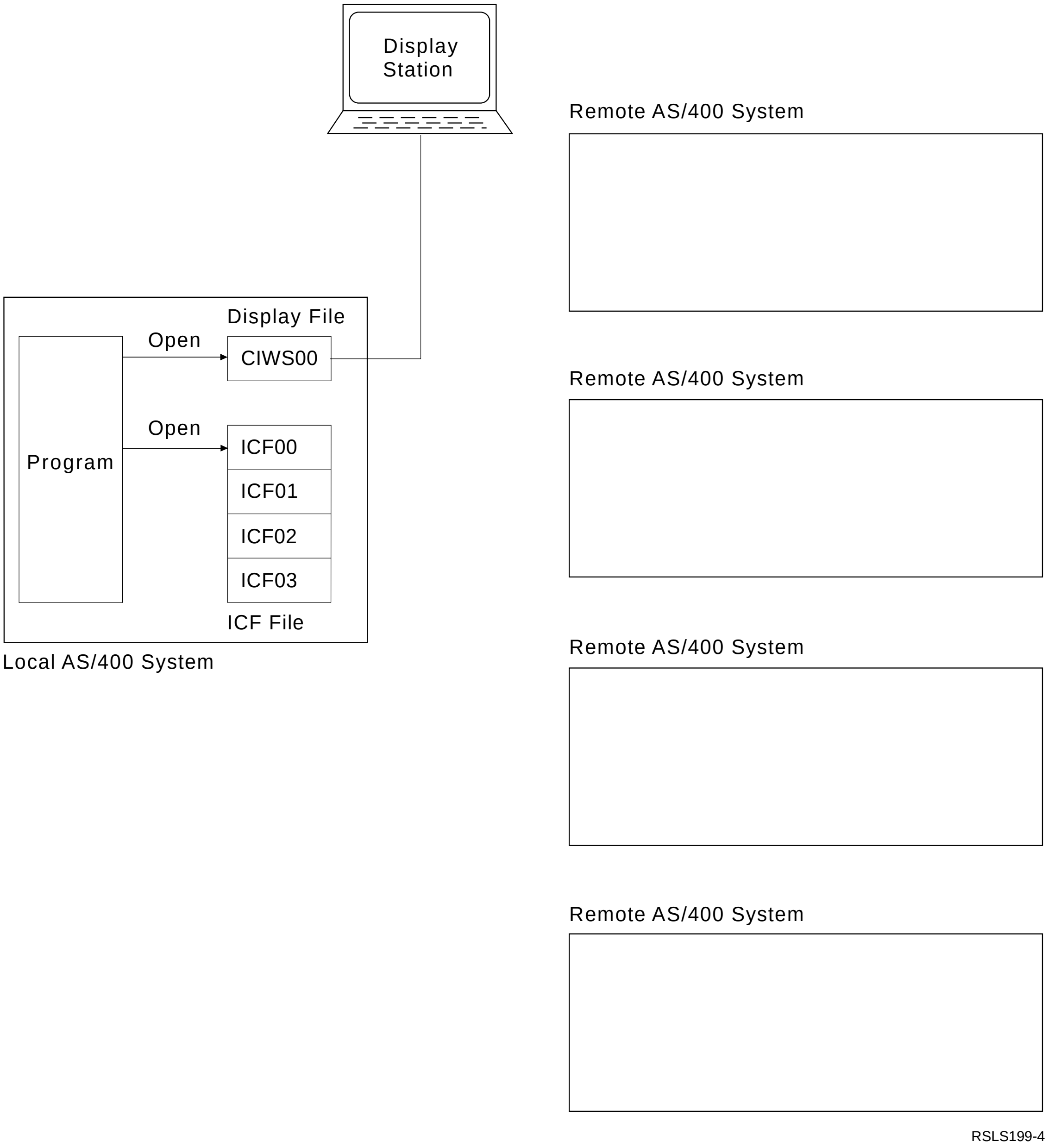


Figure 10-13. Program Starts at Display Station

All other program devices must be explicitly acquired by the program, as shown in Figure 10-14.

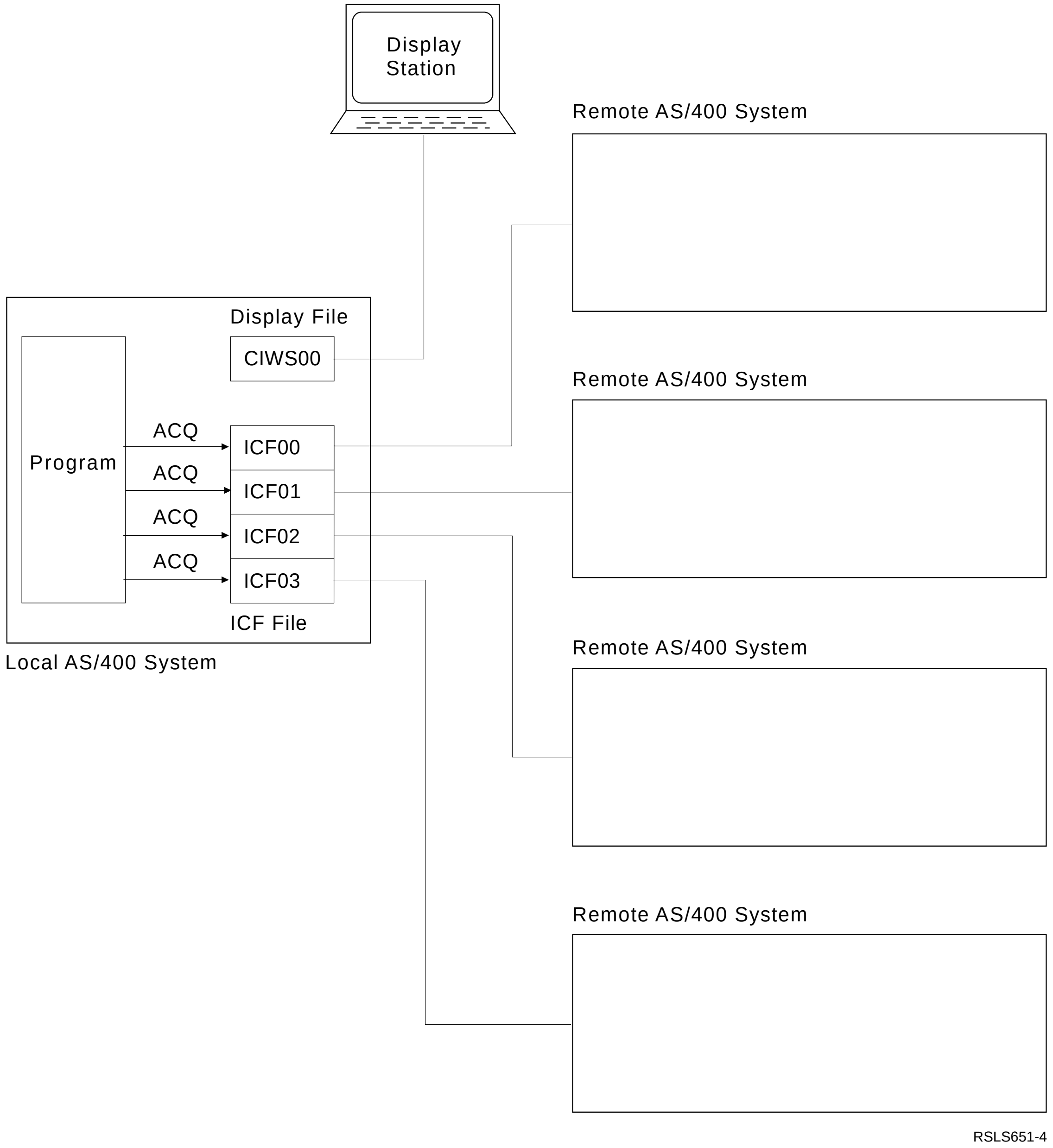


Figure 10-14. Program Devices Explicitly Acquired

All target programs are started with the evoke, as shown in Figure 10-15.

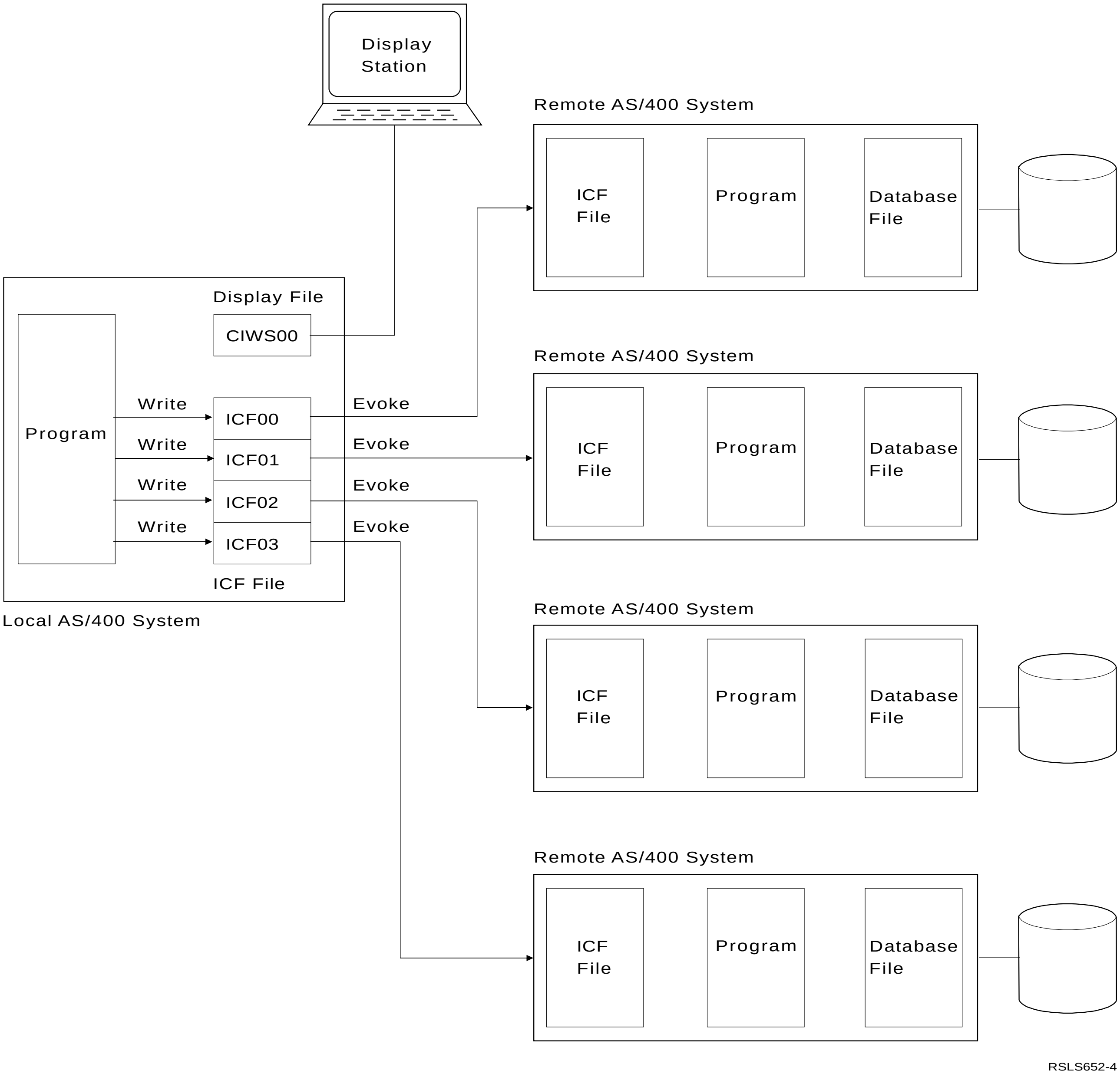


Figure 10-15. Evoke Starts Target Programs

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.

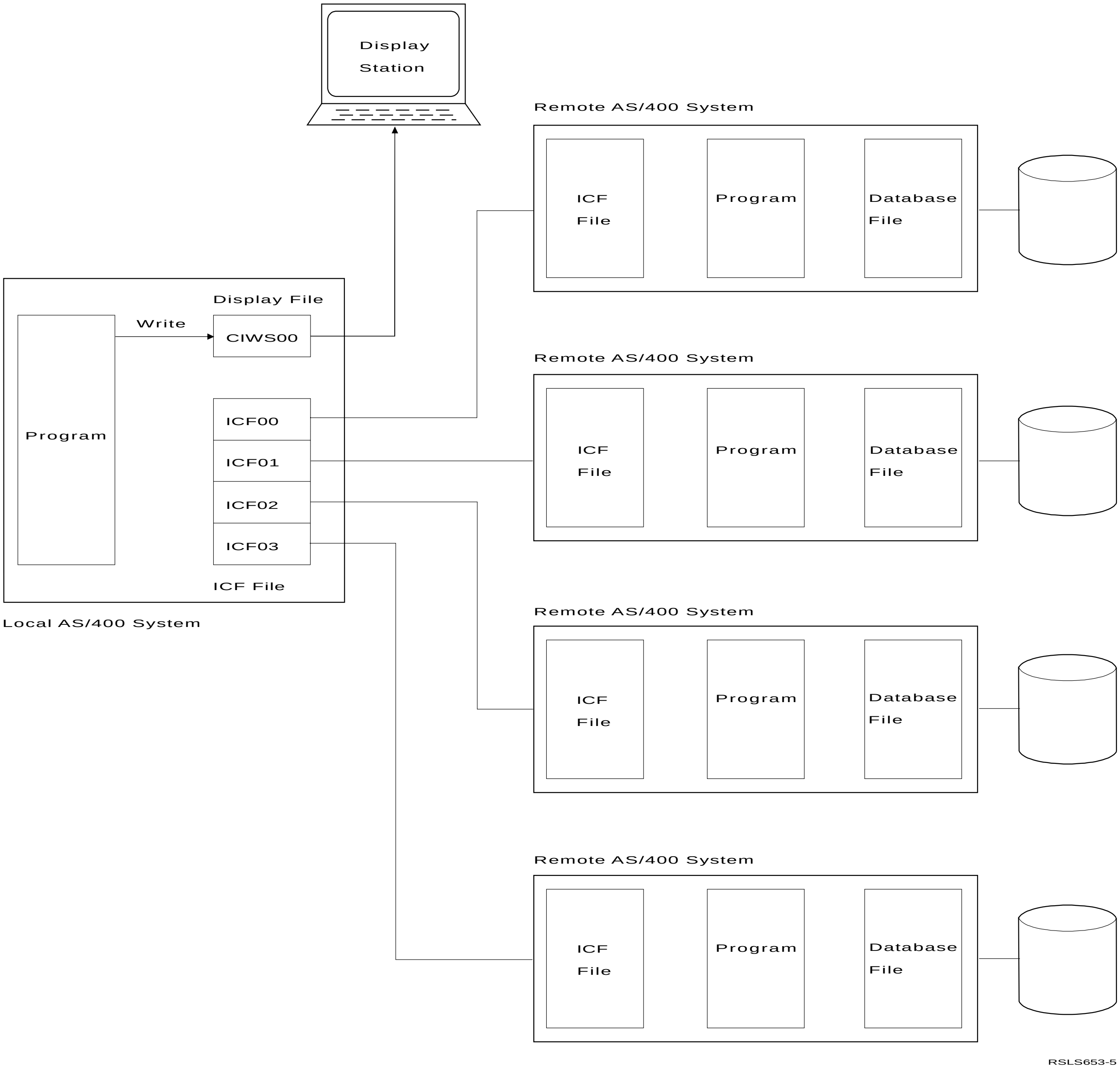


Figure 10-16. Main Menu Written to Display Station

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.

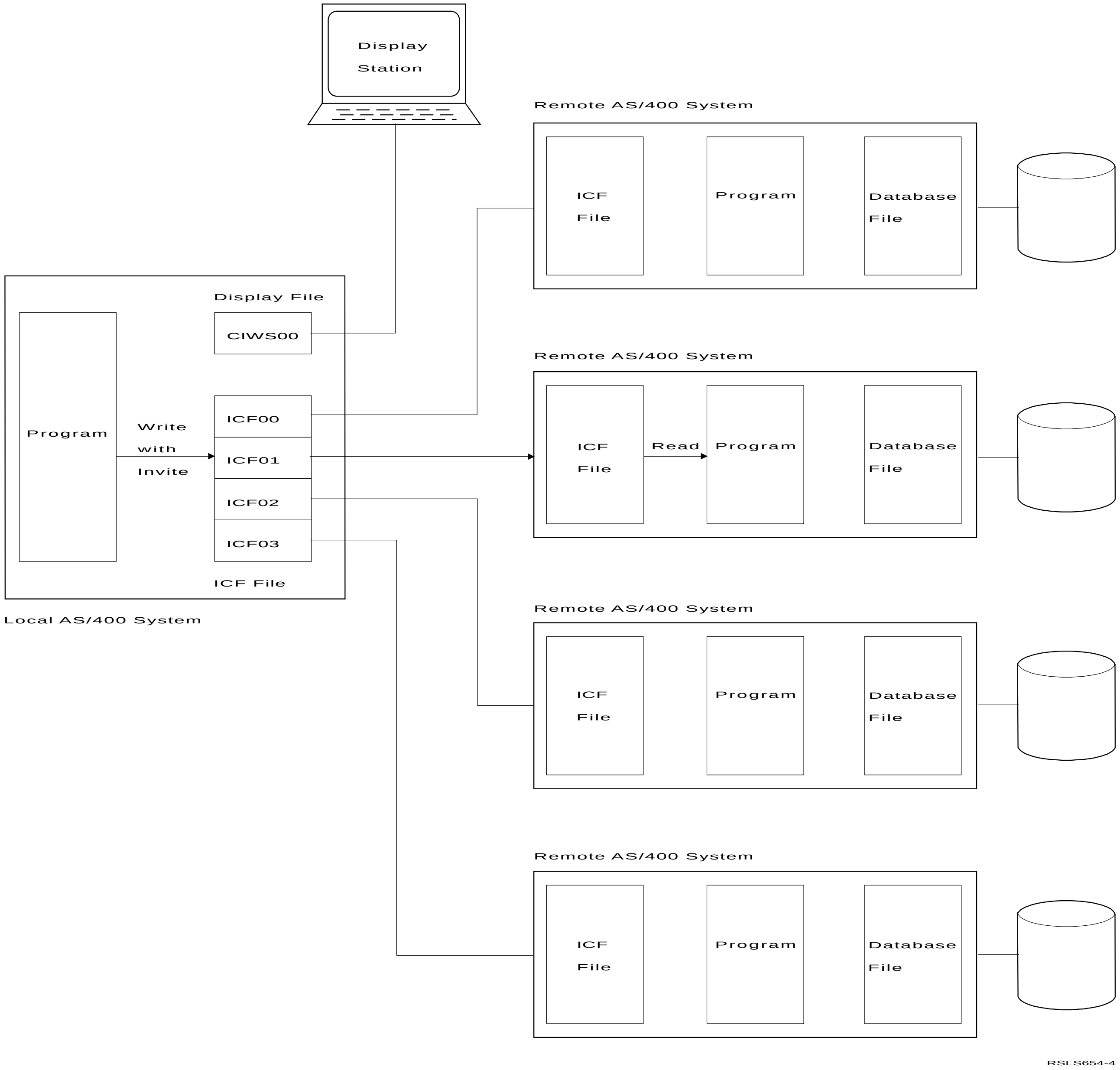


Figure 10-17. Program Sends Inquiry Request to Remote System

The target program responds to the inquiry by sending a reply, as shown in Figure 10-18.

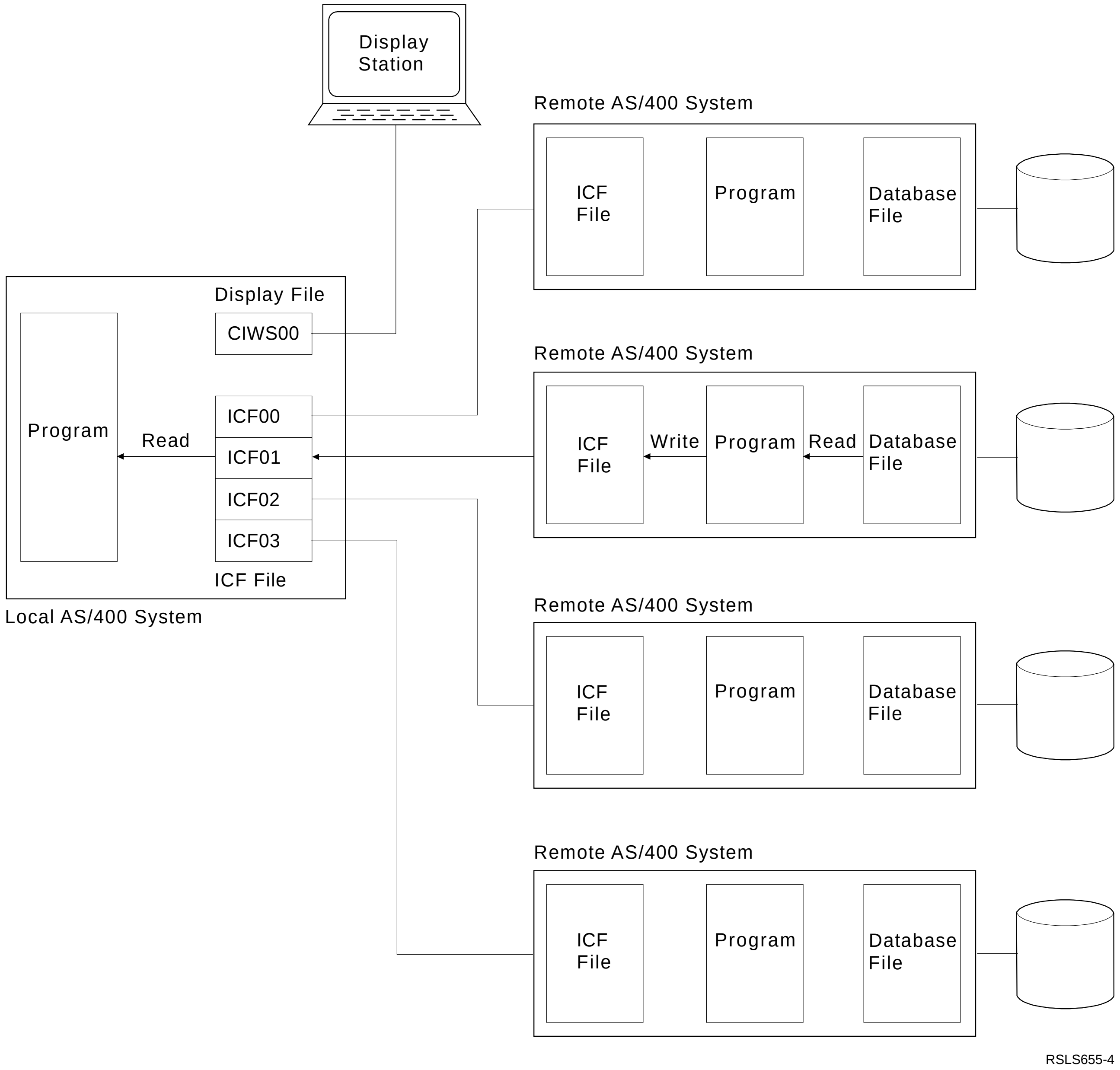


Figure 10-18. Target Program Sends a Reply

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.

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.

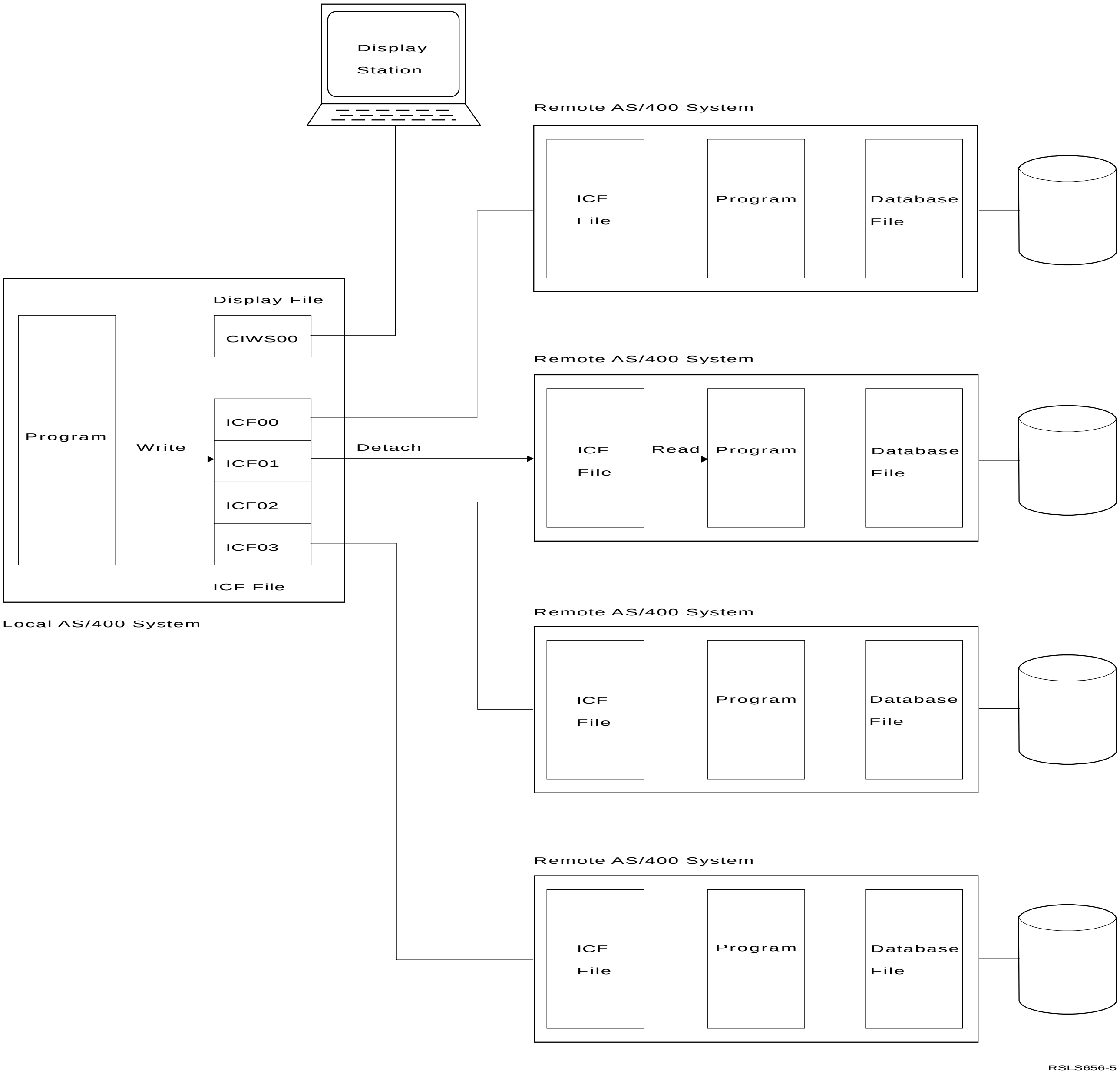


Figure 10-19. Program Ends the Session

DDS Source: The DDS for the ICF file (CMNFIL) is illustrated in Figure 10-20.

```
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
 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
003100871008     A                                       THIS    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')
005600871007     A                                      OVERLAY
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
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 ')
```

Figure 10-21 (Part 1 of 2). DDS for Source Program Multiple Session Inquiry Using DSPFIL

```
007900871008     A****************************
008000871008     A*  ITEM ADDITIONAL DISPLAY
008100871008     A****************************
008200871007     A          R ITMSC3                    TEXT('ITEM INQUIRY SCREEN 3  ')
008300871007     A                                      OVERLAY
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')
010200871007     A                                      OVERLAY
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 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 10-37 and Figure 10-23 on page 10-51. 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 printer file. 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 not valid 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 not valid 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 run when the program detects data in an incorrect record format. It writes an error message to the printer file, 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.

```
Program . . . . . . . . . . . . . . . :   CSDMUL
  Library . . . . . . . . . . . . . . :     ICFLIB
Source file . . . . . . . . . . . . . :   QICFPUB
  Library . . . . . . . . . . . . . . :     ICFLIB
Source member . . . . . . . . . . . . :   CSDMUL     10/03/90 14:28:28
Generation severity level  . . . . . :   29
Text 'description' . . . . . . . . . :   CBL Multiple Session Inquiry - Source DDS
Source listing options . . . . . . . :   *SOURCE
Generation options . . . . . . . . . :   *NONE
Message limit:
  Number of messages . . . . . . . . :   *NOMAX
  Message limit severity . . . . . . :   29
Print file . . . . . . . . . . . . . :   QSYSPRT
  Library . . . . . . . . . . . . . . :     *LIBL
FIPS flagging  . . . . . . . . . . . :   *NOFIPS *NOSEG *NODEB *NOOBSOLETE
SAA flagging . . . . . . . . . . . . :   *NOFLAG
Flagging severity  . . . . . . . . . :   0
Replace program  . . . . . . . . . . :   *YES
Target release . . . . . . . . . . . :   *CURRENT
User profile . . . . . . . . . . . . :   *USER
Authority  . . . . . . . . . . . . . :   *LIBCRTAUT
Compiler . . . . . . . . . . . . . . :   IBM AS/400 COBOL/400

    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
```

Figure 10-22 (Part 1 of 14). Source Program Example — CSDMUL (User-Defined Formats)

```
 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             *                    10/03/90
    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
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
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
```

Figure 10-22 (Part 2 of 14). Source Program Example — CSDMUL (User-Defined Formats)

```
    +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
    +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
 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
```

Figure 10-22 (Part 3 of 14). Source Program Example — CSDMUL (User-Defined Formats)

```
    +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
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
    +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
```

Figure 10-22 (Part 4 of 14). Source Program Example — CSDMUL (User-Defined Formats)

```
    +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
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
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
     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
```

Figure 10-22 (Part 5 of 14). Source Program Example — CSDMUL (User-Defined Formats)

```
     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.                                            10/03/90
204  014200     IF MAJ = "83"                                                                      09/30/87
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 DATABASE                   *                     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
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
     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
```

Figure 10-22 (Part 6 of 14). Source Program Example — CSDMUL (User-Defined Formats)

```
     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.                                *                     10/03/90
     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
223  018800                PERFORM ERROR-RECOVERY                                                   09/30/87
224  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.        *                     10/03/90
     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
     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
     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
```

Figure 10-22 (Part 7 of 14). Source Program Example — CSDMUL (User-Defined Formats)

```
     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
     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
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
     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, AN 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
```

Figure 10-22 (Part 8 of 14). Source Program Example — CSDMUL (User-Defined Formats)

```
     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 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
           *       ** CORRESPONDING items for statement 266:
           *       **     ITEMNO
           *       ** End of CORRESPONDING items for statement 266
 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
 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
 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
     034700/                                                                                        10/14/87
```

Figure 10-22 (Part 9 of 14). Source Program Example — CSDMUL (User-Defined Formats)

```
     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
310  040100        GO TO END-JOB.                                                                   10/12/87
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
```

Figure 10-22 (Part 10 of 14). Source Program Example — CSDMUL (User-Defined Formats)

```
     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 2 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
     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
           *       ** CORRESPONDING items for statement 341:
           *       **     CUSTNO
           *       ** End of CORRESPONDING items for statement 341
342  046200     MOVE ZEROS TO CMNF-INDIC-AREA.                                                      09/30/87
343  046300     WRITE CMNREC FORMAT IS "DTLREQ"                                                     09/30/87
     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
```

Figure 10-22 (Part 11 of 14). Source Program Example — CSDMUL (User-Defined Formats)

```
     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
     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.                                       *                     10/03/90
     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
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
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
     055400/                                                                                        09/30/87
```

Figure 10-22 (Part 12 of 14). Source Program Example — CSDMUL (User-Defined Formats)

```
     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
     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
     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
```

Figure 10-22 (Part 13 of 14). Source Program Example — CSDMUL (User-Defined Formats)

```
       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   * * * * *

*   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.
*   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.
                * * * * *   E N D   O F   M E S S A G E S   * * * * *
                                 Message Summary
  Total    Info(0-4)    Warning(5-19)    Error(20-29)    Severe(30-39)    Terminal(40-99)
     7         0            7               0               0                0
 Source records read . . . . . . . . :   652
 Copy records read . . . . . . . . . :   171
 Copy members processed  . . . . . . :   2
 Sequence errors . . . . . . . . . . :   0
 Highest severity message issued . . :   10
  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)

```
Program . . . . . . . . . . . . . . . :   CSFMUL
  Library . . . . . . . . . . . . . . :     ICFLIB
Source file . . . . . . . . . . . . . :   QICFPUB
  Library . . . . . . . . . . . . . . :     ICFLIB
Source member . . . . . . . . . . . . :   CSFMUL     10/03/90 14:26:27
Generation severity level . . . . . . :   29
Text 'description' . . . . . . . . . :   CBL Multiple Session Inquiry - Source $$
Source listing options . . . . . . . :   *SOURCE
Generation options . . . . . . . . . :   *NONE
Message limit:
  Number of messages . . . . . . . . :   *NOMAX
  Message limit severity . . . . . . :   29
Print file . . . . . . . . . . . . . :   QSYSPRT
  Library . . . . . . . . . . . . . . :     *LIBL
FIPS flagging . . . . . . . . . . . . :   *NOFIPS *NOSEG *NODEB *NOOBSOLETE
SAA flagging . . . . . . . . . . . . :   *NOFLAG
Flagging severity . . . . . . . . . . :   0
Replace program . . . . . . . . . . . :   *YES
Target release . . . . . . . . . . . :   *CURRENT
User profile . . . . . . . . . . . . :   *USER
Authority . . . . . . . . . . . . . . :   *LIBCRTAUT
Compiler . . . . . . . . . . . . . . :   IBM AS/400 COBOL/400

    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             *                  10/03/90
       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
```

Figure 10-23 (Part 1 of 14). Source Program Example — CSFMUL (System-Supplied Formats)

```
003800     SELECT CMNFIL ASSIGN TO WORKSTATION-CMNFIL-SI                                       10/08/87
003900         ORGANIZATION IS TRANSACTION                                                     10/01/87
004000         CONTROL-AREA IS TR-CTL-AREA                                                     10/01/87
004100         FILE STATUS IS STATUS-IND MAJ-MIN.                                              10/01/87
004200     SELECT DSPFIL ASSIGN TO WORKSTATION-DSPFIL                                          10/08/87
004300         ORGANIZATION IS TRANSACTION                                                     10/01/87
004400         CONTROL-AREA IS DISPLAY-FEEDBACK                                                10/01/87
004500         FILE STATUS IS STATUS-DSP.                                                      10/01/87
004600     SELECT QPRINT ASSIGN TO PRINTER-QSYSPRT.                                            10/01/87
004700 DATA DIVISION.                                                                          10/01/87
004800 FILE SECTION.                                                                           10/01/87
004900 FD  CMNFIL                                                                              10/08/87
005000     LABEL RECORDS ARE STANDARD.                                                         10/01/87
005100 01  CMNREC.                                                                             10/01/87
005200     05  ITMRSP.                                                                         10/01/87
005300         07 RECITM                    PIC X.                                             10/01/87
005400         07 ITEMNO                    PIC 9(6).                                          10/01/87
005500         07 DESC                      PIC X(30).                                         10/01/87
005600         07 QTYLST                    PIC 9(7).                                          10/01/87
005700         07 QTYOH                     PIC 9(7).                                          10/01/87
005800         07 QTYOO                     PIC 9(7).                                          10/01/87
005900         07 QTYBO                     PIC 9(7).                                          10/01/87
006000         07 UNITQ                     PIC 99.                                            10/01/87
006100         07 PR01                      PIC 9(5)V99.                                       10/01/87
006200         07 PR05                      PIC 9(7).                                          10/01/87
006300         07 UFRT                      PIC 999V99.                                        10/01/87
006400         07 SLSTM                     PIC 9(7)V99.                                       10/01/87
006500         07 SLSTY                     PIC 9(9)V99.                                       10/01/87
006600         07 CSTTM                     PIC 9(7)V99.                                       10/01/87
006700         07 CSTTY                     PIC 9(9)V99.                                       10/01/87
006800         07 PRO                       PIC 999V99.                                        10/01/87
006900         07 LOS                       PIC 9(7)V99.                                       10/01/87
007000         07 FILL1                     PIC X(56).                                         10/01/87
007100     05  ITMREQ REDEFINES ITMRSP.                                                        10/01/87
007200         07  LNGTH                    PIC 9(4).                                          02/22/89
007300         07  ITEMNO                   PIC 9(6).                                          10/01/87
007400     05  DTLREQ REDEFINES ITMRSP.                                                        10/01/87
007500         07  LNGTH                   PIC 9(4).                                           02/22/89
007600         07  CUSTNO                   PIC 9(6).                                          10/01/87
007700     05  DTLRSP REDEFINES ITMRSP.                                                        10/01/87
007800         07 RECCUS                    PIC X.                                             10/01/87
007900         07 CUSTNO                    PIC 9(6).                                          10/01/87
008000         07 DNAME                     PIC X(30).                                         10/01/87
008100         07 DLSTOR                    PIC 9(6).                                          10/01/87
008200         07 DSLSTM                    PIC 9(9).                                          10/01/87
008300         07 DSPM01                    PIC 9(9).                                          10/01/87
008400         07 DSPM02                    PIC 9(9).                                          10/01/87
008500         07 DSPM03                    PIC 9(9).                                          10/01/87
008600         07 DSTTYD                    PIC 9(11).                                         10/01/87
008700         07 IDEPT                     PIC 999.                                           10/01/87
008800         07 FILL2                     PIC X(57).                                         10/01/87
008900     05  EVKREQ REDEFINES ITMRSP.                                                        10/01/87
009000         07  PGMID                    PIC X(8).                                          10/01/87
009100         07  FILLER                   PIC X(16).                                         10/01/87
009200         07  LIB                      PIC X(8).                                          10/01/87
009300         07  FILLER                   PIC X(20).                                         10/01/87
009400         07  LNGTH                    PIC 9(4).                                          02/22/89
009500 FD  DSPFIL                                                                              10/08/87
009600     LABEL RECORDS ARE STANDARD.                                                         10/01/87
009700 01  DSPREC.                                                                             10/01/87
009800     COPY DDS-ALL-FORMATS-I-O OF DSPFIL.                                                 10/08/87
+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
+000004       05  CIMENU-I      REDEFINES DSPFIL-RECORD.                             <-ALL-FMTS
+000005           06 CIMENU-I-INDIC.                                                 <-ALL-FMTS
+000006                07 IN99             PIC 1  INDIC 99.                          <-ALL-FMTS
+000007                07 IN98             PIC 1  INDIC 98.                          <-ALL-FMTS
+000008                07 IN97             PIC 1  INDIC 97.                          <-ALL-FMTS
+000009           06 OPTION                PIC X(1).                                 <-ALL-FMTS
```

Figure 10-23 (Part 2 of 14). Source Program Example — CSFMUL (System-Supplied Formats)

```
    +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
 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
 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 DLSLM                 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
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
```

Figure 10-23 (Part 3 of 14). Source Program Example — CSFMUL (System-Supplied Formats)

```
    +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
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 LIBRARY ICFLIB             <-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
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
     014100/                                                                                        10/01/87
```

Figure 10-23 (Part 4 of 14). Source Program Example — CSFMUL (System-Supplied Formats)

```
181  014200 PROCEDURE DIVISION.                                                                     10/01/87
     014300 DECLARATIVES.                                                                           10/01/87
     014400* 2                                                                                      10/14/87
     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.                                            10/03/90
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 DATABASE 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
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
     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
     020900*                                                                                        10/01/87
```

Figure 10-23 (Part 5 of 14). Source Program Example — CSFMUL (System-Supplied Formats)

```
     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.                                       *                     10/03/90
     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
207  022100                PERFORM ERROR-RECOVERY                                                   10/01/87
208  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.        *                     10/03/90
     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 "CTFMULCL" ON REMOTE SYSTEM IN LIBRARY ICFLIB.   *                     10/03/90
     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
     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
```

Figure 10-23 (Part 6 of 14). Source Program Example — CSFMUL (System-Supplied Formats)

```
     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
 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.                                                       10/01/87
 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.                                                       10/08/87
 235 028400     WRITE DSPREC FORMAT IS "CIMENU".                                        10/01/87
 236 028500     GO TO READRQ.                                                           10/01/87
     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*    THE 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*******************************************************************          10/13/87
     029800* 7                                                                          10/14/87
     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
     030900/                                                                            10/01/87
```

Figure 10-23 (Part 7 of 14). Source Program Example — CSFMUL (System-Supplied Formats)

```
     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
     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
           *       ** CORRESPONDING items for statement 250:
           *       **     ITEMNO
           *       ** End of CORRESPONDING items for statement 250
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
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
```

Figure 10-23 (Part 8 of 14). Source Program Example — CSFMUL (System-Supplied Formats)

```
     036700 RETRY-ITEM.                                                                              10/01/87
264  036800     READ CMNFIL.                                                                         10/08/87
265  036900     IF MAJ-MIN = "0310"                                                                  10/01/87
266  037000        WRITE DSPREC FORMAT IS "TIMOUT"                                                   10/01/87
267  037100        READ DSPFIL INDICATORS ARE DSPF-INDIC-AREA                                        10/08/87
268  037200        IF TIMRSP = "1" GO TO RETRY-ITEM END-IF                                           01/21/88
270  037300        IF TIMRSP = "2" GO TO END-JOB END-IF.                                             01/21/88
272  037400     IF MAJ = "03"                                                                        10/01/87
273  037500        GO TO XITMIN.                                                                     10/01/87
274  037600     IF RCD-FMT-NME IS NOT EQUAL "ITMRSP" GO TO EXIT-FORMAT-ERR.                          10/02/87
276  037700     PERFORM ITMOUT-ROUTINE THRU ITMOUT-EXIT.                                             10/01/87
     037800 ITMIN-EXIT.                                                                              10/01/87
     037900     EXIT.                                                                                10/01/87
     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 IS 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
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
```

Figure 10-23 (Part 9 of 14). Source Program Example — CSFMUL (System-Supplied Formats)

```
     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
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
     046900     EXIT.                                                                               10/01/87
     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*    IS 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
           *       ** CORRESPONDING items for statement 326:
           *       **     CUSTNO
           *       ** End of CORRESPONDING items for statement 326
```

Figure 10-23 (Part 10 of 14). Source Program Example — CSFMUL (System-Supplied Formats)

```
 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
      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
 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
      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.                                     *                       10/03/90
      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
```

Figure 10-23 (Part 11 of 14). Source Program Example — CSFMUL (System-Supplied Formats)

```
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
     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
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********************************************************************                     02/21/89
     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
```

Figure 10-23 (Part 12 of 14). Source Program Example — CSFMUL (System-Supplied Formats)

```
     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
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
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   * * * * *
```

Figure 10-23 (Part 13 of 14). Source Program Example — CSFMUL (System-Supplied Formats)

```
*   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.
                        * * * * *   E N D   O F   M E S S A G E S   * * * * *
                                          Message Summary
  Total    Info(0-4)    Warning(5-19)    Error(20-29)    Severe(30-39)    Terminal(40-99)
      4          0            4                0                0                 0
 Source records read . . . . . . . . :   686
 Copy records read . . . . . . . . . :   99
 Copy members processed  . . . . . . :   1
 Sequence errors . . . . . . . . . . :   0
 Highest severity message issued . . :   10
  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 14 of 14). 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.

```
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.

```
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 Database 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 10-67 and Figure 10-27 on page 10-72. 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 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 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
```

```
Program . . . . . . . . . . . . . . . :   CTDMUL
  Library . . . . . . . . . . . . . . :     ICFLIB
Source file . . . . . . . . . . . . . :   QICFPUB
  Library . . . . . . . . . . . . . . :     ICFLIB
Source member . . . . . . . . . . . . :   CTDMUL     10/03/90 14:30:28
Generation severity level . . . . . . :   29
Text 'description' . . . . . . . . . . :   CBL Multiple Session Inquiry - Target DDS
Source listing options . . . . . . . . :   *SOURCE
Generation options . . . . . . . . . . :   *NONE
Message limit:
  Number of messages . . . . . . . . . :   *NOMAX
  Message limit severity . . . . . . . :   29
Print file . . . . . . . . . . . . . . :   QSYSPRT
  Library . . . . . . . . . . . . . . :     *LIBL
FIPS flagging . . . . . . . . . . . . :   *NOFIPS *NOSEG *NODEB *NOOBSOLETE
SAA flagging . . . . . . . . . . . . . :   *NOFLAG
Flagging severity . . . . . . . . . . :   0
Replace program . . . . . . . . . . . :   *YES
Target release . . . . . . . . . . . . :   *CURRENT
User profile . . . . . . . . . . . . . :   *USER
Authority . . . . . . . . . . . . . . :   *LIBCRTAUT
Compiler . . . . . . . . . . . . . . . :   IBM AS/400 COBOL/400

     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 DATABASE 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           *                  10/03/90
        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
```

Figure 10-26 (Part 1 of 5). Target Program Example — CTDMUL (User-Defined Formats)

```
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
   +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
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
  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
      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.                 *                     10/03/90
      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
  93  011400 END DECLARATIVES.                                                                       10/01/87
      011500********************************************************************                     10/01/87
      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
```

Figure 10-26 (Part 3 of 5). Target Program Example — CTDMUL (User-Defined Formats)

```
     012400********************************************************************                     10/01/87
     012500*                                                                  *                     10/01/87
     012600*  READ THE REQUEST FROM THE SOURCE PROGRAM.  INDICATOR 40         *                     10/01/87
     012700*  INDICATES RCVTRNRND OCCURRED.  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                *                     10/03/90
     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
     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 OCCURRED, 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.          *                     10/03/90
     019100* DETACH INDICATION IS CHECKED FOR AND IF IT OCCURRED, 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
```

Figure 10-26 (Part 4 of 5). Target Program Example — CTDMUL (User-Defined Formats)

```
   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   * * * * *

*   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.
                        * * * * *   E N D   O F   M E S S A G E S   * * * * *
                                          Message Summary
  Total    Info(0-4)    Warning(5-19)    Error(20-29)    Severe(30-39)    Terminal(40-99)
      2         0            2              0               0                0
 Source records read . . . . . . . . :   219
 Copy records read . . . . . . . . . :   44
 Copy members processed  . . . . . . :   2
 Sequence errors . . . . . . . . . . :   0
 Highest severity message issued . . :   10
  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)

```
Program  . . . . . . . . . . . . . . :   CTFMUL
  Library  . . . . . . . . . . . . . :     ICFLIB
Source file  . . . . . . . . . . . . :   QICFPUB
  Library  . . . . . . . . . . . . . :     ICFLIB
Source member  . . . . . . . . . . . :   CTFMUL     10/03/90 14:32:29
Generation severity level  . . . . . :   29
Text 'description' . . . . . . . . . :   CBL Multiple Session Inquiry - Target $$
Source listing options . . . . . . . :   *SOURCE
Generation options . . . . . . . . . :   *NONE
Message limit:
  Number of messages . . . . . . . . :   *NOMAX
  Message limit severity . . . . . . :   29
Print file . . . . . . . . . . . . . :   QSYSPRT
  Library  . . . . . . . . . . . . . :     *LIBL
FIPS flagging  . . . . . . . . . . . :   *NOFIPS *NOSEG *NODEB *NOOBSOLETE
SAA flagging . . . . . . . . . . . . :   *NOFLAG
Flagging severity  . . . . . . . . . :   0
Replace program  . . . . . . . . . . :   *YES
Target release . . . . . . . . . . . :   *CURRENT
User profile . . . . . . . . . . . . :   *USER
Authority  . . . . . . . . . . . . . :   *LIBCRTAUT
Compiler . . . . . . . . . . . . . . :   IBM AS/400 COBOL/400

    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 DATABASE 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          *                     10/03/90
       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
      +000002*    I-O FORMAT:DBREC      FROM FILE PFILE      OF LIBRARY ICFLIB             <-ALL-FMTS
      +000003*                                                                             <-ALL-FMTS
```

Figure 10-27 (Part 1 of 4). Target Program Example — CTFMUL (System-Supplied 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  004300 FD  CFILE                                                                             10/01/87
  33  004400     LABEL RECORDS ARE STANDARD.                                                       10/01/87
  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
      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
      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
```

Figure 10-27 (Part 2 of 4). Target Program Example — CTFMUL (System-Supplied Formats)

```
    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.          *         10/03/90
    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
    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 OCCURRED. 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         *         10/03/90
    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
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 DATABASE  *         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
```

Figure 10-27 (Part 3 of 4). Target Program Example — CTFMUL (System-Supplied Formats)

```
       017600*******************************************************************         10/01/87
       017700*                                                                 *         02/21/89
       017800* THIS ROUTINE ISSUES THE READ OPERATION TO THE PROGRAM DEVICE     *         02/21/89
       017900* UNTIL RCVTRNRND OCCURS.                                          *         02/21/89
       018000* DETACH INDICATION IS CHECKED FOR AND IF IT OCCURRED, THE         *         10/01/87
       018100* PROGRAM IS ENDED (RC=__08).                                      *         10/01/87
       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
       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   * * * * *

                     * * * * *   E N D   O F   M E S S A G E S   * * * * *
                                      Message Summary
  Total    Info(0-4)    Warning(5-19)    Error(20-29)    Severe(30-39)    Terminal(40-99)
     0         0            0                0               0                 0
 Source records read . . . . . . . . :   207
 Copy records read . . . . . . . . . :   11
 Copy members processed  . . . . . . :   1
 Sequence errors . . . . . . . . . . :   0
 Highest severity message issued . . :   0
  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 4 of 4). Target Program Example — CTFMUL (System-Supplied Formats)

Chapter 11. RPG/400 Communications Applications

Previous chapters in this book 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 book 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.

Figure 11-1 briefly introduces the RPG/400 statements you use in the communications portion of your program.

Figure 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 book 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 book for details on the syntax and function of each operation.

Error Handling: The INFDS option of the RPG/400 file specification allows you to obtain specific exception or error information for a file by defining and naming a data structure to contain error information, such as the ICF major and minor return codes.

You must understand the relationship between RPG *STATUS values and ICF major/minor return codes.

Figure 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.

*Figure 11-2 (Page 1 of 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

*Figure 11-2 (Page 2 of 2). *STATUS Values for Major and Minor Return Codes*

Major Code	Minor Code	*STATUS Value
04	All	01299
08	00	01285
11	00	00011
34	All	01201
80, 81	All	01251
82, 83	All	01255

Note: The mapping in Figure 11-2 on page 11-1 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.

Accessing the Feedback Areas: Use the RPG/400 POST operation to obtain the open or I/O feedback areas for an ICF file. For the RPG/400 support to access information from the I/O feedback area, add the following RPG/400 offset values to the offset values listed in Appendix C.

81	Open Feedback Area
241	Common I/O Feedback Area
367	File Dependent Feedback Area

Example Programs

The programs presented in this section are:

- Example I (Batch Data Transfer)

  Figure 11-3 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.

- Example II (Multiple-Session Inquiry)

  Figure 11-4 on page 11-3 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 11-4 on page 11-3 contains a display device and four ICF communications program devices.

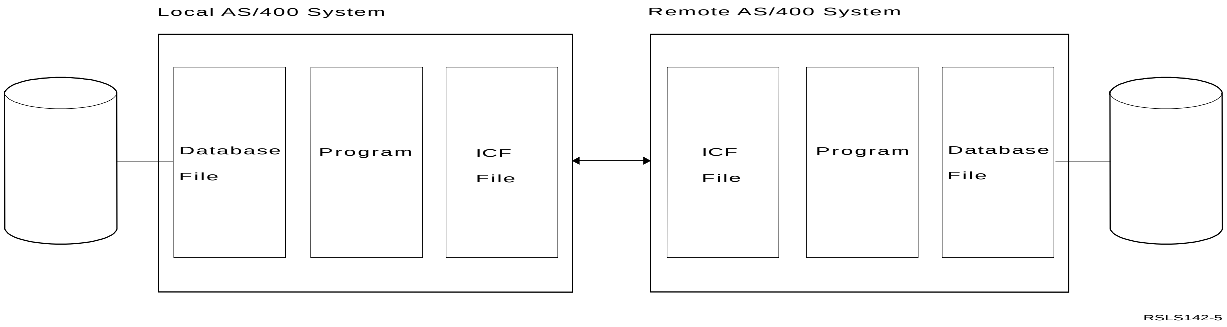


Figure 11-3. Batch Data Transfer

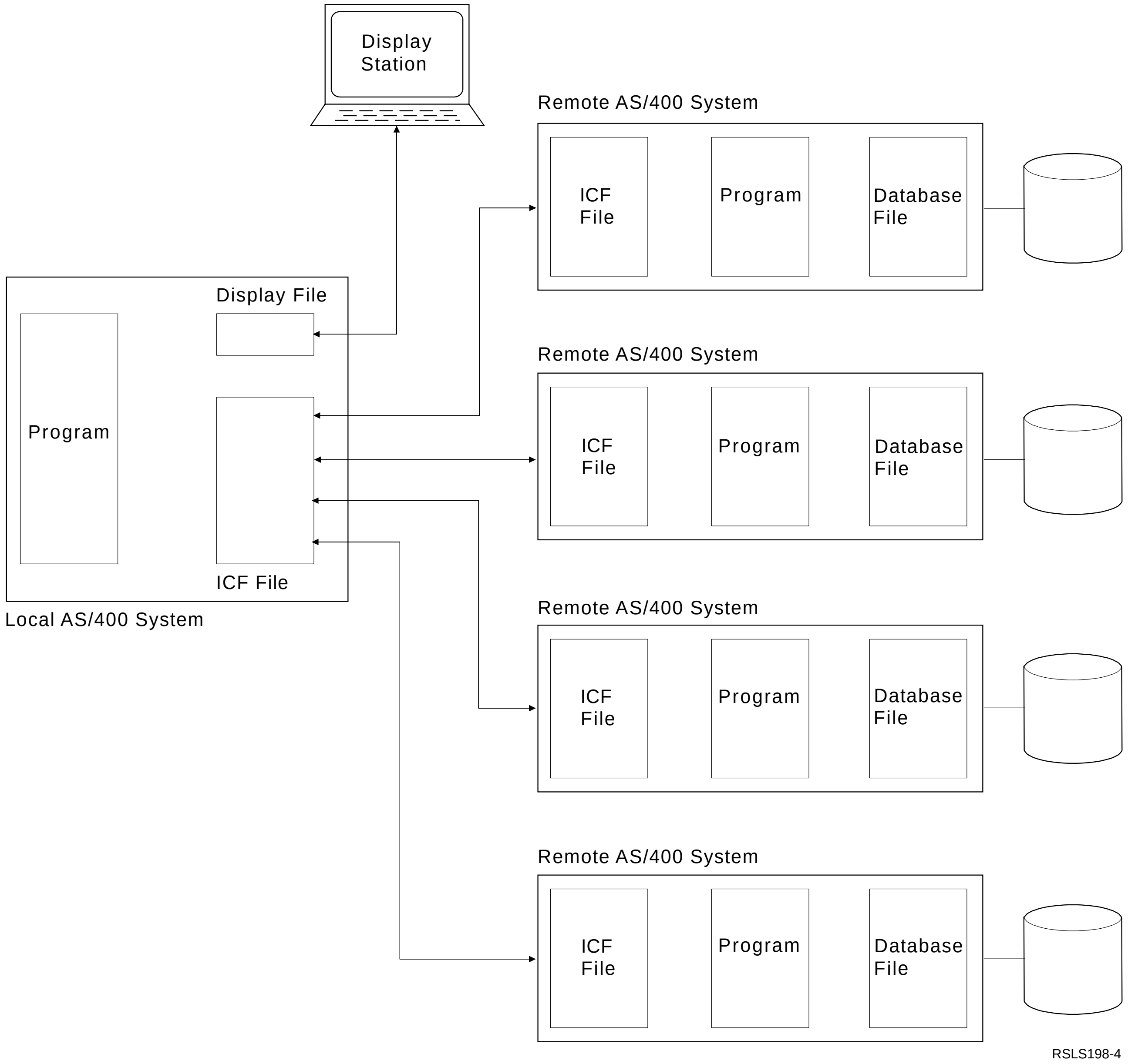


Figure 11-4. Multiple-Session Inquiry

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 11-5, 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.

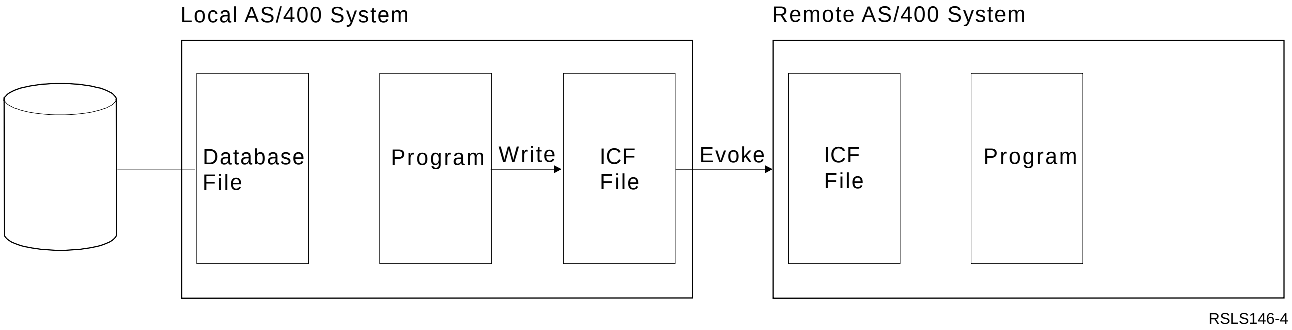


Figure 11-5. Evoke Request Starts a Target Program

After issuing the evoke request, the source program sends a database file to the target program, which prints the records as shown in Figure 11-6.

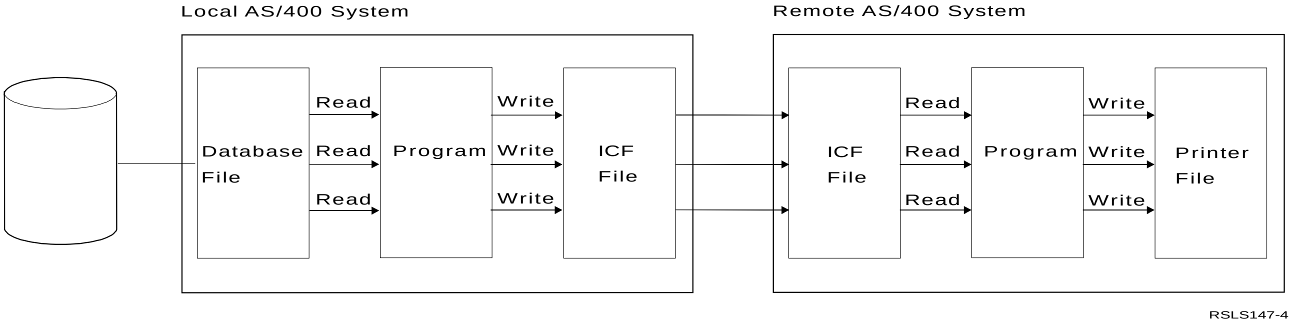


Figure 11-6. Target Program Prints Records

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 11-7.

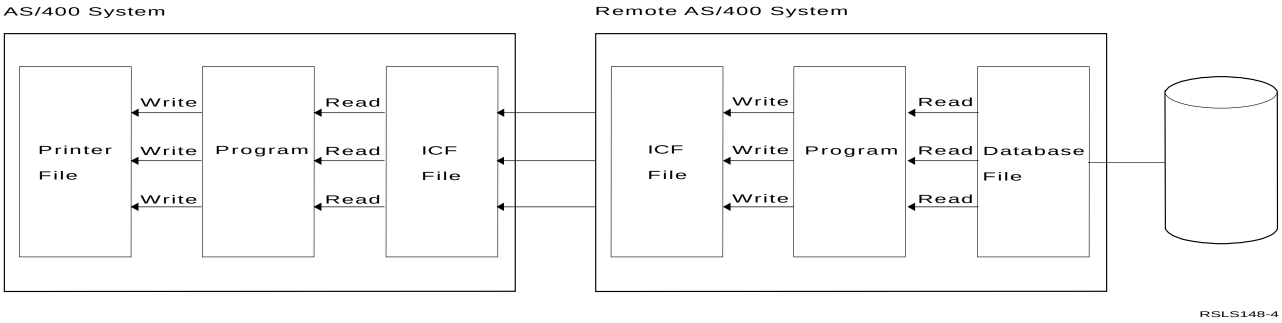


Figure 11-7. Source Program Prints the Received Records

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 11-8 on page 11-5.

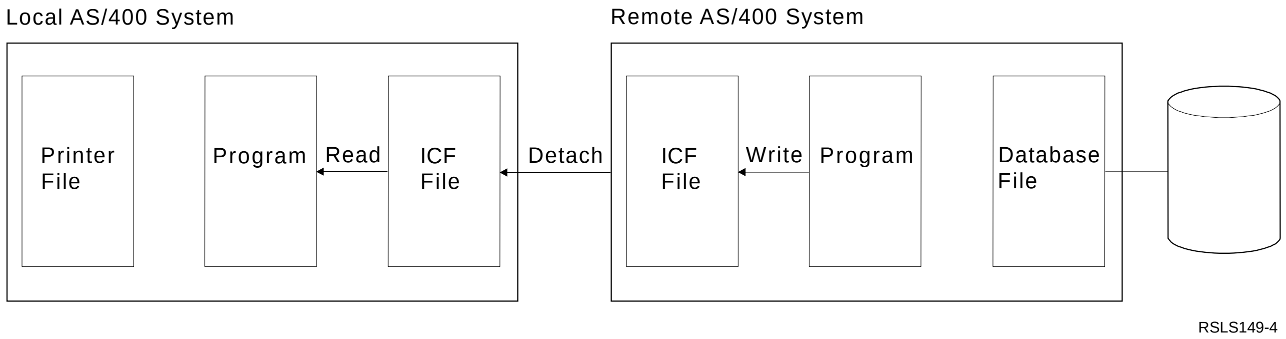


Figure 11-8. Target Program Ends the Transaction

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 below. 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
```

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 11-9 on page 11-7 and Figure 11-10 on page 11-12. 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 printer 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 printer 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.

```
 Compiler . . . . . . . . . . . . . . :   IBM AS/400 RPG/400
 Command Options:
   Program  . . . . . . . . . . . . . :   ICFLIB/RSDBAT
   Source file  . . . . . . . . . . . :   ICFLIB/QICFPUB
   Source member  . . . . . . . . . . :   *PGM
Text not available for message RXT0073 file QRPGMSG.
   Generation options . . . . . . . . :   *NOLIST     *NOXREF     *NOATR      *NODUMP     *NOOPTIMIZE
   Source listing indentation . . . . :   *NONE
   SAA flagging . . . . . . . . . . . :   *NOFLAG
   Generation severity level  . . . . :   9
   Print file . . . . . . . . . . . . :   *LIBL/QSYSPRT
   Replace program  . . . . . . . . . :   *YES
   Target release . . . . . . . . . . :   *CURRENT
   User profile . . . . . . . . . . . :   *USER
   Authority  . . . . . . . . . . . . :   *LIBCRTAUT
   Text . . . . . . . . . . . . . . . :   *SRCMBRTXT
   Phase trace  . . . . . . . . . . . :   *NO
   Intermediate text dump . . . . . . :   *NONE
   Snap dump  . . . . . . . . . . . . :   *NONE
   Codelist . . . . . . . . . . . . . :   *NONE
   Ignore decimal data error  . . . . :   *NO
 Actual Program Source:
   Member . . . . . . . . . . . . . . :   RSDBAT
   File . . . . . . . . . . . . . . . :   QICFPUB
   Library  . . . . . . . . . . . . . :   ICFLIB
   Last Change  . . . . . . . . . . . :   03/20/89  15:26:21
   Description  . . . . . . . . . . . :   rpg batch file transfer using dds source

                          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*  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
```

Figure 11-9 (Part 1 of 5). Source Program Example — RSDBAT (User-Defined Formats)

```
   1700  I* 2                                                                                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      DS                                                                     10/14/87
   1900  I                                       38  45 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
   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
   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 EXECUTION 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   * 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 11-9 (Part 2 of 5). Source Program Example — RSDBAT (User-Defined Formats)

```
     8300  C************************************************************                       10/14/87
     8400  C* 7                                                                                10/14/87
     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
    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
    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  * * * * *
```

Figure 11-9 (Part 3 of 5). Source Program Example — RSDBAT (User-Defined 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.

                         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

 Indicator References:
        INDICATOR  REFERENCES (M=MODIFIED D=DEFINED)
        *IN           3400M    3600M    7700    12600M
        LR           10400M
        OA            1400D   14501
* 7031  15
* 7031  30
        35            7700
        45           12600M
        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   * * * * *
```

Figure 11-9 (Part 4 of 5). Source Program Example — RSDBAT (User-Defined Formats)

```
                         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:    5
          Message . . . . :   The Name or indicator is not referenced.
* QRG7086 Severity:  00   Number:    1
          Message . . . . :   The RPG handles blocking function for file.
            INFDS contents updated only when blocks of data transferred.
* QRG7089 Severity:  00   Number:    1
          Message . . . . :   The RPG provides Separate-Indicator area for
            file.
      * * * * *   E N D   O F   M E S S A G E   S U M M A R Y   * * * * *

                           F i n a l   S u m m a r y
 Message Count:  (by Severity Number)
            TOTAL    00     10     20     30     40     50
              8       8      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 11-9 (Part 5 of 5). Source Program Example — RSDBAT (User-Defined Formats)

```
 Compiler . . . . . . . . . . . . . :   IBM AS/400 RPG/400
 Command Options:
   Program  . . . . . . . . . . . . :   ICFLIB/RSFBAT
   Source file  . . . . . . . . . . :   ICFLIB/QICFPUB
   Source member  . . . . . . . . . :   *PGM
Text not available for message RXT0073 file QRPGMSG.
   Generation options . . . . . . . :   *NOLIST     *NOXREF     *NOATR      *NODUMP     *NOOPTIMIZE
   Source listing indentation . . . :   *NONE
   SAA flagging . . . . . . . . . . :   *NOFLAG
   Generation severity level  . . . :   9
   Print file . . . . . . . . . . . :   *LIBL/QSYSPRT
   Replace program  . . . . . . . . :   *YES
   Target release . . . . . . . . . :   *CURRENT
   User profile . . . . . . . . . . :   *USER
   Authority  . . . . . . . . . . . :   *LIBCRTAUT
   Text . . . . . . . . . . . . . . :   *SRCMBRTXT
   Phase trace  . . . . . . . . . . :   *NO
   Intermediate text dump . . . . . :   *NONE
   Snap dump  . . . . . . . . . . . :   *NONE
   Codelist . . . . . . . . . . . . :   *NONE
   Ignore decimal data error  . . . :   *NO
 Actual Program Source:
   Member . . . . . . . . . . . . . :   RSFBAT
   File . . . . . . . . . . . . . . :   QICFPUB
   Library  . . . . . . . . . . . . :   ICFLIB
   Last Change  . . . . . . . . . . :   03/20/89  15:30:19
   Description  . . . . . . . . . . :   rpg batch file transfer using $$FORMAT

                          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                       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  45 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 11-10 (Part 1 of 4). Source Program Example — RSFBAT (System-Supplied Formats)

```
 3700  C******************************************************************                 10/16/87
 3800  C*                                                                                  03/20/89
 3900  C*   AFTER THE SUCCESSFUL EXECUTION 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
 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
 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
```

Figure 11-10 (Part 2 of 4). Source Program Example — RSFBAT (System-Supplied Formats)

```
    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 '$$EVOKNI'                                   10/16/87
    14600  O                                    8 'RTFBATCL'                                   10/16/87
    14700  O                                   16 'QSECOFR '                                   11/16/88
    14800  O                                   24 'QSECOFR '                                   11/16/88
    14900  O                                   32 'ICFLIB  '                                   10/16/87
    15000  O        E               SNDATA                                                     10/16/87
    15100  O                                   K8 '$$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   * * * * *
          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.

                         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
```

Figure 11-10 (Part 3 of 4). Source Program Example — RSFBAT (System-Supplied Formats)

```
 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
        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  * * * * *

                         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  * * * * *

                           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 11-10 (Part 4 of 4). 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 the following example. 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
```

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 11-18 and Figure 11-12 on page 11-22. 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 printer 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 printer 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 the 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.

```
 Compiler . . . . . . . . . . . . . :   IBM AS/400 RPG/400
 Command Options:
   Program  . . . . . . . . . . . . :   ICFLIB/RTDBAT
   Source file  . . . . . . . . . . :   ICFLIB/QICFPUB
   Source member  . . . . . . . . . :   *PGM
Text not available for message RXT0073 file QRPGMSG.
   Generation options . . . . . . . :   *NOLIST     *NOXREF     *NOATR      *NODUMP     *NOOPTIMIZE
   Source listing indentation . . . :   *NONE
   SAA flagging . . . . . . . . . . :   *NOFLAG
   Generation severity level  . . . :   9
   Print file . . . . . . . . . . . :   *LIBL/QSYSPRT
   Replace program  . . . . . . . . :   *YES
   Target release . . . . . . . . . :   *CURRENT
   User profile . . . . . . . . . . :   *USER
   Authority  . . . . . . . . . . . :   *LIBCRTAUT
   Text . . . . . . . . . . . . . . :   *SRCMBRTXT
   Phase trace  . . . . . . . . . . :   *NO
   Intermediate text dump . . . . . :   *NONE
   Snap dump  . . . . . . . . . . . :   *NONE
   Codelist . . . . . . . . . . . . :   *NONE
   Ignore decimal data error  . . . :   *NO
 Actual Program Source:
   Member . . . . . . . . . . . . . :   RTDBAT
   File . . . . . . . . . . . . . . :   QICFPUB
   Library  . . . . . . . . . . . . :   ICFLIB
   Last Change  . . . . . . . . . . :   03/20/89  15:40:57
   Description  . . . . . . . . . . :   rpg batch file transfer using dds source

                          Source Listing
      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   * 1                                                                                10/16/87
           H                                                                                              *****
     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* 2                                                                                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      DS                                                                     10/16/87
     1900  I                                       38  45 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
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
8400  C* 7                                                                                10/16/87
8500  C           END       TAG                                                           10/16/87
8600  C                     SETON                     LR                    1             10/16/87
8700  C                     RETRN                                                         10/16/87
```

Figure 11-11 (Part 2 of 4). Target Program Example — RTDBAT (User-Defined Formats)

```
     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 TERMINATION                     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   * * * * *

          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.
                         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
```

Figure 11-11 (Part 3 of 4). Target Program Example — RTDBAT (User-Defined Formats)

```
 Field References:
        FIELD       ATTR    REFERENCES (M=MODIFIED D=DEFINED)
        *IN15       A(1)       4200
        *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)
        *IN           4200    10900M
        LR            8600M
        OA            1400D   12901
        15            4200
        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   * * * * *

                         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:    5
          Message . . . . :   The Name or indicator is not referenced.
* QRG7086 Severity:  00   Number:    1
          Message . . . . :   The RPG handles blocking function for file.
            INFDS contents updated only when blocks of data transferred.
* QRG7089 Severity:  00   Number:    1
          Message . . . . :   The RPG provides Separate-Indicator area for
            file.
      * * * * *  E N D   O F   M E S S A G E   S U M M A R Y   * * * * *

                           F i n a l   S u m m a r y
 Message Count:  (by Severity Number)
            TOTAL    00     10     20     30     40     50
              8       8      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)

```
 Compiler . . . . . . . . . . . . . :   IBM AS/400 RPG/400
 Command Options:
   Program  . . . . . . . . . . . . :   ICFLIB/RTFBAT
   Source file  . . . . . . . . . . :   ICFLIB/QICFPUB
   Source member  . . . . . . . . . :   *PGM
Text not available for message RXT0073 file QRPGMSG.
   Generation options . . . . . . . :   *NOLIST     *NOXREF     *NOATR      *NODUMP     *NOOPTIMIZE
   Source listing indentation . . . :   *NONE
   SAA flagging . . . . . . . . . . :   *NOFLAG
   Generation severity level  . . . :   9
   Print file . . . . . . . . . . . :   *LIBL/QSYSPRT
   Replace program  . . . . . . . . :   *YES
   Target release . . . . . . . . . :   *CURRENT
   User profile . . . . . . . . . . :   *USER
   Authority  . . . . . . . . . . . :   *LIBCRTAUT
   Text . . . . . . . . . . . . . . :   *SRCMBRTXT
   Phase trace  . . . . . . . . . . :   *NO
   Intermediate text dump . . . . . :   *NONE
   Snap dump  . . . . . . . . . . . :   *NONE
   Codelist . . . . . . . . . . . . :   *NONE
   Ignore decimal data error  . . . :   *NO
 Actual Program Source:
   Member . . . . . . . . . . . . . :   RTFBAT
   File . . . . . . . . . . . . . . :   QICFPUB
   Library  . . . . . . . . . . . . :   ICFLIB
   Last Change  . . . . . . . . . . :   03/20/89  15:15:51
   Description  . . . . . . . . . . :   rpg batch file transfer using $$FORMAT

 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  45 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                     EXCPTPTPREC                                                   10/16/87
4400  C           CHKTRN    TAG                                                           10/16/87
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 TERMINATION 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
```

Figure 11-12 (Part 2 of 4). Target Program Example — RTFBAT (System-Supplied Formats)

```
     9700  C* 8                                                                                10/16/87
     9800  C           WCFRTN    BEGSR                                                         10/16/87
     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.

                         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
        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  * * * * *

                         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:    2
          Message . . . . :   The Name or indicator is not referenced.
* QRG7086 Severity:  00   Number:    1
          Message . . . . :   The RPG handles blocking function for file.
            INFDS contents updated only when blocks of data transferred.
* QRG7089 Severity:  00   Number:    1
          Message . . . . :   The RPG provides Separate-Indicator area for
            file.
      * * * * *  E N D  O F  M E S S A G E  S U M M A R Y  * * * * *

                           F i n a l   S u m m a r y
 Message Count:  (by Severity Number)
            TOTAL    00     10     20     30     40     50
              5       5      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 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.

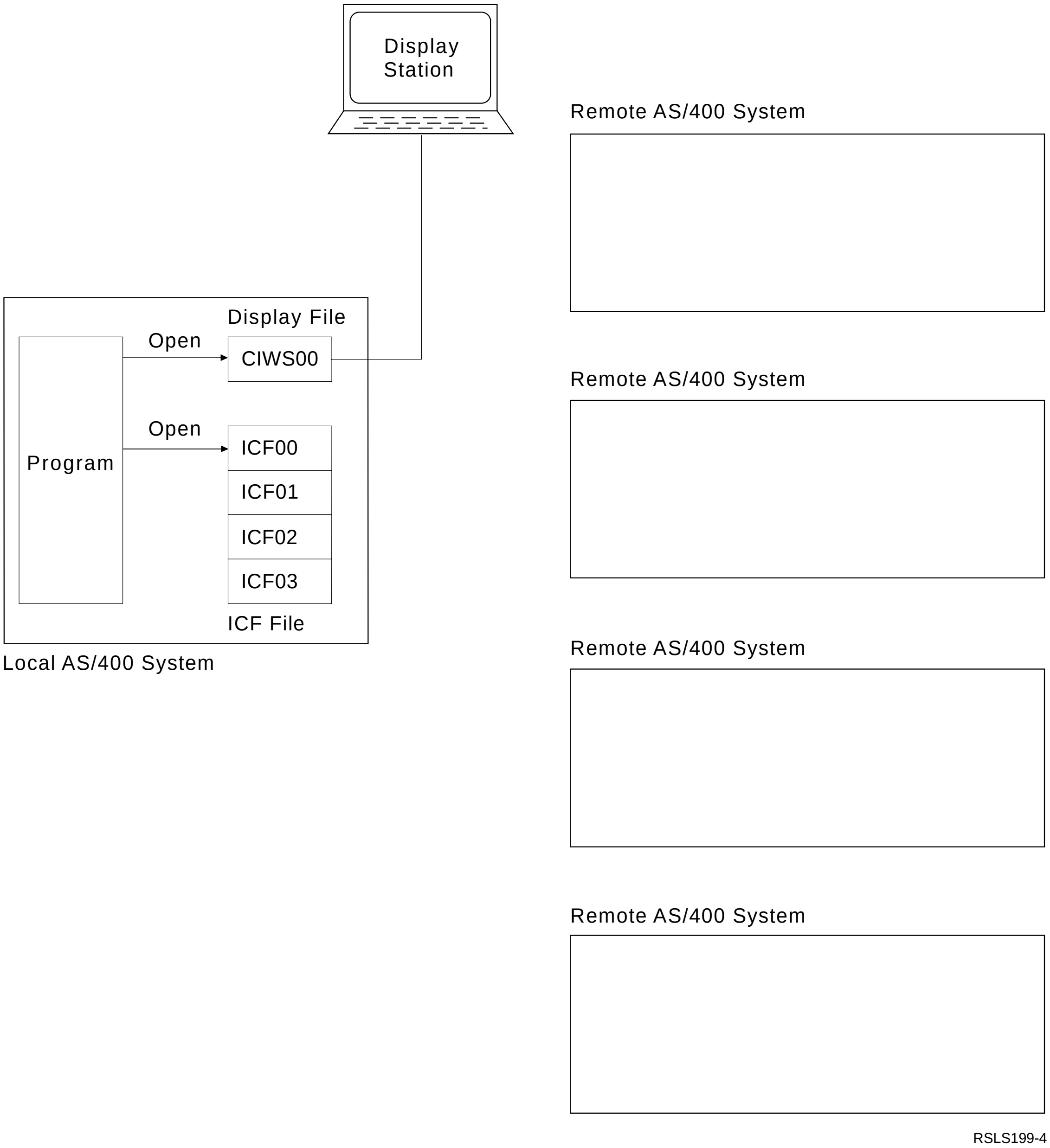


Figure 11-13. Program Starts at Display Station

All other program devices must be explicitly acquired by the program, as shown in Figure 11-14.

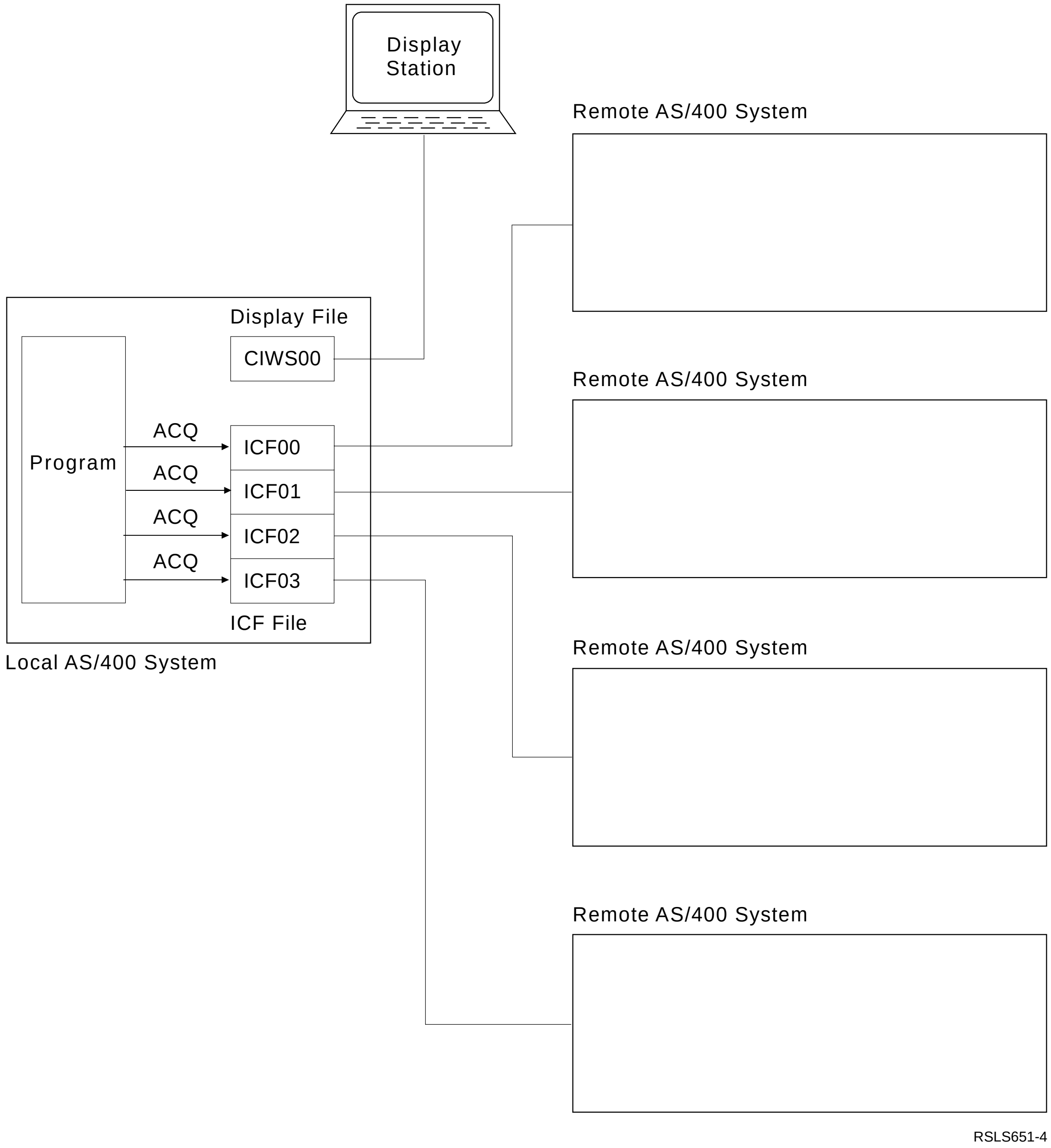


Figure 11-14. Program Devices Explicitly Acquired

All target programs are started with an evoke, as shown in Figure 11-15.

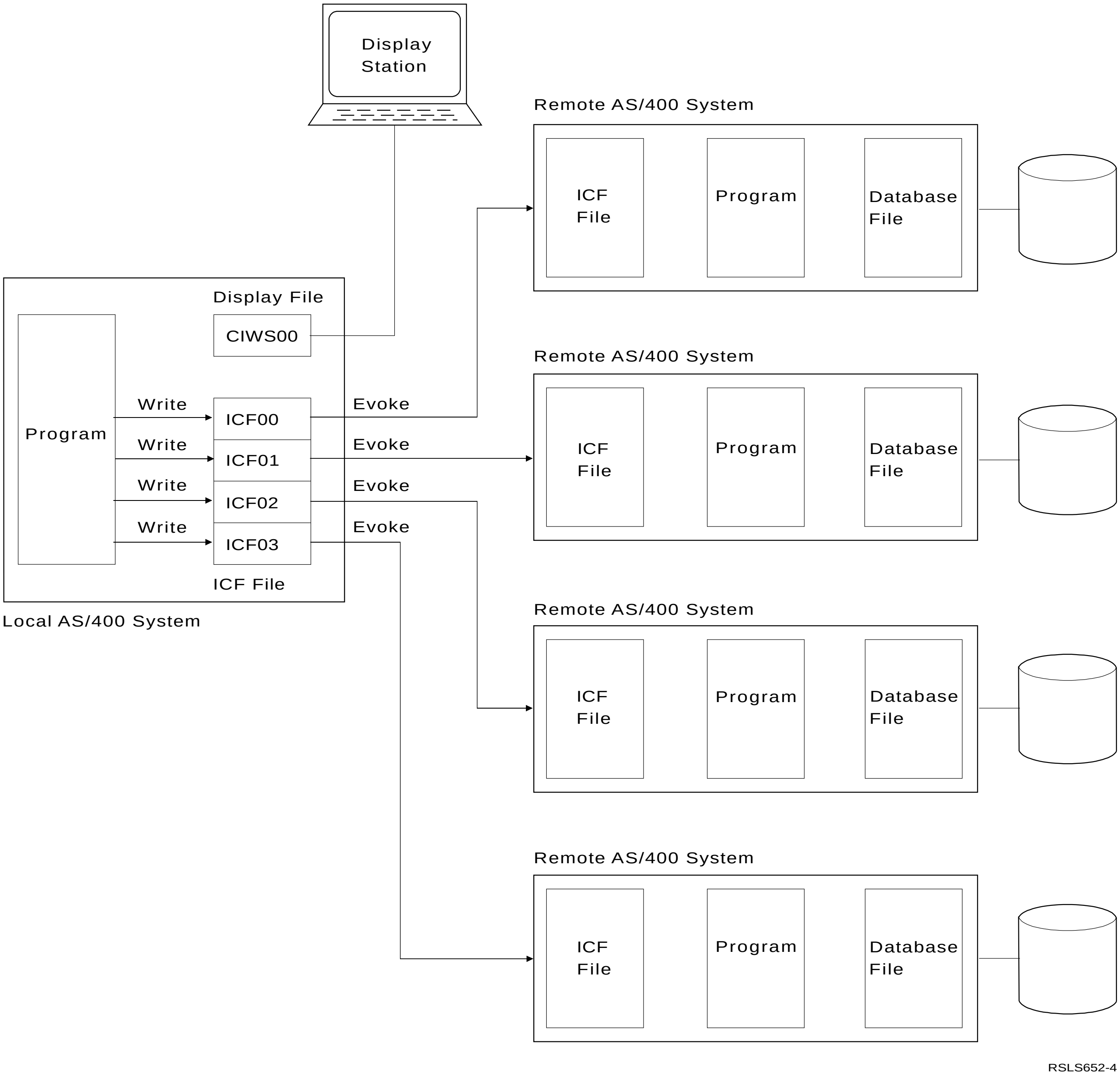


Figure 11-15. Evoke Starts Target Programs

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.

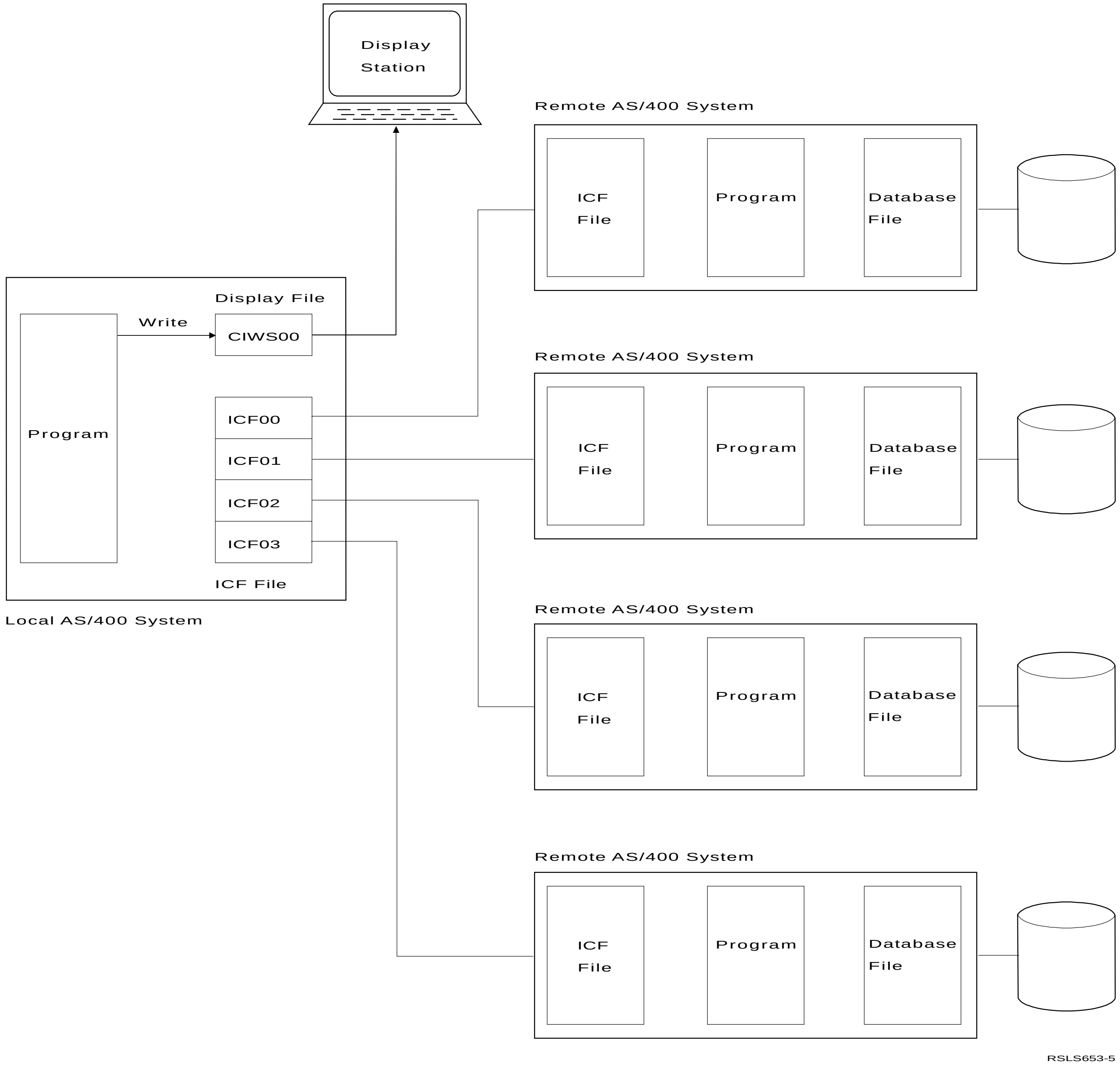


Figure 11-16. Main Menu Written to Display Station

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.

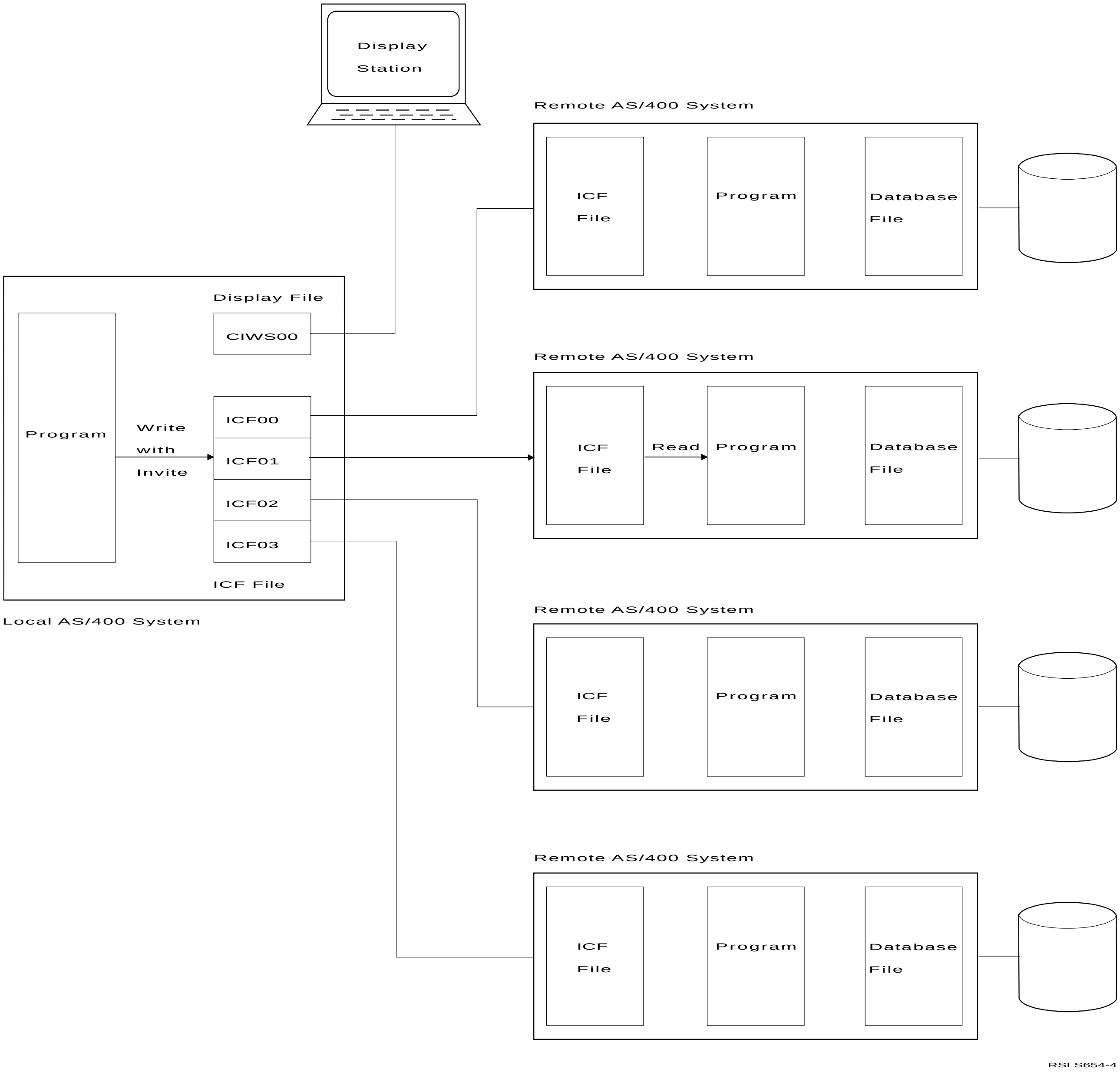


Figure 11-17. Program Sends Inquiry Request to Remote System

The target program responds to the inquiry by sending a reply, as shown in Figure 11-18.

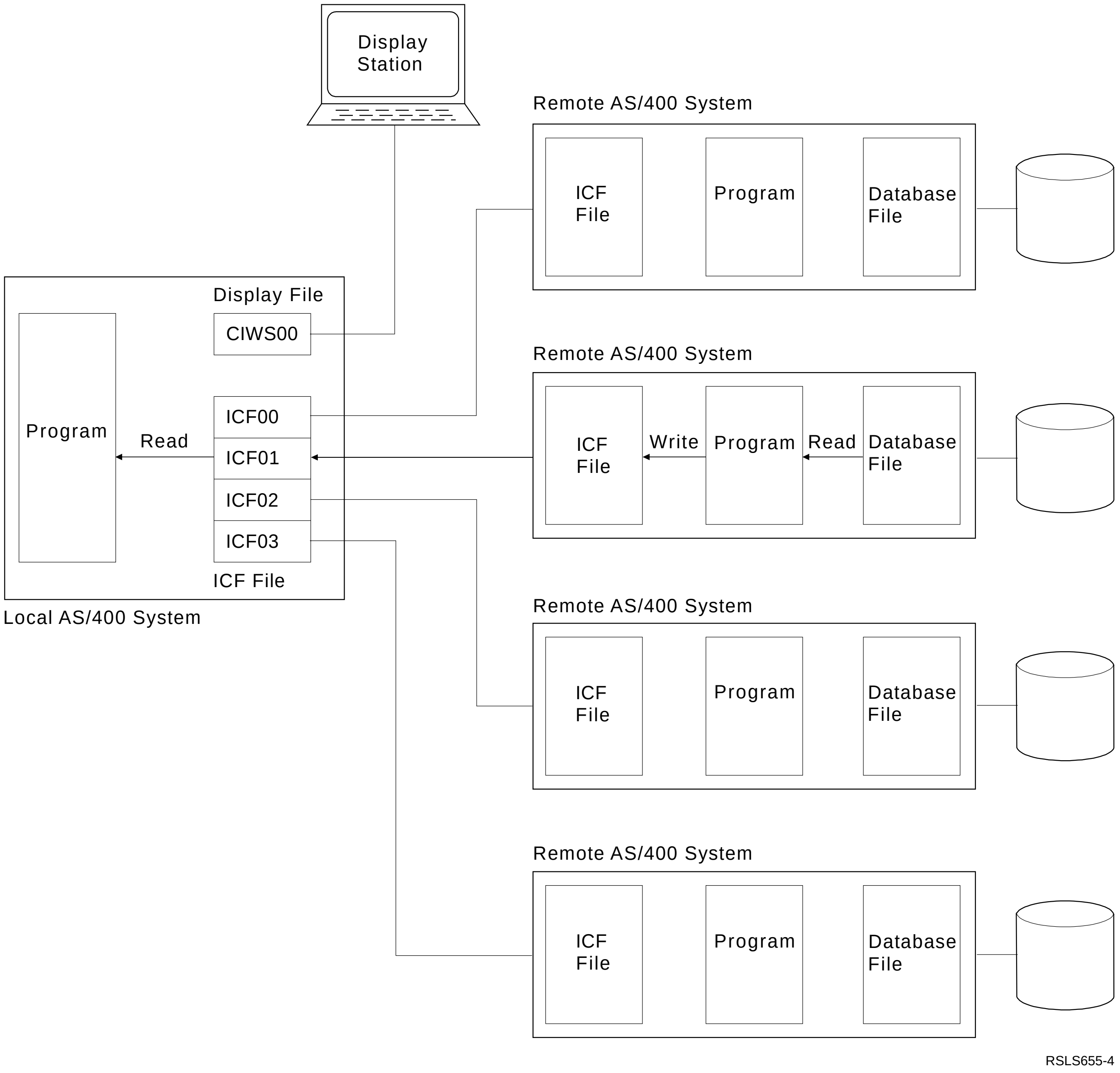


Figure 11-18. Target Program Sends a Reply

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.

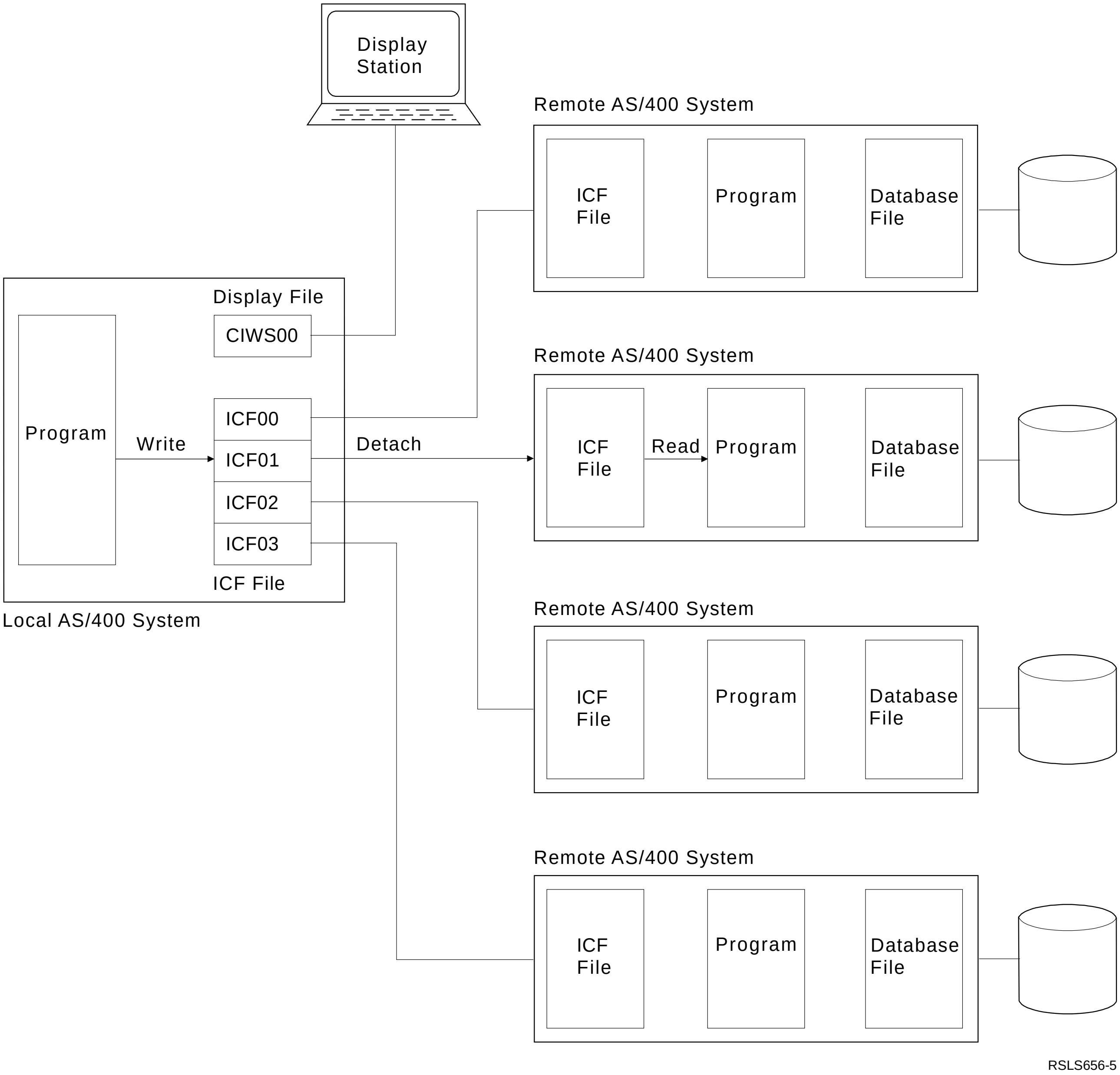


Figure 11-19. Program Ends the Session

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 below.

```
 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
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  * * * *
```

The DDS source file for the display file (DSPFIL) is shown below.

```
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
003100871008     A                                       THIS    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')
005600871007     A                                      OVERLAY
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
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  ')
008300871007     A                                      OVERLAY
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')
010200871007     A                                      OVERLAY
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
```

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 11-20 on page 11-38 and in Figure 11-21 on page 11-52. 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 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 not valid, 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 no valid 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.

```
 Compiler . . . . . . . . . . . . . :   IBM AS/400 RPG/400
 Command Options:
   Program  . . . . . . . . . . . . :   ICFLIB/RSDMUL
   Source file  . . . . . . . . . . :   ICFLIB/QICFPUB
   Source member  . . . . . . . . . :   *PGM
Text not available for message RXT0073 file QRPGMSG.
   Generation options . . . . . . . :   *NOLIST     *NOXREF     *NOATR      *NODUMP     *NOOPTIMIZE
   Source listing indentation . . . :   *NONE
   SAA flagging . . . . . . . . . . :   *NOFLAG
   Generation severity level  . . . :   9
   Print file . . . . . . . . . . . :   *LIBL/QSYSPRT
   Replace program  . . . . . . . . :   *YES
   Target release . . . . . . . . . :   *CURRENT
   User profile . . . . . . . . . . :   *USER
   Authority  . . . . . . . . . . . :   *LIBCRTAUT
   Text . . . . . . . . . . . . . . :   *SRCMBRTXT
   Phase trace  . . . . . . . . . . :   *NO
   Intermediate text dump . . . . . :   *NONE
   Snap dump  . . . . . . . . . . . :   *NONE
   Codelist . . . . . . . . . . . . :   *NONE
   Ignore decimal data error  . . . :   *NO
 Actual Program Source:
   Member . . . . . . . . . . . . . :   RSDMUL
   File . . . . . . . . . . . . . . :   QICFPUB
   Library  . . . . . . . . . . . . :   ICFLIB
   Last Change  . . . . . . . . . . :   10/03/90  14:46:07
   Description  . . . . . . . . . . :   RPG Multi-Session example w/DDS (source)

                         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
```

Figure 11-20 (Part 1 of 14). Source Program Example — RSDMUL (User-Defined Formats)

```
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                             10/03/90
2200  F*               OF FOUR DIFFERENT TARGET PROGRAMS.  MULTIPLE                       10/13/87
2300  F*               SESSIONS ARE ACTIVE CONCURRENTLY.                                  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.                               10/03/90
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
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   * 1                                                                                10/13/87
      H                                                                                              *****
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
       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                           10/03/90
6000  I*            FILE.                                                                 10/13/87
6100  I*                                                                                  10/13/87
6200  I****************************************************************                   10/13/87
```

Figure 11-20 (Part 2 of 14). Source Program Example — RSDMUL (User-Defined Formats)

```
   6300  I* 2                                                                              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
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
```

Figure 11-20 (Part 3 of 14). Source Program Example — RSDMUL (User-Defined Formats)

```
M000003                                           1   1 *IN99
N000000   INPUT  FIELDS FOR RECORD TIMOUT FILE DSPFIL FORMAT TIMOUT.
N000000         TIME OUT SCREEN
N000001                                           3   3 *IN97
N000002                                           2   2 *IN98
N000003                                           1   1 *IN99
N000004                                           4   4 TIMRSP
   6400  IIODS        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
   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  45 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.                        10/03/90
   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   * 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
  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
```

Figure 11-20 (Part 4 of 14). Source Program Example — RSDMUL (User-Defined Formats)

```
 * 4                                                                                10/13/87
C           READRQ    TAG                                                           10/13/87
C                     SETOF                     8889  TIMEOUT IND     1 2           10/13/87
C                     READ DSPFIL                   87                    3         10/13/87
C           RECID     CABEQ'CIMENU  'MENU             MAIN MENU?                    03/20/89
C           RECID     CABEQ'ITMMNU  'ITMIN            ITEM MENU?                    03/20/89
C           RECID     CABEQ'ITMSC2  'ITMRTN           ITM SCR?                      10/13/87
C           RECID     CABEQ'ITMSC3  'ITMRTN           ITM SCR?                      10/13/87
C           RECID     CABEQ'DTLMNU  'DTLIN            DETAIL SCR?                   10/13/87
C           RECID     CABEQ'DTLSCR  'DTLRTN           CUST SCR?                     10/13/87
C                     WRITECIMENU                     MAIN MENU IF                  10/13/87
C                     GOTO READRQ                     THERE IS ERR                  10/13/87
C****************************************************************                    10/13/87
C*                                                                                  10/13/87
C*                             MAIN MENU                                            10/13/87
C*                                                                                  10/13/87
C*    THE MAIN MENU IS READ TO DETERMINE THE REQUEST ENTERED                        10/13/87
C*    BY THE USER. IF CMD 1 (*IN99) IS PRESSED, THE PROGRAM                         10/13/87
C*    IS ENDED.  IF OPTION = 1, AN ITEM INQUIRY MENU IS WRITTEN                     10/13/87
C*    TO THE SCREEN.  IF OPTION = 2, A CUSTOMER INQUIRY MENU IS                     10/13/87
C*    WRITTEN TO THE SCREEN.                                                        10/13/87
C*                                                                                  10/13/87
C****************************************************************                    10/13/87
 * 5                                                                                10/13/87
C           MENU      TAG                                                           10/13/87
C           *IN99     CABEQ'1'       END              JOB ENDS                      10/13/87
C           OPTION    IFEQ '1'                                               B001   10/13/87
C                     WRITEITMMNU                     ITEM MENU               001   10/13/87
C                     ELSE                                                   X001   10/13/87
C                     WRITEDTLMNU                     CUST MENU               001   10/13/87
C                     END                                                    E001   10/13/87
C                     GOTO READRQ                                                   10/13/87
C****************************************************************                    10/13/87
C*                                                                                  10/13/87
C*                        ITEM INQUIRY                                              10/13/87
C*                                                                                  10/13/87
C*    THE ITEM NUMBER REQUESTED BY THE USER ON THE ITEM INQUIRY                     10/13/87
C*    SCREEN IS CHECKED.  THIS IS DETERMINED BY THE DISPLAY                         03/20/89
C*    RECORD FORMAT BEING PROCESSED - IN THIS CASE ITMMNU.                          03/20/89
C*                                                                                  10/13/87
C*    IF CMD 1 (*IN99) IS PRESSED, THE PROGRAM IS ENDED.  IF CMD 2                  10/13/87
C*    IS PRESSED, THE ITEM INQUIRY REQUEST IS CANCELED, AND THE                     03/20/89
C*    MAIN MENU (CIMENU) IS WRITTEN TO THE SCREEN.                                  10/13/87
C*                                                                                  10/13/87
C*    IF AN ITEM NUMBER IS ENTERED, A ITEM INQUIRY REQUEST IS                       10/13/87
C*    SENT TO THE APPROPRIATE REMOTE SYSTEM.  THE REMOTE SYSTEM                     10/13/87
C*    IS SELECTED BASED ON THE ITEM NUMBER REQUESTED.                               10/13/87
C*                                                                                  10/13/87
C*    IF AN ERROR OCCURS, THE ERROR IS PRINTED AND THE JOB                          10/13/87
C*    IS ENDED.                                                                     10/13/87
C*                                                                                  10/13/87
C****************************************************************                    10/13/87
 * 6                                                                                10/13/87
C           ITMIN     TAG                                                           10/13/87
C           *IN99     CABEQ'1'       END              EXIT ON CMD3                  10/13/87
C           *IN98     IFEQ '1'                                               B001   10/13/87
C                     WRITECIMENU                     MAIN MENU               001   10/13/87
C                     GOTO READRQ                                             001   10/13/87
C                     END                                                    E001   10/13/87
C           ITEMNO    CABLE399999    XICF01                                         10/13/87
C           ITEMNO    CABLE699999    XICF02                                         10/13/87
C           ITEMNO    CABLE899999    XICF03                                         10/13/87
C           XICF01    TAG                                                           10/13/87
C                     MOVEL'ICF01   'CMID                                           10/13/87
C                     GOTO XITMIN                                                   10/13/87
C           XICF02    TAG                                                           10/13/87
C                     MOVEL'ICF02   'CMID                                           10/13/87
C                     GOTO XITMIN                                                   10/13/87
C           XICF03    TAG                                                           10/13/87
C                     MOVEL'ICF03   'CMID                                           10/13/87
C           XITMIN    TAG                                                           10/13/87
C                     WRITEITMREQ                     INQ W/INVITE                  10/13/87
```

Figure 11-20 (Part 5 of 14). Source Program Example — RSDMUL (User-Defined Formats)

```
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            0Q
19700  C                     MOVEL'        'CMID                                           10/13/87
19800  C                     READ CMNFIL                 8910RECV ITM INFO     2 3         10/13/87
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   * 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
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
```

Figure 11-20 (Part 6 of 14). Source Program Example — RSDMUL (User-Defined Formats)

```
26000   * 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
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
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   * 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
32700  C           MAJCOD    CABGE'04'      ERROR           ERROR RTN               001   10/13/87
32800  C           RECID2    CABNE'DTLRSP'  RECERR          PRINT MSG               001   10/13/87
```

Figure 11-20 (Part 7 of 14). Source Program Example — RSDMUL (User-Defined Formats)

```
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   * 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   08/08/89
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   * 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
38200  C****************************************************************                  10/13/87
38300   * 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
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
```

Figure 11-20 (Part 8 of 14). Source Program Example — RSDMUL (User-Defined Formats)

```
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   * 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 OUT 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   * 14                                                                               10/13/87
43500  C           ERRCHK    BEGSR                                                         10/13/87
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
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                          10/03/90
45700  C*    IS ALREADY ACQUIRED, THE ERROR IS IGNORED, OTHERWISE, THE                     03/20/89
45800  C*    PROGRAM IS TERMINATED.                                                        03/20/89
45900  C*                                                                                  10/14/87
46000  C****************************************************************                   10/14/87
46100   * 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
```

Figure 11-20 (Part 9 of 14). Source Program Example — RSDMUL (User-Defined Formats)

```
    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
* 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                            CNAME     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
  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
* 7089      4400   RPG PROVIDES SEPARATE INDICATOR AREA FOR FILE CMNFIL.
```

Figure 11-20 (Part 10 of 14). Source Program Example — RSDMUL (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  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
        CNAME       A(5)      35000M V000009D
        CSTM        P(9,2)    28400M Y000003D
        CSTTM       P(9,2)  A000014D   27700    28400
        CSTTY       P(11,2) A000015D   28600
        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
```

Figure 11-20 (Part 11 of 14). Source Program Example — RSDMUL (User-Defined Formats)

```
* 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
        MENU        TAG       12600    14600D
        MINCOD      A(2)       7600D   47300    48300
        MMERR       EXCPT     38800    46800
        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
        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
```

Figure 11-20 (Part 12 of 14). Source Program Example — RSDMUL (User-Defined Formats)

```
        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
        '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
        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
```

Figure 11-20 (Part 13 of 14). Source Program Example — RSDMUL (User-Defined Formats)

```
* 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   * * * * *

                         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   * * * * *

                           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 11-20 (Part 14 of 14). Source Program Example — RSDMUL (User-Defined Formats)

```
 Compiler . . . . . . . . . . . . . :   IBM AS/400 RPG/400
 Command Options:
   Program  . . . . . . . . . . . . :   ICFLIB/RSFMUL
   Source file  . . . . . . . . . . :   ICFLIB/QICFPUB
   Source member  . . . . . . . . . :   *PGM
Text not available for message RXT0073 file QRPGMSG.
   Generation options . . . . . . . :   *NOLIST     *NOXREF     *NOATR      *NODUMP     *NOOPTIMIZE
   Source listing indentation . . . :   *NONE
   SAA flagging . . . . . . . . . . :   *NOFLAG
   Generation severity level  . . . :   9
   Print file . . . . . . . . . . . :   *LIBL/QSYSPRT
   Replace program  . . . . . . . . :   *YES
   Target release . . . . . . . . . :   *CURRENT
   User profile . . . . . . . . . . :   *USER
   Authority  . . . . . . . . . . . :   *LIBCRTAUT
   Text . . . . . . . . . . . . . . :   *SRCMBRTXT
   Phase trace  . . . . . . . . . . :   *NO
   Intermediate text dump . . . . . :   *NONE
   Snap dump  . . . . . . . . . . . :   *NONE
   Codelist . . . . . . . . . . . . :   *NONE
   Ignore decimal data error  . . . :   *NO
 Actual Program Source:
   Member . . . . . . . . . . . . . :   RSFMUL
   File . . . . . . . . . . . . . . :   QICFPUB
   Library  . . . . . . . . . . . . :   ICFLIB
   Last Change  . . . . . . . . . . :   10/03/90  14:47:52
   Description  . . . . . . . . . . :   RPG Multi-Session example w/$$FORMAT (source)

                          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                             10/03/90
     2100  F*               OF FOUR DIFFERENT TARGET PROGRAMS.  MULTIPLE                       10/13/87
     2200  F*               SESSIONS ARE ACTIVE CONCURRENTLY.                                  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
```

Figure 11-21 (Part 1 of 14). Source Program Example — RSFMUL (System-Supplied Formats)

```
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                           10/03/90
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
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   *1                                                                                 10/13/87
      H                                                                                              *****
4500  FCMNFIL  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
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                           10/03/90
6100  I*            FILE.                                                                 10/13/87
6200  I*                                                                                  03/20/89
6300  I****************************************************************                   10/13/87
6400   *2                                                                                 03/20/89
6500  ICMNFIL  NS  80   1 CC                                                              03/20/89
6600  I                                        1   1 RECCUS                               10/13/87
6700  I                                        2   70CUSTNO                               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
```

Figure 11-21 (Part 2 of 14). Source Program Example — RSFMUL (System-Supplied Formats)

```
   7700  I        NS  81   1 CI                                                              10/13/87
   7800  I                                        1   1 RECITM                               10/13/87
   7900  I                                        2   70ITEMNO                               10/13/87
   8000  I                                        8  37 DESC                                 10/13/87
   8100  I                                       38  440QTYLST                               10/13/87
   8200  I                                       45  510QTYOH                                10/13/87
   8300  I                                       52  580QTYOO                                10/13/87
   8400  I                                       59  650QTYBO                                10/13/87
   8500  I                                       66  67 UNIT                                 10/13/87
   8600  I                                       68  742PR01                                 10/13/87
   8700  I                                       75  800PR05                                 10/13/87
   8800  I                                       81  852UFRT                                 10/13/87
   8900  I                                       86  942SLSTM                                10/13/87
   9000  I                                       95 1052SLSTY                                10/13/87
   9100  I                                      106 1142CSTTM                                10/13/87
   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        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
```

Figure 11-21 (Part 3 of 14). Source Program Example — RSFMUL (System-Supplied Formats)

```
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  45 FMTNM                                10/13/87
10600  I                                      273 282 CMID                                 10/13/87
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
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   * 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   * 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
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
```

Figure 11-21 (Part 4 of 14). Source Program Example — RSFMUL (System-Supplied Formats)

```
C****************************************************************                   10/13/87
C*                                                                                  10/13/87
C*                             MAIN MENU                                            10/13/87
C*                                                                                  10/13/87
C*    THE MAIN MENU IS READ TO DETERMINE THE REQUEST ENTERED BY                     03/20/89
C*    THE USER. IF CMD 1 (*IN99) IS PRESSED, THE PROGRAM IS ENDED.                  03/20/89
C*    IF OPTION = 1, AN ITEM INQUIRY MENU IS WRITTEN TO THE SCREEN.                 03/20/89
C*    IF OPTION = 2, A CUSTOMER INQUIRY MENU IS WRITTEN TO THE                      03/20/89
C*    SCREEN.                                                                       03/20/89
C*                                                                                  10/13/87
C****************************************************************                   10/13/87
 * 5                                                                                10/13/87
C           MENU      TAG                                                           10/13/87
C           *IN99     CABEQ'1'       END             END PROGRAM                    10/13/87
C           OPTION    IFEQ '1'                                                B001    10/13/87
C                     WRITEITMMNU                     ITEM MENU                001    10/13/87
C                     ELSE                                                    X001    10/13/87
C                     WRITEDTLMNU                     CUST MENU                001    10/13/87
C                     END                                                     E001    10/13/87
C                     GOTO READRQ                                                   10/13/87
C***************************************************************                    10/13/87
C*                                                                                  10/13/87
C*                        ITEM INQUIRY                                              10/13/87
C*                                                                                  10/13/87
C*    THE ITEM NUMBER REQUESTED BY THE USER ON THE ITEM INQUIRY                     10/13/87
C*    SCREEN IS CHECKED.  THIS IS DETERMINED BY THE DISPLAY                         03/20/89
C*    RECORD FORMAT BEING PROCESSED - IN THIS CASE ITMMNU.                          03/20/89
C*                                                                                  10/13/87
C*    IF CMD 1 (*IN99) IS PRESSED, THE PROGRAM IS ENDED.  IF CMD 2                  10/13/87
C*    IS PRESSED, THE ITEM INQUIRY REQUEST IS CANCELED, AND THE                     10/13/87
C*    MAIN MENU (CIMENU) IS WRITTEN TO THE SCREEN.                                  10/13/87
C*                                                                                  10/13/87
C*    IF AN ITEM NUMBER IS ENTERED, A ITEM INQUIRY REQUEST IS                       10/13/87
C*    SENT TO THE APPROPRIATE REMOTE SYSTEM.  THE REMOTE SYSTEM                     10/13/87
C*    IS SELECTED BASED ON THE ITEM NUMBER REQUESTED.                               10/13/87
C*                                                                                  10/13/87
C*    IF AN ERROR OCCURS, THE ERROR IS PRINTED AND THE JOB                          10/13/87
C*    IS ENDED.                                                                     10/13/87
C*                                                                                  10/13/87
C*****************************************************************                  10/13/87
 * 6                                                                                10/13/87
C           ITMIN     TAG                                                           10/13/87
C           *IN99     CABEQ'1'       END             EXIT ON CMD3                   10/13/87
C           *IN98     IFEQ '1'                                                B001    10/13/87
C                     WRITECIMENU                     MAIN MENU                001    10/13/87
C                     GOTO READRQ                                              001    10/13/87
C                     END                                                     E001    10/13/87
C           ITEMNO    CABLE399999    XICF01                                         10/13/87
C           ITEMNO    CABLE699999    XICF02                                         10/13/87
C           ITEMNO    CABLE899999    XICF03                                         10/13/87
C           XICF01    TAG                                                           10/13/87
C                     MOVEL'ICF01   'CMID                                           10/13/87
C                     GOTO XITMIN                                                   10/13/87
C           XICF02    TAG                                                           10/13/87
C                     MOVEL'ICF02   'CMID                                           10/13/87
C                     GOTO XITMIN                                                   10/13/87
C           XICF03    TAG                                                           10/13/87
C                     MOVEL'ICF03   'CMID                                           10/13/87
C           XITMIN    TAG                                                           10/13/87
C                     EXCPTITEMRQ                     INQ W/INV                     10/13/87
C           MAJCOD    CABGE'04'      ERROR           ERROR RTN                      10/13/87
```

Figure 11-21 (Part 5 of 14). Source Program Example — RSFMUL (System-Supplied Formats)

```
C           TRY89     TAG                                                           10/13/87
C                     SETOF                     89                    1             10/13/87
C                     MOVEL' '       RECITM           RESET RECID                   10/13/87
C                     MOVEL'        'CMID                                           10/13/87
C                     READ CMNFIL                 8910                  2 3         10/13/87
C   89                EXSR ERRCHK                     FILE ERROR?                   10/13/87
C           MAJMIN    CABGE'0300'    ITMIN            NODATA?                       10/13/87
C  N81                EXCPTNOTITM                     REC NOT FD?                   10/13/87
C                     SETOF                     81                    1             10/13/87
C           MAJCOD    CABGE'04'      ERROR            ERROR RTN                     10/13/87
C           RECITM    CABNE'I'       RECERR           PRINT MSG                     10/13/87
C***************************************************************                    10/13/87
C*                                                                                  10/13/87
C*                     PROCESS ITEM INFORMATION                                     10/13/87
C*                                                                                  10/13/87
C*    THE ITEM RECORD RECEIVED FROM THE TARGET PROGRAM AND THE                      03/20/89
C*    INFORMATION ABOUT THE ITEM IS PROCESSED AND DISPLAYED. IF                     03/20/89
C*    ITEMNO IS 0 OR LESS, IT IS AN INVALID REQUEST AND A FRESH                     03/20/89
C*    ITEM MENU IS WRITTEN TO THE SCREEN. IF REQUEST IS VALID,                      03/20/89
C*    VALUES ARE CALCULATED BASED ON THE INFORMATION RECEIVED.                      03/20/89
C*                                                                                  10/13/87
C***************************************************************                    10/13/87
 * 7                                                                                03/20/89
C           ITMOUT    TAG                                                           10/13/87
C           ITEMNO    IFLE 000000                                          B001     10/13/87
C                     WRITEITMMNU                     ITEM MENU             001     10/13/87
C                     GOTO READRQ                     READ DISPLAY          001     10/13/87
C                     ELSE                                                 X001     10/13/87
C                     Z-ADD0         QAVAIL  70       QTY AVAIL.            001     10/13/87
C                     ADD  QTYOH     QAVAIL                                 001     10/13/87
C                     SUB  QTYOO     QAVAIL                                 001     10/13/87
C                     ADD  QTYBO     QAVAIL                                 001     10/13/87
C                     MOVELDESC      DSC                                    001     10/13/87
C                     MOVE QTYOO     QTYO                                   001     10/13/87
C                     MOVE QTYOH     QTYH                                   001     10/13/87
C                     MOVE QTYBO     QTYB                                   001     10/13/87
C                     MOVE UNIT      UNT                                    001     10/13/87
C                     MOVE PR01      PR1                                    001     10/13/87
C                     MOVE PR05      PR5                                    001     10/13/87
C                     MOVE UFRT      UFR                                    001     10/13/87
C                     WRITEITMSC2                     ITEM DETAIL           001     10/13/87
C                     GOTO READRQ                                           001     10/13/87
C                     END                                                  E001     10/13/87
C****************************************************************                   10/13/87
C*                                                                                  10/13/87
C*                    ADDITIONAL ITEM INFORMATION                                   10/13/87
C*                                                                                  10/13/87
C*    ADDITIONAL ITEM INFORMATION IS PROCESSED AND THE RESULT                       10/13/87
C*    IS DISPLAYED ON THE SCREEN WHEN A RESPONSE IS READ                            03/20/89
C*    FROM THE DISPLAY WITH AN ITEM SCREEN RECORD FORMAT.                           03/20/89
C*                                                                                  10/13/87
C*    IF CMD 1 (*IN99) IS PRESSED, THE PROGRAM IS ENDED.                            10/13/87
C*                                                                                  10/13/87
C*    IF CMD 2 (*IN98) IS PRESSED, THE ITEM INQUIRY IS ENDED,                       03/20/89
C*    AND THE MAIN MENU (CIMENU) IS WRITTEN TO THE SCREEN.                          03/20/89
C*                                                                                  10/13/87
C*    IF CMD 3 (*IN97) IS PRESSED, THE ITEM INQUIRY MENU IS                         10/14/87
C*    WRITTEN ON THE SCREEN.                                                        10/14/87
C*                                                                                  10/14/87
C*    IF 'ENTER' IS PRESSED WHILE SCREEN 2 FOR ITEM REQUESTED IS                    10/16/87
C*    CURRENTLY DISPLAYED, MORE INFORMATION IS CALCULATED AND                       10/16/87
C*    DISPLAYED.                                                                    10/16/87
C*                                                                                  10/16/87
C*    IF 'ENTER' IS PRESSED WHILE SCREEN 3 FOR ITEM REQUESTED IS                    10/16/87
C*    CURRENTLY DISPLAYED, THEN THE ITEM INQUIRY MENU IS WRITTEN                    03/20/89
C*    TO THE SCREEN.                                                                03/20/89
C*                                                                                  10/13/87
C****************************************************************                   10/13/87
```

Figure 11-21 (Part 6 of 14). Source Program Example — RSFMUL (System-Supplied Formats)

```
29400   *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
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
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   *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
```

Figure 11-21 (Part 7 of 14). Source Program Example — RSFMUL (System-Supplied Formats)

```
C****************************************************************                   10/13/87
C*                                                                                  10/13/87
C*                      PROCESS CUSTOMER INFORMATION                                10/13/87
C*                                                                                  10/13/87
C*     THE CUSTOMER DATA RECEIVED FROM THE TARGET PROGRAM IS                        03/20/89
C*     PROCESSED. IF CUSTOMER NUMBER IS ZERO OR LESS, IT IS AN                      03/20/89
C*     INVALID REQUEST AND THE MAIN MENU IS WRITTEN TO THE SCREEN.                  03/20/89
C*                                                                                  10/14/87
C*     WHEN THE RCVTRNRND INDICATOR(*IN90) IS RECEIVED, THE                         03/20/89
C*     CUSTOMER INFORMATION IS WRITTEN TO THE SCREEN.                               03/20/89
C*                                                                                  10/14/87
C*     IF DURING THE READ OPERATION AN ERROR IS RECEIVED,                           10/13/87
C*     CONTROL GOES TO THE ERROR ROUTINE TO END THE JOB.                            10/13/87
C*                                                                                  10/13/87
C*****************************************************************                  10/13/87
 * 10                                                                               10/13/87
C           DTOUT     TAG                                                           10/13/87
C           CUSTNO    IFEQ 000000                                            B001   10/13/87
C                     SETOF                     66                    3   001   10/13/87
C                     WRITECIMENU                 MAIN MENU               001   10/13/87
C                     GOTO READRQ                                         001   10/13/87
C                     END                                                E001   10/13/87
C                     MOVE CUSTNO    CUSTN                                          10/13/87
C                     MOVELDNAME     CNAME                                          08/08/89
C                     MOVE DLSTOR    DLSTR                                          10/13/87
C                     MOVE DSLSTM    DSLSM                                          10/13/87
C                     MOVE DSPM01    DSPM1                                          10/13/87
C                     MOVE DSPM02    DSPM2                                          10/13/87
C                     MOVE DSTTYD    DSTYD                                          10/13/87
C                     MOVE IDEPT     DEPT                                           10/13/87
C                     WRITEDTLSCR                 DETAIL INFO                       10/13/87
C                     GOTO READRQ                                                   10/13/87
C****************************************************************                   10/13/87
C*                                                                                  10/13/87
C*    THIS ROUTINE HANDLES THE USER'S REQUEST FOLLOWING THE                         10/14/87
C*    DISPLAY OF THE CUSTOMER INFORMATION.  CMD KEY 1 WILL END                      03/20/89
C*    THE JOB, CMD KEY 2 WILL DISPLAY THE MAIN MENU, AND "ENTER"                    03/20/89
C*    WILL BRING UP THE CUSTOMER INQUIRY MENU.                                      03/20/89
C*                                                                                  10/13/87
C****************************************************************                   10/13/87
 * 11                                                                               10/13/87
C           DTLRTN    TAG                                                           10/13/87
C           *IN99     CABEQ'1'       END              JOB ENDS                      10/13/87
C           *IN98     IFEQ '1'                                               B001   10/13/87
C                     WRITECIMENU                 MAIN MENU               001   10/13/87
C                     GOTO READRQ                                         001   10/13/87
C                     END                                                E001   10/13/87
C                     WRITEDTLMNU                 CUSTOMER INQ                      10/13/87
C                     GOTO READRQ                                                   10/13/87
C*****************************************************************                  10/13/87
C*                                                                                  10/13/87
C*    WHEN AN I/O OPERATION ERROR IS DETECTED, A MESSAGE IS PRINTED                 03/20/89
C*    AND THE TRANSACTION AND SESSION ARE ENDED FOR EACH OF THE                     03/20/89
C*    REMOTE SYSTEMS.                                                               03/20/89
C*                                                                                  10/13/87
C*****************************************************************                  10/13/87
 * 12                                                                               10/13/87
C           RECERR    TAG                                                           10/13/87
C                     EXCPTRECER                  WRONG RECID                       10/13/87
C                     GOTO END                    END PROGRAM                       10/13/87
C           ERROR     TAG                                                           10/13/87
C                     EXCPTMMERR                                                    10/13/87
C           END       TAG                                                           10/13/87
C                     MOVEL'ICF00   'CMID                                           10/13/87
C                     EXCPTDETACH                 DET 1ST TRANS                     10/13/87
C                     MOVEL'ICF01   'CMID                                           10/13/87
C                     EXCPTDETACH                 DET 2ND TRANS                     10/13/87
C                     MOVEL'ICF02   'CMID                                           10/13/87
C                     EXCPTDETACH                 DET 3RD TRANS                     10/13/87
C                     MOVEL'ICF03   'CMID                                           10/13/87
C                     EXCPTDETACH                 DET 4TH TRANS                     10/13/87
```

Figure 11-21 (Part 8 of 14). Source Program Example — RSFMUL (System-Supplied Formats)

```
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   * 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   * 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
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                          10/03/90
49300  C*    IS ALREADY ACQUIRED, THE ERROR IS IGNORED, OTHERWISE THE                      03/20/89
49400  C*    PROGRAM IS TERMINATED.                                                        03/20/89
49500  C*                                                                                  10/14/87
49600  C****************************************************************                   10/14/87
49700   * 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
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
```

Figure 11-21 (Part 9 of 14). Source Program Example — RSFMUL (System-Supplied Formats)

```
    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
    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                            CNAME     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
  L000006                            UNT       60  CHAR    2
  L000007                            PR1       67  ZONE  7,2
  L000008                            PR5       74  ZONE  7,0
  L000009                            UFR       79  ZONE  5,2
```

Figure 11-21 (Part 10 of 14). Source Program Example — RSFMUL (System-Supplied Formats)

```
  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.

                         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
```

Figure 11-21 (Part 11 of 14). Source Program Example — RSFMUL (System-Supplied Formats)

```
* 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
        CNAME       A(5)      38900M J000009D
        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
        CUSTN       A(6)      38800M J000001D
        CUSTNO      P(6,0)     6700D B000004D   35200    38300    38800
                              52400
        DEPT        P(3,0)    39500M J000002D
        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
```

Figure 11-21 (Part 12 of 14). Source Program Example — RSFMUL (System-Supplied Formats)

```
* 7031  MAIN        TAG       14100D
        MAJCOD      A(2)      10800D   22500    23500    35500    50700
        MAJMIN      A(4)      10700D   23200    36200    46900    48100
                              48200
        MENU        TAG       15700    17700D
        MINCOD      A(2)      10900D   50900
        MMERR       EXCPT     42700    50400
        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
        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
        '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
```

Figure 11-21 (Part 13 of 14). Source Program Example — RSFMUL (System-Supplied Formats)

```
        '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
 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   * * * * *

                         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   * * * * *

                           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 11-21 (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 11-22 on page 11-66.

```
 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/
   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 11-22. DDS Source for ICF File Used by Target Program Multiple-Session Inquiry

The DDS for the database file (PFILE) is illustrated in Figure 11-23.

```
 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 11-23. 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 11-24 on page 11-68 and Figure 11-25 on page 11-73. 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.

```
 Compiler . . . . . . . . . . . . . :   IBM AS/400 RPG/400
 Command Options:
   Program  . . . . . . . . . . . . :   ICFLIB/RTDMUL
   Source file  . . . . . . . . . . :   ICFLIB/QICFPUB
   Source member  . . . . . . . . . :   *PGM
Text not available for message RXT0073 file QRPGMSG.
   Generation options . . . . . . . :   *NOLIST     *NOXREF     *NOATR      *NODUMP     *NOOPTIMIZE
   Source listing indentation . . . :   *NONE
   SAA flagging . . . . . . . . . . :   *NOFLAG
   Generation severity level  . . . :   9
   Print file . . . . . . . . . . . :   *LIBL/QSYSPRT
   Replace program  . . . . . . . . :   *YES
   Target release . . . . . . . . . :   *CURRENT
   User profile . . . . . . . . . . :   *USER
   Authority  . . . . . . . . . . . :   *LIBCRTAUT
   Text . . . . . . . . . . . . . . :   *SRCMBRTXT
   Phase trace  . . . . . . . . . . :   *NO
   Intermediate text dump . . . . . :   *NONE
   Snap dump  . . . . . . . . . . . :   *NONE
   Codelist . . . . . . . . . . . . :   *NONE
   Ignore decimal data error  . . . :   *NO
 Actual Program Source:
   Member . . . . . . . . . . . . . :   RTDMUL
   File . . . . . . . . . . . . . . :   QICFPUB
   Library  . . . . . . . . . . . . :   ICFLIB
   Last Change  . . . . . . . . . . :   10/03/90  14:50:39
   Description  . . . . . . . . . . :   RPG Multi-Session example w/DDS (target)

                          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, RECORD                   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                                                                                            *****
     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   * 1                                                                                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 11-24 (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  45 FMTNM                                10/13/87
   2300  I                                      273 282 PGMDEV                               10/13/87
   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                        10/03/90
   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                         10/03/90
   3800  C*  INDICATES AN ERROR.                                                             10/13/87
   3900  C*                                                                                  10/13/87
   4000  C***************************************************************                    10/13/87
   4100   * 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   * 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.                                                  10/03/90
   6500  C*                                                                                  10/13/87
   6600  C*****************************************************************                  10/13/87
   6700   * 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
   7600  C*  IS SENT BACK TO THE SOURCE PROGRAM.                                             10/13/87
   7700  C*                                                                                  03/20/89
   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
```

Figure 11-24 (Part 2 of 5). Target Program Example — RTDMUL (User-Defined Formats)

```
     8400  C******************************************************************                 10/13/87
     8500  C*                                                                                  10/13/87
     8600  C* IF ANY ICF FILE ERROR OCCURS, PRINT THE ERROR MESSAGE,                           10/03/90
     8700  C* AND THEN END THE JOB.                                                            10/13/87
     8800  C*                                                                                  10/13/87
     8900  C******************************************************************                 10/13/87
     9000   * 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   * 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 TERMINATED.                                 03/20/89
    10600  C*                                                                                  10/14/87
    10700  C*****************************************************************                  10/14/87
    10800   * 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 '                              10/03/90
    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
  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.

                   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
                         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
```

Figure 11-24 (Part 3 of 5). Target Program Example — RTDMUL (User-Defined Formats)

```
 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
        '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   * * * * *

                        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   * * * * *
```

Figure 11-24 (Part 4 of 5). Target Program Example — RTDMUL (User-Defined Formats)

```
                          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 11-24 (Part 5 of 5). Target Program Example — RTDMUL (User-Defined Formats)

```
 Compiler . . . . . . . . . . . . . :   IBM AS/400 RPG/400
 Command Options:
   Program  . . . . . . . . . . . . :   ICFLIB/RTFMUL
   Source file  . . . . . . . . . . :   ICFLIB/QICFPUB
   Source member  . . . . . . . . . :   *PGM
Text not available for message RXT0073 file QRPGMSG.
   Generation options . . . . . . . :   *NOLIST     *NOXREF     *NOATR      *NODUMP     *NOOPTIMIZE
   Source listing indentation . . . :   *NONE
   SAA flagging . . . . . . . . . . :   *NOFLAG
   Generation severity level  . . . :   9
   Print file . . . . . . . . . . . :   *LIBL/QSYSPRT
   Replace program  . . . . . . . . :   *YES
   Target release . . . . . . . . . :   *CURRENT
   User profile . . . . . . . . . . :   *USER
   Authority  . . . . . . . . . . . :   *LIBCRTAUT
   Text . . . . . . . . . . . . . . :   *SRCMBRTXT
   Phase trace  . . . . . . . . . . :   *NO
   Intermediate text dump . . . . . :   *NONE
   Snap dump  . . . . . . . . . . . :   *NONE
   Codelist . . . . . . . . . . . . :   *NONE
   Ignore decimal data error  . . . :   *NO
 Actual Program Source:
   Member . . . . . . . . . . . . . :   RTFMUL
   File . . . . . . . . . . . . . . :   QICFPUB
   Library  . . . . . . . . . . . . :   ICFLIB
   Last Change  . . . . . . . . . . :   10/03/90  14:52:20
   Description  . . . . . . . . . . :   RPG Multi-Session example w/$$FORMAT (target)

                          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                             38  45 FMTNM                                       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  45 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 11-25 (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
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                       10/03/90
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                        10/03/90
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.                                                  10/03/90
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,                            10/03/90
8800  C* AND THEN END THE JOB.                                                            10/13/87
8900  C*                                                                                  10/13/87
9000  C************************************************************                       10/13/87
9100   *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
```

Figure 11-25 (Part 2 of 4). Target Program Example — RTFMUL (System-Supplied Formats)

```
     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                             10/03/90
    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.                                                                10/03/90
    10600  C*    IF THE PROGRAM DEVICE IS ALREADY ACQUIRED, THE EXCEPTION IS                   03/20/89
    10700  C*    IGNORED, OTHERWISE THE PROGRAM IS TERMINATED.                                 03/20/89
    10800  C*                                                                                  10/14/87
    10900  C*****************************************************************                  10/14/87
    11000   * 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 '                              10/03/90
    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.

                   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
                         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
        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
```

Figure 11-25 (Part 3 of 4). Target Program Example — RTFMUL (System-Supplied Formats)

```
* 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
 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  * * * * *

                         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  * * * * *

                           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 11-25 (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.

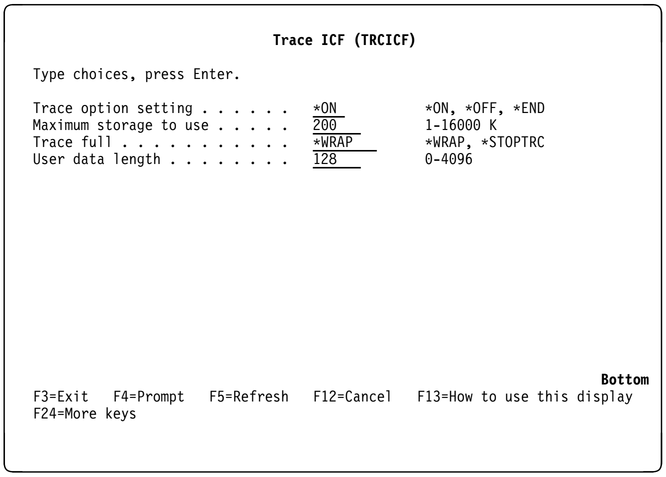
The End Service Job (ENDSRVJOB) command is used to end the service job request. No parameters are used with this command. The trace must be stopped before this command can be entered. The *CL Reference* book has more information about the STRSRVJOB and ENDSRVJOB commands.

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:



```
                                Trace ICF (TRCICF)

 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
```

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.

200KB
The number of bytes (1KB equals 1024 bytes) of maximum storage. This is the default value.

1-16000KB
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.

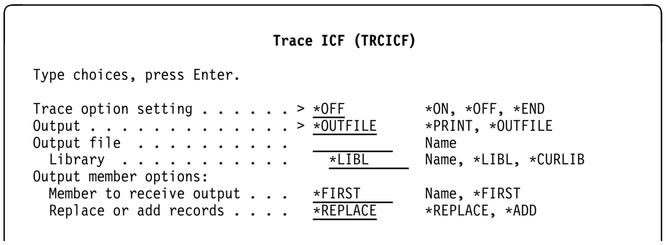
0-4096

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:



```
                           Trace ICF (TRCICF)

 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
```

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 12-1 on page 12-3 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 “Trace Records Sent to a Database File” on page 12-4 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 12-1 shows the format of the spooled trace records.

```
 -----------------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
   CTL              Control-Data
   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
   GTA              Get-Attributes
   INV              Invite
   NRP              Negative-Response
   CPN              Open with Acquire-Program-Device
   PRC              Prepare-to-Commit
   RCF              Respond-to-Confirm
   RCV              Receive
   REL              Release
   RFI              Read-From-Invited-Program-Devices
   RLB              Rollback
   RST              Restore
   RWT              Request-to-Write
   SDV              Subdevice-Selection
   SND              Send
   SPD              Suspend
   TKC              Take-Commit
   TMR              Timer
   TNS              Transaction-Sync-Level
   TRN              Turn-Around

2 Job  . . . :  DSP10     3 User . . . :   QUSER      4 Number . .    . :   00626
5 Program . . . . :  ICFTEST     /ICFMAIN           6 Program File   . . . . :   ICFTEST  /ICFTSTCF
                                                   7 Opened File   . . . . :   ICFLIB    /ICFTSTITF
8 Program      9 Record      10 Return       11                 12 Response          13 Data      14 Remote      15
 Device         Format        Code            Function            Indicator            Length       Location       Time
 DEV1                         0000            ACQ                                           0       Chicago        15:37:00.857
 DEV1           EVOKENOV      0000            EVK                                           0       Chicago        15:37:06.264
 DEV1           NOVARLEN      0001            SND,INV                                      B0       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        15:38:06.887
 DEV1           NOVARLEN      0001            SND,INV                                      B0       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  * * * * *
```

Figure 12-1. 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.

The following example 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

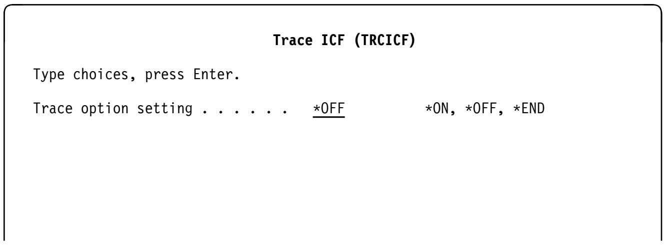
TRCICF command. Database files contain much of the same information that is contained in spooled files. Database files also contain the century and system name. For the 20th century Century would be 0, and for the 21st century Century would be 1, and so on. Date is in the YYMMDD format.

Note: If you want to use QIFTRC as an externally described file in your program, use IFDTL2 rather than IFDTLN for correct size definition of the data length.

```
A*
A* TRACE ICF OUTFILE RECORD FORMAT FOR TRCICF
A*
A   R QIFTRC            TEXT('Trace ICF 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 ICF 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      4B    COLHDG('Traced' 'data' 'length')
A                       TEXT('Traced data length')
A     IFCENT      1     COLHDG('Century')
A                       TEXT('Century of entry')
A     IFDATE      6     COLHDG('Date')
A                       TEXT('Date of entry')
A     IFSYS       8     COLHDG('System' 'name')
A                       TEXT('System name')
A     IFDTL2      5B    COLHDG('Data' 'length')
A                       TEXT('Data length')
A     IFRES       7     COLHDG('Reserved')
A                       TEXT('Reserved')
A     IFDATA   4096     COLHDG('Data')
A                       TEXT('Data')
```

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.



```
                                Trace ICF (TRCICF)

Type choices, press Enter.

Trace option setting . . . . . .   *OFF          *ON, *OFF, *END
```

Additional Considerations

Trace ICF traces only those operations that are associated with your ICF sessions. For example, ICF 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 Trace ICF traces:

- 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.
- When receiving program initialization parameters (sent by means of an evoke operation) in the System/36 environment with a read operation, the *Data length* field in the trace output will be 0, even if data was actually received by your program.
- When using BSCEL with data compression, the actual length of the data received on an input operation is not known by the system after decompression. The traced data length will be 0 and the traced data will not appear in the trace record, even if data was actually received by the program.

Displaying Communications Status

You can obtain current status information about operations and functions directed to an ICF file for all active (acquired) sessions within a job by using the Display Job (DSPJOB) or Work with Job (WRKJOB) commands. Choose option 17, `Display communications status`, from a Display Job or Work with Job display. You can also access this information from the Work with Active Jobs (WRKACTJOB) display. Refer to the *Communications Management* book for more information.

Appendix A. Language Operations, Data Description Specifications Keywords, and System-Supplied Formats

This appendix contains charts that show the following:

- All valid language operations supported by intersystem communications function (ICF)
- All valid operations for each programming language that supports ICF (ILE C, ILE COBOL, FORTRAN/400, and ILE RPG programming languages)
- Data description specifications (DDS) processing keywords supported by communications types
- System-supplied formats supported by communications types

Language Operations

Figure A-1 describes the language operations supported by ICF.

Figure 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 ILE C and ILE RPG programming languages.
Release	Attempts to end a session.
Close	Closes the ICF file.

Figure A-2 shows all the valid communications operations for each programming language that supports ICF (ILE C, ILE COBOL, and ILE RPG programming languages).

Note: When specifying a format name, program device name, or program-to-system parameters in your program, always use uppercase characters.

Figure A-2. Language Operations

ICF Operation	ILE C Function[1]	ILE COBOL/400 Procedure Statement	ILE RPG/400 Operation	FORTRAN/400 Statement
Open	fopen, _Ropen	OPEN	OPEN	OPEN
Acquire	_Racquire	ACQUIRE	ACQ	Not supported[2]
Get-attributes	_Rdevatr	ACCEPT	POST	Not supported
Read	fread, _Rreadn	READ	READ	READ
Read-from-invited-program-devices	_Rreadindv	READ[3]	READ[3]	Not supported
Write	fwrite, _Rwrite	WRITE	WRITE	WRITE
Write/Read	_Rwriterd	Not supported	EXFMT	Not supported
Release	_Rrelease	DROP	REL	Not supported
Close	fclose, _Rclose	CLOSE	CLOSE	CLOSE

[1] ILE C functions and statements are case sensitive.

[2] To acquire a program device using the FORTRAN/400 language, you must specify the program device on the ACQPGMDEV parameter on the CRTICFF, CHGICFF, or OVRICFF commands. The program device will then be implicitly acquired when the ICF file is opened.

[3] 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 operation is issued with a format or terminal specified, the read operation is interpreted as an ICF read operation. Refer to the appropriate language manual for more information.

DDS Keyword Support

Figure A-3 on page A-2 defines the communications processing control DDS keywords supported for the ICF file, and the communications type that supports these keywords.

Note: DDS keywords supported by APPC apply to APPC over TCP/IP, as well.

Figure A-3 (Page 1 of 4). Processing Control DDS Keywords

DDS Keyword	APPC	SNUF	BSCEL	Async	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		
CTLDTA[2] Informs the remote program that control data is being sent.	X						
DETACH Informs the remote program that the sending program is ending the transaction.	X	X	X	X[1]	X		X
DFREVOKE[2] Delays an evoke request until either the output buffer is full or the output buffer is flushed.	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	X

Figure A-3 (Page 2 of 4). Processing Control DDS Keywords

DDS Keyword	APPC	SNUF	BSCEL	Async	Intra-system	Finance	Retail
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
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
PRPCMT Indicates that the remote program is preparing for a synchronization point.	X						
RCVCANCEL[2] Indicates that the remote program sent a cancel request.		X			X		X
RCVCONFIRM[2] Indicates that the remote program is requesting a confirmation of transaction activity.	X	X			X		
RCVCTLDTA[2] Indicates that control data has been received.	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	X
RCVNEGRSP[2] Indicates that the remote program issued a negative-response request.		X			X	X	X
RCVROLLB Indicates if a rollback operation has been received.	X						

Figure A-3 (Page 3 of 4). Processing Control DDS Keywords

DDS Keyword	APPC	SNUF	BSCEL	Async	Intra-system	Finance	Retail
RCVTKCMT Indicates if a take-commit request has been received.	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			X		
SECURITY Includes security information needed to start a program on the remote system.	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
TNSSYNLVL Specifies the transaction synchronization level that is performed while issuing a write operation.	X						
VARBUFMGT[2, 4] Allows the user to send or receive multiple or partial records, rather than just one record, by using one record format per write or read operation.	X						

Figure A-3 (Page 4 of 4). Processing Control DDS Keywords

DDS Keyword	**APPC**	**SNUF**	**BSCEL**	**Async**	**Intra-system**	**Finance**	**Retail**
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 Programming* book 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 Programming* book for details.

[4] Use of this keyword is restricted. Refer to the *APPC Programming* book for more details.

System-Supplied Format Support

Figure A-4 defines the system-supplied formats supported for ICF, and shows the communications types that supports these formats.

Note: System-supplied formats that apply to APPC also apply to APPC over TCP/IP.

Figure A-4. System-Supplied Format Support

Operation	APPC	SNUF	BSCEL	Async	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
$$POSRSP Positive response		X					
$$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	X
$$SENDNF Send FMH		X		X	X	X	X
$$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 Programming* book for more information.

Appendix B. Communications Error Handling

This appendix 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 Management* book.

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 Programming* book for more information about the message types.

Figure B-1 is a summary of the messages, by operation, that can be issued.

Figure B-1. File Error Message Identifier Groups

Operation	**Message Type**	**Message Identifiers**
Open	Diagnostic and status	CPF4001 through CPF40FF
Open	Escapes that make the file unusable	CPF4101 through CPF43FF
Close	Diagnostic and status	CPF4401 through CPF44FF
Close	Escapes that make the file unusable	CPF4501 through CPF46FF
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 CPF48FF and CPF5001 through CPF50FF
Input/Output, Acquire, and Release	Notify with a default reply of ignore	CPF4901 through CPF49FF
Input/Output, Acquire, and Release	Escapes that make the file or program device unusable	CPF5101 through CPF53FF and CPF5501 through CPF56FF

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 book 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 book 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 Retrieve Configuration Status (RTVCFGSTS) command is also available to determine line, controller, and device status. See the *Communications Management* book 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 book 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.
4. Return codes that are used only by applications running in the System/36 environment are not included in this book. See the *Concepts and Programmer's Guide for the System/36 Environment*, SC41-9663, for information about System/36 environment return codes.

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.

Note: Error codes for APPC also apply to APPC over TCP/IP.

Figure B-2 (Page 1 of 2). Major Code 00

Code	APPC	Asynchronous	BSCEL	Finance	Intra-system	Retail	SNUF
0000	X	X	X	X	X	X	X
0001	X		X	X	X	X	X
0002	X						
0003				X	X	X	X
0004	X	X			X		X
0005	X				X	X	X
0006	X						
0007	X			X	X	X	X
0008	X		X		X		X
000C	X						X
0010	X		X		X		X
0011	X						
0013	X						
0014	X				X		
0015	X				X		
0016		X					
0017					X		
0018	X						
001C	X				X		
001D	X						
0020			X				X
0021			X				X
0023							X

Figure B-2 (Page 2 of 2). Major Code 00

Code	APPC	Asynchronous	BSCEL	Finance	Intra-system	Retail	SNUF
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				X		
0045	X				X		
0046	X						
0047					X		
0054	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 or write operation with invite. A turnaround or end-of-transmission indication was not received.

0002 Control-data indication received with program start request.

0003 End-of-group indication received on successful input operation.

0004 Function-management-header or control-data indication and turnaround indication received on successful input operation; or, a PAD message received from a remote PAD.

0005 Function-management-header or control-data indication received on a successful input operation.

0006 Control-data and turnaround indications received with program start request.

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 or control-data indication received, and detach indication received, on a successful input operation.

0010 Request-to-write, reverse-interrupt (RVI), or request-to-change-direction received on successful output operation.

0011 Control-data and detach indications received with program start request.

0013 Control-data and turnaround indications received with program start request. In addition, the remote system requested confirmation.

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.

0018 Control-data indication received with program start request. In addition, the remote system requested confirmation.

001C Detach indication received on successful input operation. In addition, the remote system requested confirmation.

001D Control-data and detach indications received with program start request. 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 or control-data indication received, and turnaround indication received, on a successful input operation. In addition, the remote system requested confirmation.

0045 Function-management-header or control-data indication received on a successful input operation. In addition, the remote system requested confirmation.

0046 Control-data and detach indications received on a 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.

0054 Rollback is required. The transaction program (TP) has entered the rollback-required state.

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).

Figure B-3 (Page 1 of 2). Major Code 02

Code	APPC	Asynchro-nous	BSCEL	Finance	Intra-system	Retail	SNUF
0200	X	X		X	X	X	X
0201	X		X		X		X
0202	X						
0203				X	X	X	X
0204	X	X			X		X
0205	X				X	X	X
0206	X						
0207				X	X	X	X
0208	X		X		X		X
020C	X						X
0211	X						
0213	X						
0214	X				X		
0215	X				X		
0216		X					
0217					X		
0218	X						
021C	X				X		
021D	X						
0220			X				X
0221			X				X
0223							X
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				X		
0245	X				X		
0246	X						
0247					X		
0254	X						
0257	X						
0258	X						

Figure B-3 (Page 2 of 2). Major Code 02

Code	APPC	Asynchro-nous	BSCEL	Finance	Intra-system	Retail	SNUF
0259	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.

0202 Control-data indication received with job ended (controlled) indication on program start request.

0203 End-of-group indication received with job ended (controlled) indication on successful input operation.

0204 Function-management-header or control-data indication and turnaround indication, or a PAD message from a remote PAD, received with job ended (controlled) indication on a successful input operation.

0205 Function-management-header or control-data indication received with job ended (controlled) indication on successful input operation.

0206 Control-data and turnaround indications received with job ended (controlled) indication on program start request.

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 or control-data indication and detach indication received with job ended (controlled) indication on successful input operation.

0211 Control-data and detach indications received with job ended (controlled) indication on program start request.

0213 Control-data and turnaround indications received with job ended (controlled) indication on program start request. In addition, the remote system requested confirmation.

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.

0218 Control-data indication received with job ended (controlled) indication on program start request. 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.

021D Control-data and detach indications received with job ended (controlled) indication on program start request. 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 or control-data indication and turnaround indication with job ended (controlled) indication received on a successful input operation. In addition, the remote system requested confirmation.

0245 Function-management-header or control-data indication with job ended (controlled) indication received on a successful input operation. In addition, the remote system requested confirmation.

0246 Function-management-header or control-data indication and detach indication 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.

0254 Rollback is required. The transaction program (TP) has entered the rollback-required state.

0257 The remote program has issued either a commit operation or a prepare-for-commit function. This requests the local program to respond by issuing a commit operation in order to perform the two-phase commit processing on all protected resources. Also, your job is being ended (controlled).

0258 The remote program has issued an allow-write function with the transaction-synchronization-level function followed by either a commit operation or a prepare-for-commit function. The synchronization level is *COMMIT. Your program will be in send state after issuing a commit operation, once the commit operation completes successfully. Also, your job is being ended (controlled).

0259 The remote program has issued a detach function with the transaction-synchronization-level function followed by either a commit operation or a prepare-for-commit function. The synchronization level is *COMMIT. Your program will be deallocated after issuing a commit operation, once the commit operation completes successfully. Also, your job is being ended (controlled).

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.

Figure B-4. Major Code 03

Code	APPC	Asynchronous	BSCEL	Finance	Intra-system	Retail	SNUF
0300	X	X	X		X	X	X
0301	X		X		X		X
0302	X	X			X		
0303				X	X	X	X
0305	X				X	X	X
0306	X						
0308	X		X		X	X	X
0309	X	X	X	X	X	X	X
030C	X						X
0310	X	X	X	X	X	X	X
0311	X						
0313	X						
0314	X				X		
0315	X				X		
0317					X		
0318	X						
031C	X				X		
031D	X						
0344	X				X		
0345	X				X		
0346	X						
0357	X						
0358	X						
0359	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.

0305 Function-management-header or control indication received with no data on successful input operation.

0306 Control-data and turnaround indications received with no data on program start request.

0308 Detach indication received with no data on successful input operation.

0309 Job ended (controlled) indication received on read-from-invited-program-devices operation.

030C Function-management-header or control-data indication and detach indication received with no data on successful input operation.

0310 Timer interval has ended.

0311 Control-data indication and detach indication received with no data on an input operation.

0313 Control-data and turnaround indications received with no data on program start request. In addition, the remote system requested confirmation.

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.

0318 Control-data indication received with no data on program start request. 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.

031D Control-data and detach indications received with no data on program start request. In addition, the remote system requested confirmation.

0344 Function-management-header or control-data indication and turnaround indication received with no data on a successful input operation. In addition, the remote system requested confirmation.

0345 Function-management-header or control-data indication received with no data on a successful input operation. In addition, the remote system requested confirmation.

0346 Function-management-header or control-data indication and detach indication received with no data on a successful input operation. In addition, the remote system requested confirmation.

0357 The remote program has issued either a commit operation or a prepare-for-commit function. This requests the local program to respond by issuing a commit operation in order to perform the two-phase commit processing on all protected resources.

0358 The remote program has issued an allow-write function with the transaction-synchronization-level function. function followed by either a commit operation or a prepare-for-commit function. The synchronization level is *COMMIT. Your program will be in send state after issuing a commit operation, once the commit operation completes successfully.

0359 The remote program has issued a detach function with the transaction-synchronization-level function followed by either a commit operation or a prepare-for-commit function. The synchronization level is *COMMIT. Your program will be deallocated after issuing a commit operation, once the commit operation completes successfully.

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.

Figure B-5. Major Code 04

Code	APPC	Asynchronous	BSCEL	Finance	Intra-system	Retail	SNUF
0402	X				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.

Figure B-6. Major Codes 08-11

Code	APPC	Asynchronous	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.

Figure B-7 (Page 1 of 2). Major Code 34

Code	APPC	Asynchronous	BSCEL	Finance	Intra-system	Retail	SNUF
3401			X	X	X	X	X
3421	X						
3422	X						

Figure B-7 (Page 2 of 2). Major Code 34

Code	APPC	Asynchronous	BSCEL	Finance	Intra-system	Retail	SNUF
3431	X						
3441	X	X	X	X	X	X	X
3451	X	X	X	X	X	X	X
3461	X						
3471	X						
3481	X						

Code **Description**

3401 Input operation rejected because data received was too long for your program's input buffer.

3421 Control-data indication received. 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.

3422 Control-data indication received with program start request. 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.

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 indicators received, or it is not large enough for the indicators received.

3461 Partial record received because remote system sent an error condition before completing the record.

3471 An input exception occurred because the program received data that exceeded its maximum record length. The data has been retained and will be returned on subsequent input operations. The local application program continues to receive data.

3481 Control-data indication received. An input exception occurred because the program received data that exceeded its maximum record length. The data has not been truncated. The local application program may continue to receive the remaining data on subsequent input operations.

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.

Figure B-8 (Page 1 of 2). Major Code 80

Code	APPC	Asynchronous	BSCEL	Finance	Intra-system	Retail	SNUF
8081	X	X	X	X	X	X	X
8082	X	X	X	X	X	X	X

Figure B-8 (Page 2 of 2). Major Code 80

Code	APPC	Asynchro-nous	BSCEL	Finance	Intra-system	Retail	SNUF
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
80F9	X						
80FA	X						
80FB	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.

80F9 The operation attempted by your program was not successful because a system error condition was detected. Rollback required.

80FA The operation attempted by your program was not successful because the device supporting communications between your program and the partner location is not usable. For example, this may have occurred because communications were stopped for the device by a Hold Communications Device (HLDCMNDEV) command. Your program should not issue any operations to the device. Rollback required.

80FB An irrecoverable error has occurred on the session. The session was ended abnormally either by the partner system or because of a partner protocol error. Rollback required.

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.

Figure B-9. Major Code 81

Code	APPC	Asynchronous	BSCEL	Finance	Intrasystem	Retail	SNUF
8101	X						
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
81F0	X						
81F1	X						
81F2	X						
81F3	X						
81F4	X						
81F5	X						

Code **Description**

8101 Protected password could not be built.

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.

81F0 A network interface, permanent line, or controller error occurred on an input or output operation, and the system operator attempted recovery in response to the error message. You can learn what type of error occurred by checking the system operator message queue (QSYSOPR). The session has ended. Data may have been lost. Rollback required.

81F1 The partner system sent a Systems Network Architecture (SNA) UNBIND command to your system, or the session was ended locally. Rollback required.

81F2 On an input or output operation, the partner system ended the transmission abnormally because it could not continue the session. The session has ended. Rollback required.

81F3 The partner program or partner system abnormally ended the session (TYPE=SVC). Rollback required.

81F4 The partner program or partner system abnormally ended the session (TYPE=TIMER). For example, the partner program may have been canceled by the operator. Rollback required.

81F5 An input operation was issued and the format selection option for the ICF file was *RECID, but the data received did not match any record formats in the file. There was no format in the file defined without a RECID keyword, so there was no default record format to use. The session has ended. Rollback required.

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.

Figure B-10 (Page 1 of 2). Major Code 82

Code	APPC	Asynchronous	BSCEL	Finance	Intrasystem	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
82AC			X				
82AD				X		X	
82B3	X	X	X	X		X	X
82B4							X
82B5							X
82BB							X
82C3	X						
82EA	X	X	X	X	X	X	X

Figure B-10 (Page 2 of 2). Major Code 82

Code	APPC	Asynchro-nous	BSCEL	Finance	Intra-system	Retail	SNUF
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
82FA	X						
82FB	X						
82FC	X						
82FD	X						
82FE	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, was already acquired, or a CPI-Communications requesting conversation was already allocated.

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.

82AC Acquire operation failed because the remote location specified for the device was not *REQUESTER.

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 program interface.

82B5 Operation not successful because SNA 3270 program interface 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.

82FA The local LU rejected the allocation request because the local program specified a synchronization level (on the evoke function) that the remote LU does not support.

82FB Protected conversations are not supported on single session devices.

82FC Protected conversations are not supported by the System/36 and System/38 environments.

82FD The exchange log name process failed.

82FE The evoke function issued by your program was not successful because a resource could not be placed under commitment control.

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.

Figure B-11 (Page 1 of 2). Major Code 83

Code	APPC	Asynchronous	BSCEL	Finance	Intrasystem	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
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
8331					X		X
8332							X
8334	X		X		X	X	
83B6				X		X	X
83B7							X
83B8							X
83C7	X						

Figure B-11 (Page 2 of 2). Major Code 83

Code	APPC	Asynchro-nous	BSCEL	Finance	Intra-system	Retail	SNUF
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						
83FB	X						
83FC	X						
83FD	X						
83FE	X						
83FF	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.

83FB Your program closed the ICF file while the transaction was still active. The system abnormally ended the transaction with the partner program. Rollback required.

83FC Your program attempted to issue an operation to a program device that is marked in error due to a previous I/O or acquire operation. Your program may have handled the error incorrectly. Rollback required.

83FD All protected resources have rolled back in the part of the distributed transaction affected by the function.

83FE The state of one or more protected resources is not known. The changes probably are or will be rolled back, but changes to some resources may be committed instead. Your protected LUW is in the rollback required state.

83FF The state of the protected resources is not consistent. One or more resources have advanced to a new synchronization point (have been committed instead of rolled back). Your protected LUW is in the rollback required state.

This return code occurs only when processing has been abnormally interrupted through operator intervention.

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 system 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.

Appendix C. Open Feedback and I/O Feedback

This appendix contains information concerning the open feedback and I/O feedback areas.

If you are using the ILE COBOL, or ILE RPG support, you need to add the offset values shown in Figure C-1 to the offset values listed in this appendix to access information from the feedback areas.

For the ILE C programming language, the Open Feedback Area is accessed with a call to the _Ropnfbk function. The Common I/O Feedback Area is accessed with a call to the _Riofbk function. The File Dependent Feedback Area is accessed by adding an offset in the Common I/O Feedback Area to a pointer to the Common I/O Feedback Area. See the *ILE C/400 Programmer's Guide* for more information.

Open Feedback Area

You can use the open feedback area 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 book for more information.

The complete open feedback area is described in the *Data Management* book. Figure C-2 shows the fields relevant to ICF support.

Figure C-1. Offset Values for ILE COBOL, and ILE RPG

Language	Open Feedback Area	Common I/O Feedback Area	File Dependent Feedback Area
ILE COBOL	0[1]	0	144
FORTRAN/400	0[1]	0	144[2]
ILE RPG	81	241	367

[1] Separate structure from the Common I/O Feedback Area.

[2] Use offset in the first 2 bytes of the Common I/0 Feedback Area to get to the File Dependent Feedback Area.

Figure 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.
133	Character	2	Open identifier. This value is unique for a full open of a file (SHARE (* NO)) or the first open of a file with (SHARE (* YES)). It allows the user to match this file to an entry on the associated data queue.
135	Binary	2	Maximum record length. This value includes the data, source sequence numbers, option indicators, response indicators, and P-data, if applicable. If this field is 0, then use the field from offset 44.
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 Figure C-3.

Program Device Definition List

Figure C-3 on page C-2 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.

Figure C-3. 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: 0A BSCEL device 0E APPC device 1E Intrasystem device 1F Asynchronous device 20 SNUF device 42 Finance device 43 Retail device
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
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. Hex 02 The transaction was started with SYNLVL(*COMMIT). Confirm processing and two-phase commit processing are 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 support provided by the high-level language you are using determines how to access this information. See the appropriate communications language reference book for more information.

The 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* book. Figure C-4 shows the fields relevant to ICF.

Figure C-4. Common I/O Feedback Area

Offset	Data Type	Length in Bytes	Contents
0	Binary	2	Offset to device-dependent feedback area. Refer to Figure 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.
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, not including any indicators or program-to- system fields (P-data fields). For Evoke operations, this field specifies the length of any user-defined parameters.
46	Reserved	82	Not applicable to ICF communications.
128	Binary	2	Length of the record associated with the last I/O operation. This value includes data, option indicators, response indicators, and P-data, if applicable.
130	Reserved	14	Not applicable to ICF communications.

File-Dependent I/O Feedback Area

Figure C-5 shows the communications type-dependent fields relevant to ICF.

Figure C-5. 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 (for example, if DTACPR(*YES) is specified for BSCEL), 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
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 4,5,6,7		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
	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 4,5,6,7		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
	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)**. FTS is a function of the operating system that moves the file members from one system to another using asynchronous, advanced program-to-program communications (APPC), or binary synchronous communications equivalence link (BSCEL) communications support.

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.

Note: The database file is created with *NOMAX as the maximum member (MAXMBR) value. 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.

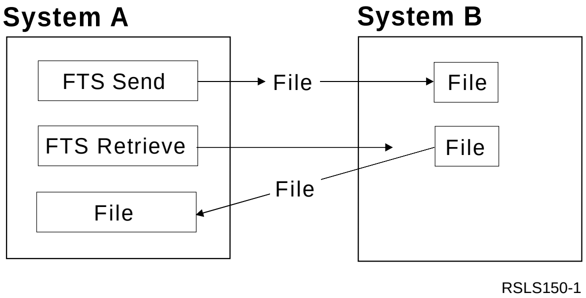


Figure E-1. Example of File Transfer Support

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 ILE C, 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.

Data Compression

The System/36 requires FTS data to be compressed. Except for token-ring lines, the AS/400 system does not compress data when the line speed (LINESPEED parameter on the line description) is set greater than or equal to 56000 bits per second (bps). Regardless of line speed or type, the optional COMPRESS parameter can be used to specify whether data compression is done.

For token-ring networks, FTS always compresses data.

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
- Physical files being sent to a receiving AS/400 system, where the receiving AS/400 system has logical files built over the physical files

  Note: The logical files must be deleted by the user, then the FTS can successfully transfer the physical files.
- 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
- Keyed files with multiple key fields

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 keeps 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
  - Integrated services digital network (ISDN)
  - Token-ring and Ethernet local area networks
  - Advanced Peer-to-Peer Networking (APPN) capability
  - Logical unit (LU) own

  Notes:

  1. If the optional MODE parameter is not specified, FTS requires use of the BLANK mode description. If you are using APPN support to automatically create APPC device descriptions, use the Display Network Attributes (DSPNETA) command to ensure that the default mode description is set to BLANK.
  2. If the optional RMTNETID parameter is not specified and in an APPN network, FTS cannot send files to a system with a different network identifier than the local system. The remote network identifier specified for APPC controller and device descriptions used by FTS must be the same as the local network identifier (LCLNETID parameter) specified in the network attributes.
  3. FTS also runs on APPC over TCP/IP using the links that TCP/IP supports.

- 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 data terminal equipment (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 Programming* book 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, see the *Communications Configuration* book. 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* book for general rules about naming libraries, database files, and database file members. Refer to the *System/36 System Reference* book for System/36 naming conventions.

To and From an AS/400 System

Figure E-2 describes the required parameters for sending and retrieving files between one AS/400 system and another AS/400 system.

Figure E-2 (Page 1 of 3). Transferring Files to and from an AS/400 System—Required Parameters

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

Figure E-2 (Page 2 of 3). Transferring Files to and from an AS/400 System—Required Parameters

Parameter	Value	Description
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
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.
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.

Figure E-2 (Page 3 of 3). Transferring Files to and from an AS/400 System—Required Parameters

Parameter	**Value**	**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-34 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-34 for more information.)

Figure E-3 describes the optional parameters for sending and retrieving files between one AS/400 system and another AS/400 system.

Figure E-3 (Page 1 of 2). Transferring Files to and from an AS/400 System—Optional Parameters

Parameter	**Value**	**Description**
RMTNETID	Remote network identifier	This field contains the network ID of the network where the remote location resides. Length: 8 characters Type: Input, not required Valid values are as follows: • Remote network ID • *LOC • *NETATR • *NONE Default: *LOC
MODE	Mode name	This field contains the mode name used. Length: 8 characters Type: Input, not required Valid values are as follows: • Mode name • *NETATR Default: *NETATR

Figure E-3 (Page 2 of 2). Transferring Files to and from an AS/400 System—Optional Parameters

Parameter	Value	Description
LCLLOCNAME	Local location name	This field contains the local location name. Length: 8 characters Type: Input, not required Valid values are as follows: • Mode name • *LOC • *NETATR Default: *LOC
COMPRESS	Compression indicator	Indicates if data compression will be done. Length: 1 character Type: Input, not required Valid values are as follows: • Y—Compress data • N—Do not compress data • L—Data compression is determined by the speed specified on the line description. "Data Compression" on page E-1 has more information about data compression. Default: L
WAITFILE	Maximum file wait time	This field contains the number of seconds that the program waits for the Intersystem Communications Function (ICF) file resources to be allocated when the file is opened. Length: 6 characters Type: Input, not required Valid values are as follows: – 000001 through 032767 seconds in character format (leading zeros are required) – *IMMED – *CLS Default: 000030

AS/400 System Sending to System/36

Figure E-4 describes the required 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.

Figure E-4 (Page 1 of 3). AS/400 System Sending to System/36—Required Parameters

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-9. 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

Figure E-4 (Page 2 of 3). AS/400 System Sending to System/36—Required Parameters

Parameter	Value	Description
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.
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

Figure E-4 (Page 3 of 3). AS/400 System Sending to System/36—Required Parameters

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-34 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-34 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.

Figure E-5 describes the optional parameters for sending from an AS/400 system to System/36.

Figure E-5. Transferring Files to and from an AS/400 System—Optional Parameters

Parameter	Value	Description
RMTNETID	Remote network identifier	This field contains the network ID of the network where the remote location resides. Length: 8 characters Type: Input, not required Valid values are as follows: • Remote network ID • *LOC • *NETATR • *NONE Default: *LOC
MODE	Mode name	This field contains the mode name used. Length: 8 characters Type: Input, not required Valid values are as follows: • Mode name • *NETATR Default: *NETATR
LCLLOCNAME	Local location name	This field contains the local location name. Length: 8 characters Type: Input, not required Valid values are as follows: • Mode name • *LOC • *NETATR Default: *LOC
COMPRESS	Compression indicator	Indicates if data compression will be done. Length: 1 character Type: Input, not required Valid values are as follows: • Y—Compress data • N—Do not compress data • L—Data compression is determined by the speed specified on the line description. "Data Compression" on page E-1 has more information about data compression. Default: L
WAITFILE	Maximum file wait time	This field contains the number of seconds that the program waits for the Intersystem Communications Function (ICF) file resources to be allocated when the file is opened. Length: 6 characters Type: Input, not required Valid values are as follows: – 000001 through 032767 seconds in character format (leading zeros are required) – *IMMED – *CLS Default: 000030

AS/400 System Retrieving a File from System/36

Figure E-6 describes the required parameters for retrieving a file from System/36.

Figure E-6 (Page 1 of 2). AS/400 System Retrieving a File from System/36—Required Parameters

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-6. 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
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.

Figure E-6 (Page 2 of 2). AS/400 System Retrieving a File from System/36—Required Parameters

Parameter	Value	Description
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
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-34 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-34 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.

Figure E-7 describes the optional parameters for receiving a file from System/36.

Figure E-7. Retrieving Files from System/36—Optional Parameters

Parameter	Value	Description
RMTNETID	Remote network identifier	This field contains the network ID of the network where the remote location resides. Length: 8 characters Type: Input, not required Valid values are as follows: • Remote network ID • *LOC • *NETATR • *NONE Default: *LOC
MODE	Mode name	This field contains the mode name used. Length: 8 characters Type: Input, not required Valid values are as follows: • Mode name • *NETATR Default: *NETATR
LCLLOCNAME	Local location name	This field contains the local location name. Length: 8 characters Type: Input, not required Valid values are as follows: • Mode name • *LOC • *NETATR Default: *LOC
COMPRESS	Compression indicator	Indicates if data compression will be done. Length: 1 character Type: Input, not required Valid values are as follows: • Y—Compress data • N—Do not compress data • L—Data compression is determined by the speed specified on the line description. “Data Compression” on page E-1 has more information about data compression. Default: L
WAITFILE	Maximum file wait time	This field contains the number of seconds that the program waits for the Intersystem Communications Function (ICF) file resources to be allocated when the file is opened. Length: 6 characters Type: Input, not required Valid values are as follows: – 000001 through 032767 seconds in character format (leading zeros are required) – *IMMED – *CLS Default: 000030

Retrieving a Library Member from System/36

Figure E-8 describes the required parameters for retrieving a library member from System/36.

Figure E-8 (Page 1 of 2). Retrieving a Library Member from System/36—Required Parameters

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-6. 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
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.

Figure E-8 (Page 2 of 2). Retrieving a Library Member from System/36—Required Parameters

Parameter	Value	Description
TODATE	Blanks	Not needed in retrieving a library member from System/36. Length: 6 characters Type: Input, 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. • 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-34 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-34 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.

Figure E-9 describes the optional parameters for retrieving a library member from System/36.

Figure E-9. Retrieving a library member from System/36—Optional Parameters

Parameter	Value	Description
RMTNETID	Remote network identifier	This field contains the network ID of the network where the remote location resides. Length: 8 characters Type: Input, not required Valid values are as follows: • Remote network ID • *LOC • *NETATR • *NONE Default: *LOC
MODE	Mode name	This field contains the mode name used. Length: 8 characters Type: Input, not required Valid values are as follows: • Mode name • *NETATR Default: *NETATR
LCLLOCNAME	Local location name	This field contains the local location name. Length: 8 characters Type: Input, not required Valid values are as follows: • Mode name • *LOC • *NETATR Default: *LOC
COMPRESS	Compression indicator	Indicates if data compression will be done. Length: 1 character Type: Input, not required Valid values are as follows: • Y—Compress data • N—Do not compress data • L—Data compression is determined by the speed specified on the line description. “Data Compression” on page E-1 has more information about data compression. Default: L
WAITFILE	Maximum file wait time	This field contains the number of seconds that the program waits for the Intersystem Communications Function (ICF) file resources to be allocated when the file is opened. Length: 6 characters Type: Input, not required Valid values are as follows: – 000001 through 032767 seconds in character format (leading zeros are required) – *IMMED – *CLS Default: 000030

Calling File Transfer Support for the ILE C Programming Language

Figure E-10 is an example of a ILE C 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-3.

```
                                         * * * * *   P R O L O G   * * * * *
Module  . . . . . . . . . . . . :   CDRIVER
  Library . . . . . . . . . . . :     KPSLIB
Source file . . . . . . . . . . :   QCSRC
  Library . . . . . . . . . . . :     KPSLIB
Source member . . . . . . . . . :   CDRIVER
Text Description  . . . . . . . :
Output  . . . . . . . . . . . . :   *PRINT
Compiler options  . . . . . . . :   *NOAGR   *NOEXPMAC   *GEN   *LOGMSG   *NOPPONLY   *NOSECLVL
                                :   *NOSHOWINC   *NOSHOWSKP   *NOXREF   *USRINCPATH
Checkout options  . . . . . . . :   *NOACCURACY   *NOENUM   *NOEXTERN   *NOGENERAL   *NOGOTO   *NOINIT
                                :   *NOPARM   *NOPORT   *NOPPCHECK   *NOPPTRACE
Optimization  . . . . . . . . . :   *NONE
Inline options:
  Inliner . . . . . . . . . . . :   *OFF
  Mode  . . . . . . . . . . . . :   *NOAUTO
  Threshold . . . . . . . . . . :   250
  Limit . . . . . . . . . . . . :   2000
Debugging view  . . . . . . . . :   *NONE
Define names  . . . . . . . . . :
Language level  . . . . . . . . :   *SOURCE
Source margins:
  Left margin . . . . . . . . . :   1
  Right margin  . . . . . . . . :   32754
Sequence columns:
  Left Column . . . . . . . . . :
  Right Column  . . . . . . . . :
Message flagging level  . . . . :   0
Compiler messages:
  Message limit . . . . . . . . :   *NOMAX
  Message limit severity  . . . :   30
Replace program object  . . . . :   *YES
Authority . . . . . . . . . . . :   *LIBCRTAUT
Target release  . . . . . . . . :   *CURRENT
System includes . . . . . . . . :   *YES
Last change . . . . . . . . . . :   06/08/94 12:56:15
Source description  . . . . . . :
Compiler  . . . . . . . . . . . :   IBM ILE C/400 Compiler

                                         * * * * *   S O U R C E   * * * * *
Line  STMT                                                                                             SEQNBR   INCNO
   1       |#pragma linkage(QY2FTML,OS)                /* Define program to be called */              |     1
   2       |                                                                                          |     2
   3       |#define END 1                              /* Signals program end */                      |     3
   4       |#define NOEND 0                                                                           |     4
   5       |#define ERROR 1                            /* Signals an error during I/O */              |     5
   6       |#define NOERROR 0                                                                         |     6
   7       |#include <stdio.h>                         /* Standard I/O header */                      |     7
   8       |#include <stdlib.h>                        /* General utilities */                        |     8
   9       |#include <stddef.h>                        /* Standard definitions */                     |     9
  10       |#include <string.h>                        /* String handling utilities */                |    10
  11       |                                                                                          |    11
```

Figure E-10 (Part 1 of 7). ILE C Coding for File Transfer Support

```
/*--------------------------------------------------------------------*/
/* Define header structures to be written to the print file.          */
/*--------------------------------------------------------------------*/

struct {
    char filler??(124??);
} header_line_1 = {
             "     "
             "FRMLIB     "
             "FRMFIL     "
             "FRMMBR     "
             "TYPE     "
             "TOLIB      "
             "TOFIL      "
             "TOMBR      "
             "TODATE   "
             "OPTION "
             "REPL "
             "RMTLOC   "
             "RCODE "
             "MSGNUM  "
};

struct {
    char filler??(124??);
} header_line_2 = {
             "     "
             "__________ "
             "__________ "
             "__________ "
             "________ "
             "__________ "
             "__________ "
             "__________ "
             "________ "
             "______ "
             "____ "
             "________ "
             "_____ "
             "_______ "
};

/*--------------------------------------------------------------------*/
/* Define data file structure that contains the values to be assigned  */
/* to parameters passed on call to program QY2FTML.                    */
/*--------------------------------------------------------------------*/

struct {
    char rec_num??(3??);
    char option;
    char repl;
    char filler1??(4??);
    char frmlib??(10??);
    char frmfil??(10??);
    char frmmbr??(10??);
    char typ??(6??);
    char filler2??(4??);
    char tolib??(10??);
    char tofil??(10??);
    char tombr??(10??);
    char todate??(6??);
    char filler3??(4??);
    char rmtloc??(8??);
    char passwd??(10??);
    char rcode;
    char msgnum??(8??);
} call_rec;

```

Figure E-10 (Part 2 of 7). ILE C Coding for File Transfer Support

```
/*----------------------------------------------------------------------*/
/* Define the structure types of the parameters on call to QY2FTML. The */
/* type definition is needed for prototyping.                           */
/*----------------------------------------------------------------------*/

typedef struct {
    char option;
} option;

typedef struct {
    char frmlib??(10??);
} lib;

typedef struct {
    char frmfil??(10??);
} file;

typedef struct {
    char frmmbr??(10??);
} file_member;

typedef struct {
    char typ??(6??);
} type;

typedef struct {
    char tolib??(10??);
} tgt_lib;

typedef struct {
    char tofil??(10??);
} tgt_file;

typedef struct {
    char tombr??(10??);
} tgt_member;

typedef struct {
    char todate??(6??);
} tgt_file_date;

typedef struct {
    char repl;
} replace_member;

typedef struct {
    char rmtloc??(8??);
} rmt_loc;

typedef struct {
    char passwd??(10??);
} pword;

typedef struct {
    char rcode;
} ret_code;

typedef struct {
    char msgnum??(8??);
} msg_num;

```

Figure E-10 (Part 3 of 7). ILE C Coding for File Transfer Support

```
/*----------------------------------------------------------------------*/
/* Define structure for data to be written to the print file.           */
/*----------------------------------------------------------------------*/

struct {
    char prec_num??(3??);
    char filler1??(2??);
    char pfrmlib??(10??);
    char filler2;
    char pfrmfil??(10??);
    char filler3;
    char pfrmmbr??(10??);
    char filler4;
    char ptyp??(6??);
    char filler5??(3??);
    char ptolib??(10??);
    char filler6;
    char ptofil??(10??);
    char filler7;
    char ptombr??(10??);
    char filler8;
    char ptodate??(6??);
    char filler9??(5??);
    char poption;
    char filler10??(5??);
    char prepl;
    char filler11??(3??);
    char prmtloc??(8??);
    char filler12??(3??);
    char prcode;
    char filler13??(3??);
    char pmsgnum??(8??);
} print_rec;

/*----------------------------------------------------------------------*/
/* Declare structures to use as parameters on call to QY2FTML using the */
/* type definitions already defined.                                    */
/*----------------------------------------------------------------------*/

option         option_parm;
lib            lib_parm;
file           file_parm;
file_member    file_member_parm;
type           type_parm;
tgt_lib        tgt_lib_parm;
tgt_file       tgt_file_parm;
tgt_member     tgt_member_parm;
tgt_file_date  tgt_file_date_parm;
replace_member replace_member_parm;
rmt_loc        rmt_loc_parm;
pword          pword_parm;
ret_code       ret_code_parm;
msg_num        msg_num_parm;

char op_name??(5??);                        /* Current operation */

extern void QY2FTML(option *, lib *, file *, file_member *, type *,
                    tgt_lib *, tgt_file *, tgt_member *, tgt_file_date *,
                    replace_member *, rmt_loc *, pword *, ret_code *,
                    msg_num *);
int print_header(FILE *);
int file_trans(FILE *);
void init_parms(void);
int print_parms(FILE *);
int pos_ret_code_printf(void);
void print_file_error(void);
void close_files(FILE *, FILE *);

```

Figure E-10 (Part 4 of 7). ILE C Coding for File Transfer Support

```
main()
{
    FILE  *dtafptr;                         /* Pointer to the database file */
    FILE  *prtfptr;                         /* Pointer to the printer file */

    if ((dtafptr = fopen("FTTEST", "rb type=record")) == NULL) {
        printf("\nUNEXPECTED ERROR WHILE OPENING DATA FILE.\n");
        exit(ERROR);
    }
    if ((prtfptr = fopen("QSYSPRT", "wb type=record")) == NULL) {
        fclose(dtafptr);
        printf("\nUNEXPECTED ERROR WHILE OPENING PRINT FILE.\n");
        exit(ERROR);
    }
    if (print_header(prtfptr) == NOERROR) {
        while (1) {
            if (file_trans(dtafptr) == END)
                break;
            else
                if (print_parms(prtfptr) == ERROR) {
                    print_file_error();
                    close_files(dtafptr, prtfptr);
                    exit(ERROR);
                }
        }
    }
    else {
       print_file_error();
       close_files(dtafptr, prtfptr);
       exit(ERROR);
   }
   close_files(dtafptr, prtfptr);
   exit(NOERROR);
}


/*----------------------------------------------------------------------*/
/* The routine prints a header to the print file.                       */
/*----------------------------------------------------------------------*/

print_header(FILE *prtfptr)
{
    strcpy(op_name, "WRITE");
    fwrite(&header_line_1, sizeof(header_line_1), 1, prtfptr);
    if (pos_ret_code_printf() == NOERROR) {
        fwrite(&header_line_2, sizeof(header_line_2), 1, prtfptr);
        return(pos_ret_code_printf());
    }
    else
        return(ERROR);
}
```

Figure E-10 (Part 5 of 7). ILE C Coding for File Transfer Support

```
/*----------------------------------------------------------------------*/
/* The routine gets parameters from the data file and calls QY2FTML.    */
/*----------------------------------------------------------------------*/

file_trans(FILE *dtafptr)
{
    int len;

    strcpy(op_name, "READ ");
    if ((len = fread(&call_rec, sizeof(call_rec), 1, dtafptr)) == 0)
        return(END);
    else {
        init_parms();
        QY2FTML(&option_parm, &lib_parm, &file_parm, &file_member_parm,
                &type_parm, &tgt_lib_parm, &tgt_file_parm, &tgt_member_parm,
                &tgt_file_date_parm, &replace_member_parm, &rmt_loc_parm,
                &pword_parm, &ret_code_parm, &msg_num_parm);
        return(NOEND);
        }
}


/*----------------------------------------------------------------------*/
/* This routine initializes the parameters for the call to QY2FTML.     */
/* Parameters passed to external programs in C/400 must be a structure  */
/* type, so the fields of the structure call_rec may not be sent indiv- */
/* idually to QY2FTML.                                                   */
/*----------------------------------------------------------------------*/

void init_parms()
{
    option_parm.option = call_rec.option;
    strcpy(lib_parm.frmlib, call_rec.frmlib);
    strcpy(file_parm.frmfil, call_rec.frmfil);
    strcpy(file_member_parm.frmmbr, call_rec.frmmbr);
    strcpy(type_parm.typ, call_rec.typ);
    strcpy(tgt_lib_parm.tolib, call_rec.tolib);
    strcpy(tgt_file_parm.tofil, call_rec.tofil);
    strcpy(tgt_member_parm.tombr, call_rec.tombr);
    strcpy(tgt_file_date_parm.todate, call_rec.todate);
    replace_member_parm.repl = call_rec.repl;
    strcpy(rmt_loc_parm.rmtloc, call_rec.rmtloc);
    strcpy(pword_parm.passwd, call_rec.passwd);
    ret_code_parm.rcode = ' ';
    strcpy(msg_num_parm.msgnum, "        ");
}


/*----------------------------------------------------------------------*/
/* This routine prints the parameters passed to QY2FTML after the call. */
/*----------------------------------------------------------------------*/

print_parms(FILE *prtfptr)
{
    strncpy(print_rec.prec_num, call_rec.rec_num, 3);
    print_rec.poption = call_rec.option;
    print_rec.prepl = call_rec.repl;
    strncpy(print_rec.pfrmlib, lib_parm.frmlib, 10);
    strncpy(print_rec.pfrmfil, file_parm.frmfil, 10);
    strncpy(print_rec.pfrmmbr, file_member_parm.frmmbr, 10);
    strncpy(print_rec.ptyp, type_parm.typ, 6);
    strncpy(print_rec.ptolib, tgt_lib_parm.tolib, 10);
    strncpy(print_rec.ptofil, tgt_file_parm.tofil, 10);
    strncpy(print_rec.ptombr, tgt_member_parm.tombr, 10);
    strncpy(print_rec.ptodate, tgt_file_date_parm.todate, 6);
    strncpy(print_rec.prmtloc, rmt_loc_parm.rmtloc, 8);
    print_rec.prcode = ret_code_parm.rcode;
```

Figure E-10 (Part 6 of 7). ILE C Coding for File Transfer Support

```
  14 |    strncpy(print_rec.pmsgnum, msg_num_parm.msgnum, 8);
  15 |    strncpy(print_rec.filler1, "  ", 2);
  16 |    strncpy(print_rec.filler5, "   ", 3);
  17 |    strncpy(print_rec.filler9, "     ", 5);
  18 |    strncpy(print_rec.filler10, "     ", 5);
  19 |    strncpy(print_rec.filler11, "   ", 3);
  20 |    strncpy(print_rec.filler12, "   ", 3);
  21 |    strncpy(print_rec.filler13, "   ", 3);
     |    print_rec.filler2 = print_rec.filler3 = print_rec.filler4
  22 |    = print_rec.filler6 = print_rec.filler7 = print_rec.filler8 = ' ';
  23 |    strcpy(op_name, "WRITE");
  24 |    fwrite(&print_rec, sizeof(print_rec), 1, prtfptr);
  25 |    return(pos_ret_code_printf());
     |}
     |
     |
     |/*------------------------------------------------------------------*/
     |/* This routine checks to see if the last operation on the print file   */
     |/* was successful.                                                  */
     |/*------------------------------------------------------------------*/
     |
     |pos_ret_code_printf()
     |{
   1 |    if (strncmp(_Maj_Min_rc.major_rc, "00", 2) == NOERROR)
   2 |        return(NOERROR);
     |    else
   3 |        return(ERROR);
     |}
     |
     |
     |/*------------------------------------------------------------------*/
     |/* This routine prints an error message to the display.             */
     |/*------------------------------------------------------------------*/
     |
     |void print_file_error()
     |{
   1 |        printf("\nUNEXPECTED ERROR ON %s FOR PRINT FILE\n", op_name);
     |}
     |/*------------------------------------------------------------------*/
     |/* Close the data and print files.                                  */
     |/*------------------------------------------------------------------*/
     |
     |void close_files(FILE *dtafptr, FILE *prtfptr)
     |{
   1 |    fclose(dtafptr);
   2 |    fclose(prtfptr);
     |}
     |
                                         * * * * *   E N D   O F   S O U R C E   * * * * *

                                              * * * * *   I N C L U D E S   * * * * *
INCNBR  Include Name                     Last change        Actual Include Name
    1   stdio.h                          04/19/94 14:18:08  QCLE/H/STDIO
    2   stdlib.h                         04/19/94 14:18:09  QCLE/H/STDLIB
    3   stddef.h                         04/19/94 14:18:07  QCLE/H/STDDEF
    4   string.h                         04/19/94 14:18:09  QCLE/H/STRING
                                      * * * * *   E N D   O F   I N C L U D E S   * * * * *

                                       * * * * *   M E S S A G E   S U M M A R Y   * * * * *
        Total            Informational(00)         Warning(10)           Error(30)            Severe Error(40)
          0                    0                       0                    0                        0
                               * * * * *   E N D   O F   M E S S A G E   S U M M A R Y   * * * * *
Module CDRIVER was created in library KPSLIB on 06/08/94 at 12:57:06.
                                   * * * * *   E N D   O F   C O M P I L A T I O N   * * * * *
```

Figure E-10 (Part 7 of 7). ILE C Coding for File Transfer Support

Calling File Transfer Support for COBOL/400 Programming Language

Figure E-11 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-3.

```
Program . . . . . . . . . . . . . . . :   COBDRIVER
  Library . . . . . . . . . . . . . . :     KPSLIB
Source file . . . . . . . . . . . . . :   QCBLSRC
  Library . . . . . . . . . . . . . . :     KPSLIB
Source member . . . . . . . . . . . . :   COBDRIVER     05/05/89 09:23:28
Generation severity level . . . . . . :   29
Text 'description' . . . . . . . . . . :   *SAME
Source listing options . . . . . . . :   *NONE
Generation options . . . . . . . . . . :   *NONE
Print file . . . . . . . . . . . . . . :   QSYSPRT
  Library . . . . . . . . . . . . . . :     *LIBL
FIPS flagging . . . . . . . . . . . . :   *NOFIPS *NOSEG *NODEB *NOOBSOLETE
SAA flagging . . . . . . . . . . . . . :   *NOFLAG
Flagging severity . . . . . . . . . . :   0
Replace program . . . . . . . . . . . :   *YES
Target release . . . . . . . . . . . . :   *CURRENT
User profile . . . . . . . . . . . . . :   *USER
Authority . . . . . . . . . . . . . . :   *LIBCRTAUT
Compiler . . . . . . . . . . . . . . . :   IBM COBOL/400

     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
```

Figure E-11 (Part 1 of 5). COBOL/400 Coding for File Transfer Support

```
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-OCCURED       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
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
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
```

Figure E-11 (Part 2 of 5). COBOL/400 Coding for File Transfer Support

```
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
    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
    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
```

Figure E-11 (Part 3 of 5). COBOL/400 Coding for File Transfer Support

```
 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
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
116  018100     SET ERROR-OCCURED 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-OCCURED 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
```

Figure E-11 (Part 4 of 5). COBOL/400 Coding for File Transfer Support

```
      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-OCCURED 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
  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-OCCURED 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-OCCURED 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-OCCURED GO TO ERROR-TERMINATION.                                               11/09/87
      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-OCCURED GO TO ERROR-TERMINATION.                                               11/09/87
      025200*                                                                                            02/02/88
      025300 ERROR-TERMINATION.                                                                          11/09/87
  167 025400     DISPLAY "I-O ERROR OCCURED - PROCESS TERMINATION".                                      11/09/87
  168 025500     STOP RUN.                                                                               11/09/87
                          * * * * *   E N D   O F   S O U R C E   * * * * *

                      * * * * *   E N D   O F   M E S S A G E S   * * * * *
                                        Message Summary
 Total    Info(0-4)    Warning(5-19)    Error(20-29)    Severe(30-39)    Terminal(40-99)
     0         0            0               0                0                0
Source records read . . . . . . . . :   255
Copy records read . . . . . . . . . :   0
Copy members processed  . . . . . . :   0
Sequence errors . . . . . . . . . . :   0
Highest severity message issued . . :   0
 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-11 (Part 5 of 5). COBOL/400 Coding for File Transfer Support

Calling File Transfer Support for RPG/400 Programming Language

Figure E-12 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-3.

```
Compiler . . . . . . . . . . . . . :   IBM RPG/400
Command Options:
  Program  . . . . . . . . . . . . :   KPSLIB/RPGDRIVER
  Source file  . . . . . . . . . . :   KPSLIB/QRPGSRC
  Source member  . . . . . . . . . :   RPGDRIVER
  Source listing options . . . . . :   *SOURCE     *XREF       *GEN        *NODUMP     *NOSECLVL
  Generation options . . . . . . . :   *NOLIST     *NOXREF     *NOATR      *NODUMP     *NOOPTIMIZE
  Source listing indentation . . . :   *NONE
  SAA flagging . . . . . . . . . . :   *NOFLAG
  Generation severity level  . . . :   9
  Print file . . . . . . . . . . . :   *LIBL/QSYSPRT
  Replace program  . . . . . . . . :   *YES
  Target release . . . . . . . . . :   *CURRENT
  User profile . . . . . . . . . . :   *USER
  Authority  . . . . . . . . . . . :   *LIBCRTAUT
  Text . . . . . . . . . . . . . . :   *SRCMBRTXT
  Phase trace  . . . . . . . . . . :   *NO
  Intermediate text dump . . . . . :   *NONE
  Snap dump  . . . . . . . . . . . :   *NONE
  Codelist . . . . . . . . . . . . :   *NONE
  Ignore decimal data error  . . . :   *NO
Actual Program Source:
  Member . . . . . . . . . . . . . :   RPGDRIVER
  File . . . . . . . . . . . . . . :   QRPGSRC
  Library  . . . . . . . . . . . . :   KPSLIB
  Last Change  . . . . . . . . . . :   03/01/89  14:10:21
  Description  . . . . . . . . . . :   *SAME
```

Figure E-12 (Part 1 of 4). RPG/400 Coding for File Transfer Support

```
                        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
     3900  C                     CALL 'QY2FTML' QYLIST                                         11/11/87
     4000  C*                                                                                  11/03/87
     4100  OQSYSPRT H  1     1P                                                                02/23/89

     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
```

Figure E-12 (Part 2 of 4). RPG/400 Coding for File Transfer Support

```
     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.
                         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
        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   * * * * *
```

Figure E-12 (Part 3 of 4). RPG/400 Coding for File Transfer Support

```
                         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 . . . . :   RPG handles blocking function for file. INFDS
            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   * * * * *

                           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-12 (Part 4 of 4). RPG/400 Coding for File Transfer Support

Calling File Transfer Support for a CL Program

Figure E-13 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.

```
Program . . . . . . . . . . . . . . . . . . . . :   SBMRMTJOB
  Library . . . . . . . . . . . . . . . . . . . :     KPSLIB
Source file . . . . . . . . . . . . . . . . . . :   QCLSRC
  Library . . . . . . . . . . . . . . . . . . . :     KPSLIB
Source member name  . . . . . . . . . . . . . . :   SBMRMTJOB   03/21/89 10:15:10
Source printing options . . . . . . . . . . . . :   *SOURCE  *XREF  *GEN  *NOSECLVL
Program generation options  . . . . . . . . . . :   *NOLIST  *NOXREF  *NOPATCH
User profile  . . . . . . . . . . . . . . . . . :   *USER
Program logging . . . . . . . . . . . . . . . . :   *JOB
Allow RTVCLSRC command  . . . . . . . . . . . . :   *YES
Replace program . . . . . . . . . . . . . . . . :   *YES
Target release  . . . . . . . . . . . . . . . . :   *CURRENT
Authority . . . . . . . . . . . . . . . . . . . :   *LIBCRTAUT
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 data base  */                          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 which allows their choice       */
   900-  /* whether or not to submit this member as a batch job.  If a "Y"   */                          01/24/89
  1000-  /* response is given the Submit Data Base Job(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
```

Figure E-13 (Part 1 of 2). CL Coding for File Transfer Support

```
   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.                                          */
   3700-     /***************************************************************/
   3800-              CALL       PGM(QSYS/QY2FTML) PARM(R &FROMLIB +                                          03/21/89
   3900                            &FROMFILE &FROMMBR '      ' QTEMP '          -                           03/21/89
   4000   ' '          ' '      ' Y RCHAS365 &PASSWORD &RTNCODE '        ')                                  03/21/89
   4100-     /***************************************************************/
   4200-     /* Check the FTS return code to insure a good completion.     */
   4300-     /***************************************************************/
   4400-              IF         COND(&RTNCODE *EQ '0') THEN(DO)
   4500-
   4600-     /***************************************************************/                                01/24/89
   4700-     /* Display the member.                                         */                                01/24/89
   4800-     /***************************************************************/                                01/24/89
   4900-              DSPPFM     FILE(QTEMP/&FROMFILE) MBR(&FROMMBR)                                          03/21/89
   5000-                                                                                                    01/24/89
   5100-     /***************************************************************/
   5200-     /* Send the inquiry message to the user.                       */
   5300-     /***************************************************************/
   5400-              SNDUSRMSG  MSG('Submit the job to batch (Y,N)') VALUES(Y +                                01/24/89
   5500                            N) DFT(Y) MSGRPY(&SBMJOB)                                                01/24/89
   5600-                                                                                                    01/24/89
   5700-     /***************************************************************/
   5800-     /* Check the user's response. If it is "Y", submit the job   */                                01/24/89
   5900-     /* to batch using the Submit Data Base Job(SBMDBJOB) command.*/
   6000-     /* If the response is "N", return to caller.                  */                                01/24/89
   6100-     /***************************************************************/
   6200-              IF         COND(&SBMJOB *EQ 'Y') THEN(SBMDBJOB +                                        01/24/89
   6300                            FILE(QTEMP/&FROMFILE) MBR(&FROMMBR))                                     03/21/89
   6400-
   6500-              ENDDO      /* FTS return code is good */
   6600-              ENDPGM
                               * * * * *   E N D   O F   S O U R C E   * * * * *

                                          Cross Reference
 Declared Variables
 Name             Defined     Type           Length      References
 &FROMFILE          1700      *CHAR            10          1500   3800   4900   6200
 &FROMLIB           1600      *CHAR            10          1500   3800
 &FROMMBR           1800      *CHAR            10          1500   3800   4900   6200
 &PASSWORD          1900      *CHAR            10          2700   3800
 &RTNCODE           2000      *CHAR             1          3800   4400
 &SBMJOB            2100      *CHAR             1          5400   6200
* 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.

                                          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-13 (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 through 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* book.

Figure E-14 (Page 1 of 2). File Transfer Messages

FTS Message ID	System Message ID	System Message Description
FTS-1001	CPI7A01	File transfer support started.
FTS-1002	CPI7A02	File transfer support ended.
FTS-1003	CPD7A03	File transfer support ended with remote location because of an error.
FTS-1004	CPD7A04	File transfer support could not be started.
FTS-1005	CPD7A05	File transfer support canceled by system operator.
FTS-1006	CPD7A06	Permanent session error occurred while communicating with remote location.
FTS-1007	CPD7A07	Remote system error.
FTS-1008	CPD7A08	Required number of parameters not specified.
FTS-1009	CPD7A09	User ID not authorized to remote location.
FTS-1010	CPI7A10	Received member from remote location.
FTS-1011	CPI7A11	Member sent to remote location.
FTS-1012	CPD7A12	Error receiving member from remote location.
FTS-1013	CPD7A13	Error sending member to remote location.
FTS-1014	CPF7A14	File transfer support has ended abnormally.
FTS-1019	CPD7A19	Remote location not available.
FTS-1020	CPD7A20	Option code not valid.
FTS-1021	CPD7A21	Database file name not valid.
FTS-1022	CPD7A22	Library name not valid.
FTS-1023	CPD7A23	File data not valid.
FTS-1024	CPD7A24	Member type not valid.
FTS-1025	CPD7A25	Member name not valid.
FTS-1026	CPD7A26	Replace option not valid.
FTS-1027	CPD7A27	Cannot sent System/36 folder.
FTS-1028	CPD7A28	Cannot replace folder.
FTS-1030	CPD7A30	System/36 file already exists.
FTS-1031	CPD7A31	System/36 file or library member in use.
FTS-1032	CPD7A32	Not enough storage to receive member.
FTS-1033	CPD7A33	System/36 VTOC is full. System/36 file cannot be created.
FTS-1034	CPD7A34	User not authorized to access file in library.
FTS-1035	CPD7A35	Error occurred while creating file or member in library.
FTS-1036	CPD7A36	File in library not found.
FTS-1037	CPD7A37	System/36 file cannot be opened.
FTS-1038	CPD7A38	Disk error while opening System/36 file.
FTS-1039	CPD7A39	Cannot send System/36 library.

Figure E-14 (Page 2 of 2). File Transfer Messages

FTS Message ID	System Message ID	System Message Description
FTS-1040	CPD7A40	Cannot send database logical file in library.
FTS-1041	CPD7A41	Disk error while creating System/36 file.
FTS-1042	CPD7A42	Disk error while reading System/36 file.
FTS-1043	CPD7A43	Member in file in library is full.
FTS-1044	CPD7A44	Member record length does not match file definitions record length in library.
FTS-1046	CPD7A46	System/36 dictionary or definition not found in System/36 file.
FTS-1049	CPD7A49	System/36 file exists as a remote file.
FTS-1050	CPD7A50	Library not found.
FTS-1051	CPD7A51	User not authorized to access library.
FTS-1052	CPD7A52	Member already exists.
FTS-1053	CPD7A53	Member not found.
FTS-1054	CPD7A54	Not enough space in library to create member.
FTS-1055	CPD7A55	System/36 directory full. Cannot create member in library.
FTS-1056	CPD7A56	Member is in use.
FTS-1057	CPD7A57	Object in library is an IBM-supplied object.
FTS-1058	CPD7A58	Disk error while opening member.
FTS-1059	CPD7A59	Disk error while closing library member.
FTS-1060	CPD7A60	Mode in network attributes not specified in device for remote location.
FTS-1063	CPD7A63	File in library is not a physical file.
FTS-1064	CPD7A64	Member in file in library record length too long.
FTS-1065	CPD7A65	Cannot receive externally defined file.
FTS-1066	CPD7A66	Required parameters incorrectly specified.
FTS-1067	CPD7A67	Maximum record length too small.
FTS-1068	CPD7A68	TYPE parameter value does not match attributes of file in library.
FTS-1069	CPD7A69	Transparent mode required.
FTS-1070	CPD7A70	Error occurred while attempting to establish session with remote location.
FTS-1071	CPD7A71	TYPE parameter value not valid for file.
FTS-1072	CPD7A72	File wait time not valid.
FTS-1073	CPD7A73	Data compression option not valid.

Bibliography

The following books contain information you may need. The books are listed with their full title and order number. Except where otherwise indicated, each is an AS/400 system book.

Communications Books

- *APPN Support*, SC41-5407, provides the information necessary to define and use the AS/400 system Advanced Peer-to-Peer Networking (APPN) function. This book contains information on both ICF and Common Programming Interface (CPI) Communications.
- *APPC Programming*, SC41-5443, provides the application programmer with information about the advanced program-to-program communications (APPC) support provided by the AS/400 system. This book is a guide for developing application programs and for defining the communications environment for APPC communications. This book contains information on both ICF and Common Programming Interface (CPI) Communications.
- *DSNX Support*, SC41-5409, contains information about preparing a system for remote management activities and about using the change management feature distributed systems node executive (DSNX).
- *Asynchronous Communications Programming*, SC41-5444, provides the application programmer with information for creating an asynchronous communications definition, writing programs that use asynchronous communications, and responding to return codes. It also provides information on developing asynchronous communications application programs that use ICF.
- *BSC Equivalence Link Programming*, SC41-5445, provides the application programmer or programmer with the information needed to write programs that use binary synchronous communications equivalence link (BSCEL) to communicate with a remote system. It also contains information for programmers about other systems and devices that communicate with BSCEL on the AS/400 system. The book describes how to set up BSCEL and how to run application programs that use BSCEL.
- *Finance Communications Programming*, SC41-5449, provides information for the application programmer or system administrator who uses the OS/400 finance support for financial communications between devices at multiple locations. It describes how finance support communicates with a controller and how to set up finance support. This book provides information for writing application programs to communicate with application programs on the finance controller using ICF.
- *Intrasystem Communications Programming*, SC41-5447, provides the application programmer or programmer with information about interactive communications between two application programs on the same AS/400 system. This book also provides information on developing intrasystem communications application programs that use ICF.
- *Communications Management*, SC41-5406, provides the system operator or programmer with information on using AS/400 communications such as work management, communications status, error handling, aggregate line speed and subsystem storage.
- *Communications Configuration*, SC41-5401, contains information for configuring objects for communications.
- *Remote Job Entry (RJE) Guide*, SC09-1903, provides information for using the Communications Utilities remote job entry (RJE) to submit jobs to an IBM host processor.
- *Retail Communications Programming*, SC41-5448, provides information for setting up and starting retail communications between devices at multiple locations.
- *SNA Upline Facility Programming*, SC41-5446, contains information on using the Systems Network Architecture Upline Facility with the IBM AS/400 system.
- *Sockets Programming*, SC41-5422, provides information for using the sockets programming interface for the AS/400 system.
- *3270 Device Emulation Support*, SC41-5408, provides information on setting up and starting 3270 device emulation using binary synchronous communications (BSC) or Systems Network Architecture (SNA) communications.
- *Application Display Programming*, SC41-5715, provides information about:
  - – Using DDS to create and maintain displays for applications;
  - – Creating and working with display files on the system;
  - – Creating online help information;
  - – Using UIM to define panels and dialogs for an application;
  - – Using panel groups, records, or documents
- *System API Reference*, SC41-5801, provides user-defined communications information for the AS/400 system.
- *TCP/IP Configuration and Reference*, SC41-5420, provides information on TCP/IP configurations, IBM-supplied applications, and user-written applications.

Programming Language Books

- *DDS Reference*, SC41-5712, contains information about coding data description specifications for physical, logical, display, printer, and ICF files.
- *Data Management*, SC41-5710, provides information to help the system programmer manage key aspects of the system. For example, it describes how to use diskette and tape files.

- *DB2 for AS/400 Database Programming*, SC41-5701, provides information about the AS/400 database management system, and describes how to set up and use a database on the AS/400 system.
- *Distributed Data Management*, SC41-5307, 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.
- *Printer Device Programming*, SC41-5713, provides information to help the programmer manage key aspects of the system. For example, it describes how to use printer files.
- *ILE COBOL/400 Reference*, SC09-2073, provides a description of the ILE COBOL language organization, program organization, procedure division statements, and compiler directing statements.
- *ILE COBOL/400 Programmer's Guide*, SC09-2072, Provides information about how to write, compile, bind, run, debug, and maintain ILE COBOL/400 programs on the AS/400 system.
- *COBOL/400 User's Guide*, SC09-1812, provides the information needed to write, test and maintain COBOL/400 programs for the AS/400 system.
- *ILE RPG/400 Reference*, SC09-2077, provides information about the ILE RPG programming language. This manual describes, position by position and keyword by keyword, the valid entries for all RPG IV* specifications, and provides a detailed description of all the operation codes and built-in functions.
- *ILE RPG/400 Programmer's Guide*, SC09-2074, provides information about the ILE RPG programming language, including information on creating and running programs.
- *RPG/400 User's Guide*, SC09-1816, provides the information needed to use the RPG/400 programming language to code programs for the AS/400 system.
- *ILE C/400 Programmer's Reference*, SC09-2070, provides information about how to write programs that adhere to the Systems Application Architecture C Level 2 definition and use ILE C/400 specific functions such as record I/O.
- *ILE C/400 Programmer's Guide*, SC09-2069, provides the information needed to use the ILE C programming language to code programs for the AS/400 system.
- *CL Programming*, SC41-5721, provides a wide-ranging discussion of AS/400 system programming topics.
- *CL Reference*, SC41-5722, provides information on control language commands.
- *ILE Concepts*, SC41-5606, explains concepts and terminology pertaining to the Integrated Language Environment (ILE) architecture of the OS/400 licensed program. Topics covered include creating modules, binding, running programs, debugging programs, and handling exceptions.
- *Work Management*, SC41-5306, contains information on how to create and change a work management environment.
- *Security – Reference*, SC41-5302, contains information about general security concepts for the system.
- *System Operation*, SC41-4203, provides information for the system operator on how to use the system unit control panel.

System/36-Related Books

- *System/36 System Reference*, SC21-9020, contains information on System/36 naming conventions you might need for file transfer support (FTS).
- *Using S/36 Communications*, SC21-9082, contains information on the correlation between system message IDs and File Transfer Support message IDs on the System/36.

Index

Reader Comments—We'd Like to Hear from You!

AS/400
ICF Programming
Version 4

Publication No. SC41-5442-00

Overall, how would you rate this manual?

	Very Satisfied	Satisfied	Dissatis-fied	Very Dissatis-fied
Overall satisfaction				

How satisfied are you that the information in this manual is:

Accurate				
Complete				
Easy to find				
Easy to understand				
Well organized				
Applicable to your tasks				
THANK YOU!				

Please tell us how we can improve this manual:

____

____

____

____

____

May we contact you to discuss your responses? __ Yes __ No
Phone: (____) ___________ Fax: (____) ___________ Internet: ___________

To return this form:

- Mail it
- Fax it
  United States and Canada: **800+937-3430**
  Other countries: **(+1)+507+253-5192**
- Hand it to your IBM representative.

Note that IBM may use or distribute the responses to this form without obligation.

______________________________ ______________________________
Name Address

______________________________ ______________________________
Company or Organization

______________________________ ______________________________
Phone No.

Reader Comments—We'd Like to Hear from You!
SC41-5442-00

Cut or Fold
Along Line

Fold and Tape **Please do not staple** Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

ATTN DEPT 542 IDCLERK
IBM CORPORATION
3605 HWY 52 N
ROCHESTER MN 55901-9986

Fold and Tape **Please do not staple** Fold and Tape

Cut or Fold
Along Line

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC41-5442-00

Spine information:

IBM AS/400 ICF Programming *Version 4*