AS/400

Intrasystem Communications Programming

Version 4

SC41-5447-00

AS/400

Intrasystem Communications Programming

Version 4

SC41-5447-00

Take Note!

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

August Edition (August 1997)

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

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

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

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

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

or you can fax your comments to:

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

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

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

Contents

Figures

Notices

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

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

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

Address your questions to:

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

This publication could contain technical inaccuracies or typographical errors.

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

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

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

Programming Interface Information

This book is intended to help the customer develop communications between two application programs on the same system. It contains information about the intrasystem communications support provided by the AS/400 system. The *Intrasystem Communications Programming* book 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:

Application System/400
AS/400
COBOL/400
IBM
ILE C/400
ILE COBOL/400
ILE RPG/400
RPG/400
Operating System/400
OS/400
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 Intrasystem Communications Programming (SC41-5447)

This book contains information about the intrasystem communications support provided by the AS/400 system. It is intended to be used as a guide for developing communications between two application programs on the same system.

For a list of related publications, see the Bibliography.

Who Should Use This Book

This book is intended for the AS/400 application programmer responsible for defining or using intrasystem communications support. It is used for developing application programs that use the support.

You should be familiar with the following information:

- Managing jobs and messages on the AS/400 system, described in the *System Operation* book, SC41-4203.
- Using the intersystem communications function (ICF) file, described in the *ICF Programming* book, SC41-5442.
- Communications configuration information described in the *Communications Configuration* book, SC41-5401.

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 Intrasystem Communications

AS/400 **intrasystem communications** allows two application programs, which are running in two different jobs on the same system, to communicate with each other through an ICF file. Using intrasystem communications can help you debug the programs before they are used to communicate with a remote system over a communications line. AS/400 application programs can be written in the ILE C/400*, ILE COBOL/400, or ILE RPG/400* programming languages to use intrasystem communications.

The intrasystem communications support uses **intersystem communications function (ICF)**[1] data management to handle the sending and receiving of data between the two programs. For communications to begin between programs, the intrasystem communications device description first needs to be configured and varied on.

Note: Because intrasystem communications supports process-to-process communications within the same system without the use of communications lines, line and controller descriptions are not used.

Overview of Intrasystem Communications

Figure 1-1 provides an overview of the Operating System/400* (OS/400*) intrasystem communications support. Application program A communicates with application program B. ICF data management handles the communications functions and data from your program. The intrasystem communications support handles the communications protocol needed for data transfer and communications between the two programs.

Both the source program (Program A) and the program with which it is communicating (Program B) must use the same device description.

Figure 1-2 on page 1-2 shows how multiple target programs can communicate with the same source program.

When using intrasystem communications, a source program can acquire more than one session for a given device description, and can issue more than one evoke function to start multiple target programs. This means, for example, that PGMA can establish a transaction with PGMB on one session and another transaction with PGMC on another session, and have all the transactions at the same time. However, having established a communications transaction with PGMB on a given session, PGMA cannot then establish a transaction with PGMC on the *same* session.

Note: The term **target program** is used in this book to refer to the program with which the source program communicates, even though the target program is not on a remote system.

Intrasystem communications imposes no restrictions as to the maximum number of sessions that can be associated with a device. However, the maximum program device (MAXPGMDEV) parameter on the Create ICF File (CRTICFF) command specifies the maximum number of program devices that you can use with the ICF file.

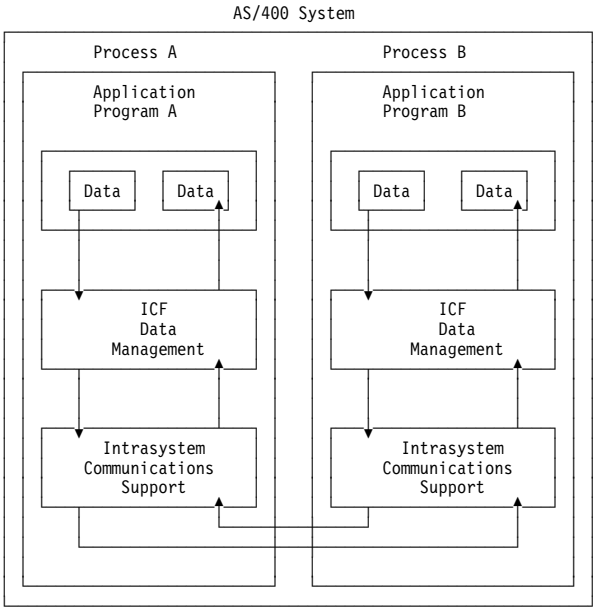


Figure 1-1. Overview of Intrasystem Communications

[1] The intersystem communications function (ICF) is a function of the operating system that allows a program to communicate interactively with another program or system.

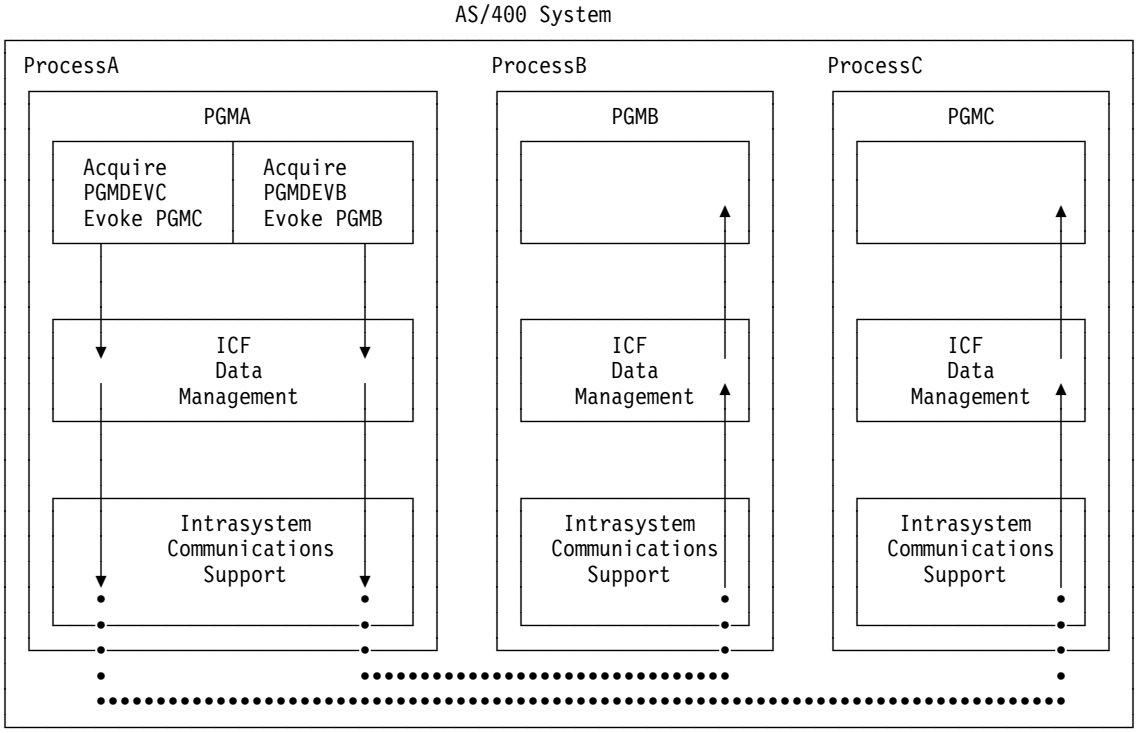


Figure 1-2. A Source Program Communicating with Multiple Target Programs

Using Intrasystem Communications to Test Communications Applications

Intrasystem communications can be used to test new communications programs to be run using other communications types. Using intrasystem communications can help you debug the programs before they are used to communicate with a remote system over a communications line.

During testing, you can check only those return codes returned by intrasystem communications and issue only those operations supported by intrasystem communications.

It is also important to note that there are differences in the way intrasystem communications supports certain read and write operations and the way other communications types support such operations. For example, intrasystem communications implicitly responds to a confirm request in the AS/400 system environment, whereas advanced program-to-program communications (APPC) does not. For further details about using intrasystem communications to test other communications types, refer to Appendix C.

Chapter 2. Configuring Intrasystem Communications

This chapter describes the commands used for configuring intrasystem communications on your system.

When using intrasystem communications configuration commands, you can enter the commands in one of two ways:

- Using the command prompt. Enter the command and press F4 (Prompt). A prompt menu is shown for the command.
- Using direct entry. Enter the command and its parameters following the syntax described in the *CL Reference* book.

In this chapter, the parameters of the CL commands that apply to intrasystem communications are described.

Defining the Intrasystem Communications Configuration

A configuration for intrasystem communications consists of an intrasystem communications device description. The device description describes the characteristics of the logical connection between the two programs. Because programs are communicating with each other on the same system, line and controller descriptions are not supported. To use the intrasystem communications device, it must first be configured and varied on. You can create or change an intrasystem communications device description using the following commands:

- Create Device Description (Intrasystem) (CRTDEVINTR) command
- Change Device Description (Intrasystem) (CHGDEVINTR) command

The parameters for the CRTDEVINTR and CHGDEVINTR commands are:

DEVD
Specifies the name for the device description.

RMTLOCNAME
Specifies the remote location name with which your program communicates. This parameter cannot be specified on the CHGDEVINTR command.

ONLINE
Specifies if this device should be automatically varied on during an initial program load (IPL).

***YES:** This device is varied on automatically at IPL.

***NO:** This device is not varied on automatically at IPL. This is the default value.

AUT
Specifies the authority you are granting users who do not have specific authority to the object, are not on the authorization list, or whose group has no specific authority to the object.

***LIBCRTAUT:** The system determines the authority for the object by using the value specified on the CRTAUT parameter on the CRTLIB command for the library containing the object to be created. If the value specified on the CRTAUT parameter is changed, the new value will not affect any existing objects. This is the default value.

***CHANGE:** Change authority allows the user to perform all operations on the object except those limited to the owner or controlled by object existence authority and object management authority. The user can change the object and perform basic functions on the object. Change authority provides object operational authority and all data authority.

***ALL:** All authority allows the user to perform all operations on the object except those limited to the owner or controlled by authorization list management authority. The user can control the object's existence, specify the security for the object, change the object, and perform basic functions on the object. The user cannot transfer ownership of the object.

***USE:** Use authority allows the user to perform basic operations on the object, such as displaying the object. The user is prevented from changing the object. Use authority provides object operational authority and read authority.

***EXCLUDE:** Exclude authority prevents other users from accessing the object.

TEXT
Specifies text that briefly describes the object.

***BLANK:** No text is specified. This is the default value.

'description': Specify no more than 50 characters, enclosed in apostrophes, provided you do not use the prompt screen.

Example

```
CRTDEVINTR  DEVD(INTRALOC)
  RMTLOCNAME(INTRARMT)
  ONLINE(*YES) AUT(*CHANGE)
  TEXT('This is an intrasystem
  device description')
```

This command creates a device named INTRALOC and a remote location name INTRARMT, allowing two programs to communicate within the same system.

Chapter 3. Running Intrasystem Communications Support

This chapter contains the information you need to run the intrasystem communications support.

Vary On and Vary Off Support

Once an intrasystem communications device has been configured, you can use the Vary Configuration (VRYCFG) command to activate and deactivate the device configuration. This can also be done from the WRKCFGSTS display.

Use the VRYCFG command and specify CFGTYPE(*DEV) and STATUS(*ON) to vary on the configured device description.

Use the VRYCFG command and specify CFGTYPE(*DEV) and STATUS(*OFF) to vary off the configured device description.

The following parameters are applicable to intrasystem communications:

CFGOBJ
Specifies the name of the description for the device to be varied on or off.

CFGTYPE
Specifies the type of configuration description to be varied on or off. This is a required parameter. The only valid entry for intrasystem communications is:

***DEV:** Device configuration

STATUS
Specifies the status of the configuration object.

***ON:** The object is varied on.

***OFF:** The object is varied off.

RANGE
Specifies what configuration elements should be varied, either the configuration element specified (*OBJ) or the configuration element specified and its attached configuration elements (*NET). Devices are considered not to have attached configuration elements. For devices, you can specify either RANGE(*OBJ) or RANGE(*NET).

VRYWAIT
Specifies whether the Ethernet, token-ring, X.25, or switched SDLC, BSC, or asynchronous line description is varied on asynchronously or synchronously. Specify how long the system waits for vary on to be completed (for synchronous vary on) after which the communications file is opened and the session is acquired.

If the VRYWAIT parameter is specified on the VRYCFG command for a line description that is not Ethernet, token-ring, X.25, or switched SDLC, BSC, or asynchronous, the parameter is accepted but ignored.

***CFGOBJ:** The VRYWAIT parameter value specified in the line description is used.

***NOWAIT:** The system does not wait for vary on completion. The line is varied on asynchronously.

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

ASCVRYOFF
Specifies whether the vary off is asynchronous. This parameter is not allowed when STATUS(*ON) is specified.

***NO:** The vary off is synchronous.

***YES:** The vary off is asynchronous.

Example

```
VRYCFG CFGOBJ(INTRALOC) CFGTYPE(*DEV)
  STATUS(*ON) RANGE(*OBJ)
```

This command varies on the configured device description INTRALOC.

Chapter 4. Writing Intrasystem Application Programs

This chapter describes how an application program uses the intersystem communications function (ICF) file and the intrasystem communications support. The program can be coded using the ILE C/400, C Set ++, ILE COBOL/400, or ILE RPG/400 programming languages, which allows the program to do the following functions:

- Start a session by opening an ICF file and acquiring a program device.
- Send and receive information by writing or reading to an ICF file.
- End a session by releasing the program device and closing the ICF file.

This chapter also includes a description of the read and write operations that specify a record format containing specific communications functions. Record formats can be defined using data description specifications (DDS), or you may use system-supplied formats.

After an operation is completed, a return code (and a high-level language file status) is returned to your application. The return code indicates whether the operation was completed successfully or unsuccessfully. Along with the return code, exception messages may also be issued. Refer to Appendix B for more information about return codes and to the appropriate language reference books for more information about the high-level language file status.

Intersystem Communications Function File

An intersystem communications function (ICF) file must be created before your application can use the intrasystem communications support. The ICF file is used to describe how data is presented to the program with which your program is communicating, and how data is received from that program. If you are using DDS keywords, use the Create Intersystem Communications Function File (CRTICFF) command to create an ICF file. If you are using the system-supplied formats (such as $$SEND), you do not need to create an ICF file. The ICF file QICDMF, which is in the library QSYS, is supplied by IBM for communications.

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

The following commands are used to manage the ICF file, and are described in detail in the *ICF Programming* book.

CRTICFF Create ICF File. This command allows you to create an ICF file and file level attributes.

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

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

DLTF Delete File. This command allows you to delete a file from the system.

DSPFD Display File Description. This command displays the file description of any file on the system. The information may be printed or displayed.

DSPFFD Display File Field Description. This command displays the description of the fields in any file on the system. This information may be printed or displayed.

ADDICFDEVE Add ICF Device Entry. This command allows you to permanently add a program device entry that contains a program device name, remote location information, and session-level attributes to an ICF file.

CHGICFDEVE Change ICF Device Entry. This command allows you to permanently change the program device attributes previously added with the ADDICFDEVE command.

OVRICFDEVE Override ICF Device Entry. This command allows you to do the following:

- Temporarily add the program device entry, the remote location information, and the session-level attributes to the ICF file.
- Temporarily change a program device entry with the specified remote location information and session-level attributes for an ICF file. These changes are only in effect for the job.

RMVICFDEVE Remove ICF Device Entry. This command allows you to permanently remove the program device entry previously added to an ICF file with the ADDICFDEVE command or changed with the CHGICFDEVE command.

Specifying the Program Device Entry Commands

The following describes the parameters for the ADDICFDEVE, CHGICFDEVE, and OVRICFDEVE commands and lists the valid values for each parameter for intrasystem communications.

FILE

Specifies the name and library of the ICF file to which you are adding or changing the program device entry. The FILE parameter is not available on the OVRICFDEVE command.

***LIBL**: Intrasystem communications support uses the library list to locate the ICF file.

***CURLIB**: Intrasystem communications support uses the current library for the job to locate the ICF file. If no current library entry exists in the library list, intrasystem communications uses QGPL.

filename: A 1- to 10-character value that specifies the name of the ICF file.

library-name: A 1- to 10-character value that specifies the library where the ICF file is located.

PGMDEV

Specifies the program device name that is defined in the ICF file and specified in the application. The total number of devices that can be acquired to an ICF file is determined by the MAXPGMDEV parameter on the CRTICFF or CHGICFF command.

pgm-device-name: Enter a 1- to 10- character value for the program device name being defined. This name is used on device-specific input and output operations to identify the program device and the attributes.

RMTLOCNAME

Specifies the remote location name with which your program communicates. A remote location name must be specified on the ADDICFDEVE command or an OVRICFDEVE command. If a remote location name is not specified, a major and minor error code are returned when the program device is acquired.

***REQUESTER**: The name used to refer to the communications device through which the program was started. The session that is assigned when the program device is acquired is the same session that receives the program start request. If the program is not started as a result of a program start request, the acquire operation for the program device fails. The target program always uses *REQUESTER as the remote location name in the ICF file to connect to the session that the source program uses to send the program start request.

remote-location-name: Enter a 1- to 8-character name for the remote location name that should be associated with the program device.

FMTSLT

Specifies the type of record format selection used for input operations for all devices.

***PGM**: The program determines what record formats are selected. If an input (read) operation with a record format name is specified, that format is always selected. If an input operation without a record format is specified, the default format (the first record format in the file) is always selected. This also means that if any record identification (RECID) keywords are specified in the data description specifications (DDS) for the file, they are not taken into consideration when the record is selected.

***RECID**: The RECID keywords specified in DDS for the file are used to specify record selection. If no RECID keywords are specified in the file, an error message is sent and an acquire operation for the program device will fail.

***RMTFMT:** The remote format names received from the sending program are used to select the record format.

CMNTYPE

Identifies the communications type for which you define a program device entry. You should specify the value *INTRA or *ALL for this parameter.

***INTRA**: The prompt for all intrasystem communications-supported attributes.

Note: When you specify *REQUESTER for the remote location name (RMTLOCNAME), you are only prompted for the attributes of the Format Select parameter (FMTSLT) and the Secure from Override parameter (SECURE).

BATCH

Specifies if batch processing is performed for the session. If you specify RMTLOCNAME(*REQUESTER), this parameter is ignored. The program that issues the evoke function determines whether batch processing can occur.

***NO:** Specifies that batch processing does not occur.

***YES:** Specifies that batch processing occurs.

Note: Function-management-header, cancel, and negative-response functions are only valid if BATCH(*YES) is specified.

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.

***NO:** Specifies no protection from other program device overrides.

***YES:** Specifies program device override protection from override commands started in earlier programs.

Communications Operations

This section provides a description of the operations you can code into a program that uses intrasystem communications support to communicate with another program.

Starting a Session

A communications session is a logical connection by which a program running in one job can communicate with another program running in a different job. A communications session is established with an acquire operation, and is ended with a release operation or end-of-session function.

Open/Acquire Operation

Your application program uses the acquire operation to establish the session on which your program will communicate with another program. Intrasystem communications uses the value of each parameter that was specified on the ADDICFDEVE or OVRICFDEVE command for the program device.

The parameters are used to determine the following session characteristics:

Format selection option: This indicates the type of processing that needs to be done to determine what record format to use on an input operation. Intrasystem communications supports all three format selection options: *PGM, *RECID, and *RMTFMT.

Batch option: This indicates whether or not batch processing is performed for the session. The following functions are only valid if BATCH(*YES) is specified on the ADDICFDEVE or OVRICFDEVE command:

- Function-management-header function
- Cancel function
- Negative-response function

Starting a Transaction

A **transaction** is a logical connection between two programs. Use an evoke function to start a transaction between your program and another program.

Evoke Function

Your program uses the evoke function to start a transaction with a target program after you start a session. It is not valid if your program is already communicating with another program on the same session.

The program that issues an evoke function (the source program) is initially in send state unless, for example, it also issues a read operation, an allow-write function, or an invite function to the other program. The source program would then be in receive state and the other program can send data.

The program with which the source program communicates (the target program) is initially in receive state, and should issue read operations until a receive-turnaround indication is received.

The program that is in send state controls the transaction, and determines what the other program must do. For example, if a program that is in send state sends data, then the program in receive state should issue a read operation to receive the data (it cannot also send data).

With the evoke function your program can specify the following information:

- The name of the program with which your program is to communicate
- The library in which the other program exists (optional)
- User-defined program initialization parameters (optional)
- Synchronization level (optional)
- Security information (optional)

If your program is using the EVOKE DDS keyword, you can specify all of the above information. If your program is using one of the evoke system-supplied formats, you can specify all of the above except for synchronization level. In this case, synchronization or confirmation is not allowed.

If you specify **program initialization parameters (PIP)**[1] 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 that are not predictable may occur.

For information on how to code the evoke function, refer to the *ICF Programming* book and the *DDS Reference* book.

[1] The program initialization parameter (PIP) is the initial parameter value passed to a target program as input or used to set up the process environment.

Sending Data

You can send data during a transaction using the write operation. With the write operation, you can, for example, specify the end of a group of records, or indicate when your program has finished sending data.

Write Operation

The write operation is used to send data records to another program. Each write operation sends only one data record. Intrasystem communications supports various functions that are discussed following this description. These functions may be issued by your program to another program either with or without data. The only exception is the function-management-header function, which requires data to be sent.

Only one write operation can be issued at a time. Therefore, if your program issues a second write operation while the first one has not been received by the other program, the second write operation will not complete until the data previously sent is read by the other program.

Force-Data Function

Your program uses the force-data function to send communications data currently held in the output buffer. However, because intrasystem communications does not buffer data, the force-data function does not provide any additional function.

Confirm Function

Your program uses the confirm function to indicate to the other program that a response is needed before an operation can complete. Once a confirm function is issued, control does not return to your program until the other program with which you are communicating responds to the confirm.

Note: The confirm function is allowed only if the transaction was started with a synchronization level of confirm on the evoke function.

Format-Name Function

Your program uses the format-name function to send the record format name, along with data, to the other program. This is the record format name that should be used by the other program on the corresponding read operation.

Note: The value *RMTFMT for the FMTSLT parameter needs to be specified by the other program on the ADDICFDEVE or OVRICFDEVE command for the record format name to be used.

Subdevice Selection Function

Your program uses the subdevice selection function to specify to the other program the device to which the output is to be directed. Intrasystem communications sends the subdevice code as a separate record before sending the data.

Note: The subdevice function is ignored on any write operation except for the first operation in a group.

End-of-Group Function

Your program uses the end-of-group function to indicate to the other program that this is the last record in a group. It does not indicate, however, that your program is ready to receive data. The session remains in a send state.

Function-Management-Header Function

Your program uses the function-management-header function to indicate to the other program that function-management-header data is being sent. Function-management-header data contains control information for the data that is to follow, and is only valid under the following conditions:

- If the program is running in batch mode (BATCH(*YES) was specified on the ADDICFDEVE or OVRICFDEVE command)
- If data is being sent (the data length is greater than zero)

Intrasystem communications inserts the characters FMH before the data being sent.

Receiving Data

Intrasystem communications supports various functions designed to obtain data from the other program.

Read Operation

Your application program uses the read operation to obtain data or control information from the other program. This operation causes the user to wait for the data if it is not immediately available. You can use the read operation by itself or in combination with other write operations, in which case the write operation is performed first, followed by the read operation.

Invite Function

Your application program uses the invite function to request input data from another program, but it receives control without waiting for the input. To obtain the data, you must issue either a read or a read-from-invited-program-devices operation. This function can be issued by itself or in combination with other write operations.

Read-from-Invited-Program-Devices Operation

You can use the read-from-invited-program-devices operation to obtain data from any program that has responded to an invite function previously issued in your program. If data becomes available to your program from more than one program device before the read-from-invited-program-devices operation is issued, your program receives the data that was *first* made available.

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

Use data queues when a program must wait for a display file, an ICF file, and a data queue, in any combination, at the same time. The following commands are used with the specified DTAQ parameter:

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

Use these commands to indicate a data queue that will have entries placed in it when one of the following occurs:

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

For more information, see the *CL Programming* book and the *ICF Programming* book.

Notifying the Remote Program of Problems

Your program uses the fail, cancel, and negative-response functions to indicate that an error has occurred during a transaction with the target program.

Fail Function

Your program issues the fail function to indicate that it has detected an error in the data while it was sending or receiving data. The fail function can be sent in either send or receive state. No data can be sent with the fail function.

If a program that is in the send state issues a fail function, either the data just sent was in error or some other condition occurred. Intrasystem communications support informs the other program of the error by returning a 0302 return code. The last record before the fail function was issued is still sent to the other program.

If a program is in the receive state and issues a fail function, intrasystem communications support discards the incoming data, informs the other program that a fail is being sent by returning a 0402 return code, and changes the state of your program's session from receive to send state.

In either case, the program that issued a fail function should send, and the program that received the fail must receive.

If both programs issue a fail function at the same time, the program that was receiving will be successful and should send. The program that was sending receives a fail return code.

Cancel Function

Your application program can issue a cancel function to indicate that it detected an error in the data it was sending. The cancel function is only valid under the following conditions:

- If your program is running in batch mode (BATCH(*YES) was specified on the ADDICFDEVE or OVRICFDEVE command)
- Within a group of records
- When the program is in send state

When your program is sending data and issues a cancel function, intrasystem communications support informs the other program that a cancel is being sent. No data may be sent with a cancel operation.

The program that issues a cancel function should send, and the program that receives the cancel must receive.

Issuing a cancel function is similar to issuing a fail function when your program is sending data.

Negative-Response Function

Your application program can use a negative-response function to indicate that it detected an error in the data it was receiving. The negative-response function is only valid under the following conditions:

- If your program is running in batch mode (BATCH(*YES) was specified on the ADDICFDEVE or OVRICFDEVE command)
- Within a group of records, or as the first function after receiving an end-of-group function
- When the program is in receive state, but an invite function has not yet been issued or is currently not in effect

When your program issues a negative-response function, intrasystem communications support discards any data being received, and informs the other program that a negative-response is being sent.

Your program can also send eight bytes of **sense data** with the negative-response function to inform the other program about the reason for the error. Intrasystem communications checks this data to ensure that the first four bytes are 10xx, 08xx, or 0000, where x is a digit. If not, the function is rejected, with a return code of 831B. If your program does not supply sense data, then intrasystem communications sends the code 08110000.

Issuing a negative-response function is similar to issuing a fail function when your program is receiving data.

Using Additional Functions/Operations

Intrasystem communications supports the following additional functions or operations.

Respond-to-Confirm Function

Your program uses the respond-to-confirm function to send a positive response to a received confirm request. The positive response indicates that data was received without error or that the request received may be performed (such as a detach).

You can issue the respond-to-confirm function only after receiving a confirm request from the other program.

A respond-to-confirm function is not required, however, and an implicit positive response is sent if the next operation is not a fail, cancel, negative-response, or end-of-session function.

Request-to-Write Function

Your program uses the request-to-write function to indicate that it wants to send something to the other program rather than continue receiving data. The other program decides, however, whether to stop sending data and when it will stop.

After issuing a request-to-write function, your program must continue to receive data until it receives a return code that indicates the other program is ready to begin receiving (if it decides to do so). Your program, in response to the return code, can then begin to send its data, perform other processing, or end.

Your program can issue the request-to-write function only when no invite function is in effect, and only when your program is in the receive state.

Allow-Write Function

Your program issues the allow-write function to inform the program with which it is communicating that it is finished sending data and is ready to receive.

Intrasystem communications sends data and an indication to the other program that allow-write is being sent. If the operation is successful, a return code of 0001 is returned to indicate that your program is ready to receive data.

Cancel-Invite Function

Your program uses the cancel-invite function to attempt to cancel an outstanding invite function for which no data has been received. Cancel-invite is only valid when an invite function is still in effect.

When your program issues a cancel-invite function, intrasystem communications determines if data has been received from the other program. If no data has been received, the invite is canceled, and your program is changed from an invite state to send state.

If data has already been received from the other program, the invite is not canceled, and a return code of 0412 is returned. Your program must then issue a read or read-from-invited-program-devices operation to receive the data that the other program has already sent.

Timer Function

Your program can use the timer function to set the maximum amount of time your program waits to receive data when issuing the read-from-invited-program-devices operation.

Get-Attributes Operation

Your program uses the get-attributes operation to determine the status of the session. It can be issued at any time during a session. The operation gets the current status information about the session to which your program is communicating.

Ending Transactions

The detach function is used to end an active transaction between your program and the program with which it is communicating.

Detach Function

Your program uses the detach function to inform the other program that your program is finished sending data and wants to end the transaction.

Intrasystem communications sends the data and indicates to the other program that the current record is the last record.

When a detach function is issued with a confirm function, the transaction is ended by your program if a positive response is received, and no further input or output operations with the other program is allowed. When a detach function is issued without a confirm function, the transaction ends without waiting for a response from the other program. When the target program receives the detach, it can no longer communicate with the source program and must end the logical con-

nection to the session by ending the session. A source program must issue an evoke function to establish communications again with a target program after sending or receiving a detach function.

When a detach function is issued by a target program, its logical connection to the session, as well as to the transaction, is ended.

Ending Sessions

The following function and operations can be used by your program to end a session.

Release Operation

Your program uses the release operation to attempt to end the program's attachment to a session. Depending on how the session was started, the release operation produces different results:

- If the session is associated with the source program, the release operation ends the session immediately (unless some error condition occurs). The operation frees the resources (allocated to the program) used during the session. If the release operation is not successful, the end-of-session function can be issued to end the session. The release operation is only valid when a transaction is not active.
- If the session is associated with the target program, the release operation only temporarily ends the connection to the source program. The session is kept active, and is not available for use by other programs until the target program issues an end-of-session function or ends. If a detach has not been done (that is, the transaction is still active), an acquire can be issued to continue communications on that session.

End-of-Session Function

Your program uses the end-of-session function to end a session with another program. Unlike the release operation, the end-of-session function always ends the session. However, if the function is issued during an active transaction, intrasystem communications abnormally ends the session.

When your program issues an end-of-session function, intrasystem communications ends the program's attachment to the session and frees the resources in the AS/400 system used during the session. The resources are made available to other programs in the AS/400 system that want to establish a session.

Close Operation

Your program uses the close operation to close the ICF file and to end the program's attachment to any active session the program has acquired. If the close operation is issued to a session that was established by a source program, intrasystem communications ends the session and deallocates all resources that were allocated for the file. If a transaction is active when the close operation is issued, both the session and the transaction are abnormally ended.

If the close operation is issued to a session associated with a target program, the connection to the program is only temporarily ended. The session is kept active and is not available for use by other programs until the target program issues an end-of-session function or ends.

Using Response Indicators

Response indicators[2] are defined to your program in the ICF file and are set on each input operation. However, these indicators are optional and major and minor return codes can also be used to indicate the status of input operations.

Receive-Confirm

Your program uses the receive-confirm response indicator to receive an indication from the other program that the record it received contained a confirm request. A received confirm request indicates the other program is expecting your program to perform a specific action to synchronize the programs. This action can be a respond-to-confirm function to respond positively or a fail or end-of-session function to respond negatively. Your program can also do a normal input/output operation to respond positively.

The presence of the confirm request is also indicated by the minor return codes 14, 15, 17, 1C, 44, 45, and 47 with the major return code 00 (user data received) or 02 (user data received but program is being ended), or by the minor return codes 14, 15, 17, and 1C with the major return code 03 (no data received).

Receive-End-of-Group

The receive-end-of-group response indicator is used to indicate that the other program has sent the last record in a group.

The presence of the end-of-group function is also indicated by the minor return codes 03, 07, 17, and 47 with the major return code 00 (user data received) or 02 (user data received

[2] A response indicator is a 1-character field passed with an input record from the system to a program to provide information about the data record.

but program is being ended), or by the minor return codes 03 and 17 with the major return code 03 (no data received).

Receive-Function-Management-Header

Your program uses the receive-function-management-header response indicator to receive an indication from the other program that function-management-header data was received. The first three characters of the received data are the characters FMH.

The presence of function-management-header data is also indicated by the minor return codes 04, 05, 07, 44, 45, and 47 with the major return code 00 (user data received) or 02 (user data received but program is being ended).

Receive-Fail

Your program uses the receive-fail response indicator to receive an indication that the other program encountered an error when it was sending or receiving data, and your program should take the appropriate recovery action. Your program remains in receive state after receiving the receive-fail indicator and should continue to issue read operations.

Receipt of a fail request is also indicated by the minor return code 02 with the major return code 03 (no data received) or 04 (output exception occurred).

The failure notification is always received without user data.

Receive-Cancel

Your program uses the receive-cancel response indicator to receive an indication that the other program encountered an error when it was sending data.

Receipt of a cancel request is also indicated by the minor return codes 30 and 31 with the major return code 83.

The cancel notification is always received without user data.

Receive-Negative-Response

Your program uses the receive-negative-response response indicator to receive an indication that the other program encountered an error when it was receiving data.

Your program must issue an input operation to receive the eight character sense code that the other program (or intra-system) sends with the negative-response indication.

Receipt of a negative-response function is also indicated by the 8319 return code. Refer to Appendix B for a description of the return code.

Receive-Turnaround

Your program uses the receive-turnaround response indicator to receive an indication from the other program indicating that it is ready to receive data.

The presence of the turnaround indication is also indicated by the minor return codes 00, 04, 14, and 44 with the major return code 00 (user data received) or 02 (user data received but program is being ended), or by the minor return codes 00 and 14 with the major return code 03 (no data received).

Receive-Detach

Your program uses the receive-detach response indicator to receive an indication when the received data ends a trans-action (the detach request has been received).

The presence of the detach request is also indicated by the minor codes 08 (detach only) and 1C (detach and confirm request) with the major return codes 00 (user data received), 02 (user data received but program is being ended), or 03 (no data received).

Using the Input/Output Feedback Area

Your program may have access to the file-dependent input/output (I/O) feedback area. If it does, you should be aware of certain fields when writing applications using intra-system communications:

Actual received data length
: This field contains the length of the data received on an input operation.

Major return code
: This field contains the major return code indicating the status of input and output operations.

Minor return code
: This field contains the minor return code indicating the status of input and output operations.

Request-to-write indicator
: This field indicates whether the other program has requested permission to send data.

Format name
: This field contains the record format name used to receive the data on an input operation.

Using Return Codes

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

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

Example: On an input operation, a major return code of 00 indicates that data was received. Along with this major code, intrasystem communications could return one of these minor codes:

- 01: Indicates that your program should continue receiving data.
- 08: Indicates the other program has ended the transaction. Your program can do one of the following:
  - If it is a source program, issue another evoke function or end the session.
  - If it is the target program, end the session or go to end of job.
- 1C: Indicates the program with which your program is communicating wants to end the transaction and requested confirmation. Your program must first respond either positively or negatively to the confirmation request. If your program responds positively, it should continue as for the 08 minor code. If it responds negatively, it should then inform the other program why it responded negatively or it can go to end of job without performing error recovery. In any case, if your program responds negatively, it is responsible for the appropriate error recovery.

Example, Error Condition: Another example would be a major code of 83. In this case, an error was detected that may be recoverable. Different minor codes can be returned, just as for the 00 major. For example, if your program receives a CD minor return code, your program has issued a confirm function that is currently not allowed. Your program is using a transaction that was not started with a synchronization level of confirm. For this return code, your program is responsible for the necessary error recovery. The session and transaction are still active and you can recover from this error by issuing the operation without the confirm function.

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

Refer to Appendix B for a description of the return codes that can be returned to your application when it is using intrasystem communications.

Chapter 5. Considerations for Intrasystem Communications

This chapter describes the application and performance considerations for intrasystem communications.

Application Considerations

Before writing programs that use intrasystem communications, you must understand some of the characteristics of the AS/400 environment.

General Considerations

These general considerations apply to your program and the program with which it is communicating.

- The first operation following the acquire operation by a source program should be a write operation with an evoke function specified. The evoke function starts the program with which the source program is going to communicate.
- The source program can send program initialization parameters, with the evoke function, to the other program only if the other program supports the receipt of these parameters.
- Target programs on the AS/400 system establish a connection to the session and transaction (started by the source program) by issuing an acquire operation to the program device associated with the remote location name *REQUESTER (requesting program device).
- When a program is in receive state, it can issue a read operation, an invite function, or a request-to-write function. A write operation issued with a fail function can also be used if your program is to send an error condition to the other program.
- When a program is in send state, all operations except open (to the opened ICF file), acquire (to the same program device), negative-response, and the request-to-write function are generally valid.
- When a program is receiving data, it should continue to issue input operations until one of the following is received:
  - A minor return code indicating that your program may now send data. The RCVTRNRND keyword can also be used.
  - A minor return code indicating that detach has been received. The RCVDETACH keyword can also be used.
  - A major and minor return code indicating an error condition, for example, any of the 80 major return codes.
  - Data that contains an error, in which case the program should issue a negative-response or fail function.
- To increase performance, two communicating programs can change states implicitly without using the turnaround indication by synchronizing their input and output operations. For example, if Program A is sending data and issues a read operation to Program B, which is receiving data, Program B can issue a write operation without having received a turnaround indication. Program A is then ready to receive data and Program B can send data.

Open/Acquire Considerations

The following information describes how the acquire operation is used to start a session between the source program (Program A) and the program with which it is communicating (Program B).

- If Program B acquires a program device, other than the requesting program device, a new session is established and the connection with the source program (Program A) is not established. No error is indicated because it is valid for Program B to be a target on one session and to be a source program on another session. If the program issues an input operation as the first operation to the newly established session, and an evoke function has not yet been issued, it will receive a return code indicating that no transaction is active.
- Multiple sessions (that run at the same time) can be established with multiple programs. The program device names are used to distinguish the sessions within your program.

Input Considerations

The following information describes the input considerations for your program.

- The receive indicators RCVTRNRND, RCVDETACH, RCVENDGRP, and RCVCONFIRM can be received either with data or without data (indicators only). Your program should examine the major return codes in the communications device-dependent feedback area to determine if the record contains data. A major code of 00 or 02 indicates data has been received, and a major code of 03 indicates no data has been received.
- The actual received data length can be determined from the file-dependent I/O feedback area.
- When a write operation is issued following an invite, the system performs an implicit cancel-invite function and your program can begin sending data to the other program, provided data was not waiting to be received.
- When a read operation is issued following a write, or following a read in which turnaround was received, the system performs an implicit allow-write function and begins waiting for data from the other program.

Confirm Considerations

The following information describes how the confirm function is used by both your program and the other program with which it is communicating.

- Your program requests that the other program confirm receiving the data by issuing an output request with the confirm function.
- Your program is notified that it has received a confirmation request from the other program in the following ways:
  - A major return code of 00 or 02, with the minor return codes 14, 15, 17, 1C, 44, 45, or 47, or a major return code of 03 with the minor return codes 14, 15, 17, or 1C.
  - The RCVCONFIRM indicator is set.

  Once your program has received a confirmation request, it must either respond positively or negatively to the request as follows:
  - To respond positively, issue the respond-to-confirm function, or issue any input/output operation except the fail, negative-response, or end-of-session function.
  - To respond negatively to the request, do the following:
    - Issue a fail function. In this case, your program is responsible for the appropriate level of error recovery.
    - Abnormally end the transaction and session by issuing either an end-of-session function or a close operation.
- When it is essential to your application program that the other program be started before you issue output operations to it, specify the confirm function with the evoke function. The evoke function will not complete until the other program responds to the confirmation request.
- Because the output operation with the confirm function specified waits for a positive or negative response before control is returned to the program, the source and target programs should be coded to minimize the amount of time between receiving the confirm request and sending the response. If the program receiving the confirm request performs complex processing before sending a response, the delay time can be significant.

Release, End-of-Session, and Close Considerations

The following information describes how the close operation and release and end-of-session functions are used to end communications between your program and the program with which it is communicating.

- The close operation and end-of-session function are valid in any state, but will abnormally end an active transaction with the other program and could also indicate a logic error in the program.
- The target program cannot begin error recovery using release, close, and open and acquire logic. When a permanent session error occurs, the source program is responsible for recovery.
- A release operation performed by the target program does not perform a detach function. The transaction with the source program can be resumed by a subsequent acquire of the requesting program device. That acquire can be performed either by the program that initially had the transaction or by another program running in the same job.
- A transaction remains allocated to a target job until the job ends even though a close or release operation was issued and a detach sent. As long as the job is active, the Work with Active Jobs (WRKACTJOB) command or the Work Configuration Status (WRKCFGSTS) command shows the job as an intrasystem communications target program. You can use the end-of-session function to end the session associated with a job. In this case, the job no longer shows as active.

Performance Considerations

If your program issues more than one evoke function, poor system performance may result. This is because each evoke function that results in a successful transaction causes a job to be started on the system. Because job initializations require a fair amount of system resources, you should design your application to minimize evoke functions.

In general, your program should issue multiple evoke functions only when jobs are long-running or when multiple target jobs need to run at the same time. If you issue a large number of evoke functions in your program, you can use prestart jobs, described on 5-2, to minimize the time required to start a job.

Prestarting Jobs for Program Start Requests

A program start request is a request made by your program to start another program. When your program issues an evoke function, this signals a program start request to the intrasystem communications support.

To minimize the time required to carry out a program start request, you can use the prestart jobs entry to start a job for the other program before it receives the program start request. To use prestart jobs, you need to define both communications and prestart job entries in the same subsystem description, and make certain programming changes to the prestart job program with which your program communicates. For information about how to use prestart jobs, refer to the *ICF Programming* book.

Appendix A. Language Operations, Data Description, Specifications Keywords, and System-Supplied Formats

This appendix contains charts that show the following for intrasystem communications:

- All valid language operations supported by ICF
- Valid operations for each programming language that supports ICF
- Data description specifications (DDS) processing keywords
- System-supplied formats

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.
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/400, C Set ++ for OS/400, and ILE RPG/400 only.
Release	Attempts to end a session.
Close	Closes the ICF file.

Figure A-2 shows all the valid operations for each programming language that supports ICF (LE C/400, C Set ++ for OS/400, ILE COBOL/400, and ILE RPG/400 programming languages).

Figure A-2. Valid Operations for Programming Languages

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

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

Data Description Specifications Keywords

The following lists the data description specifications (DDS) processing keywords that are valid for intrasystem communications.

DDS Keyword	Description
ALWWRT	Allow-write
CANCEL	Cancel
CNLINVITE	Cancel-invite
CONFIRM	Confirm
DETACH	Detach (End of transaction)
ENDGRP	End-of-group
EOS	End-of-session
EVOKE	Evoke
FAIL	Fail
FMH	Function-management-header
FMTNAME	Format-name
FRCDTA	Force-data
INVITE	Invite
NEGRSP	Negative-response
RCVCANCEL	Receive-cancel
RCVCONFIRM	Receive-confirm
RCVDETACH	Receive-detach
RCVENDGRP	Receive-end-of-group
RCVFAIL	Receive-fail
RCVFMH	Receive-function-management-header
RCVNEGRSP	Receive-negative-response
RCVTRNRND	Receive-turnaround
RECID	Record-identification
RQSWRT	Request-to-write
RSPCONFIRM	Respond-to-confirm
SECURITY	Security
SUBDEV	Subdevice
SYNLVL	Synchronization level
TIMER	Timer
VARLEN	Variable-length data

System-Supplied Formats

The following lists all the keyword functions performed by the system-supplied formats that are valid for intrasystem communications.

System-Supplied *Formats	Description
$$CANL	Cancel with invite
$$CANLNI	Cancel
$$CNLINV	Cancel-invite
$$EOS	End-of-session
$$EVOK	Evoke with invite
$$EVOKET	Evoke with detach
$$EVOKNI	Evoke
$$FAIL	Fail
$$NRSP	Negative-response with invite
$$NRSPNI	Negative-response
$$RCD	Request-to-write with invite
$$SEND	Send with invite
$$SENDE	Send with end-of-group
$$SENDET	Send with detach
$$SENDFM	Send function-management-header with invite
$$SENDNF	Send with function-management-header
$$SENDNI	Send
$$TIMER	Timer

Appendix B. Return Codes, Messages, and Sense Codes

Return Codes

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

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

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

Notes:

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

Major Code 00

Major Code 00 – Operation completed successfully.

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

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

Code	Description/Action
0000	**Description:** For input operations issued by your program, 0000 indicates that your program received some data with a turnaround indication. The other program is ready to receive data. For output operations issued by your program, 0000 indicates that the last output operation completed successfully and that your program can continue to send data. **Action:** If your program received a turnaround on an input operation, issue an input or output operation. For the actions which can be taken after 0000 is received, refer to the following table:

Figure B-1. Actions for Return Code 0000

Type of Session	Last Operation Issued	Actions Your Program Can Take
Started by a source program	Acquire or open	Issue an evoke or timer function, or a get-attributes operation.
	Evoke with detach or write with detach	Issue another evoke function, issue a release operation, continue local processing, or end.
	Any other output operation	Issue another output operation (except evoke), or issue an input operation.
	End-of-Session	Continue local processing or end.
Started by a remote program start request[1]	Acquire or open	Issue an input or output operation.
	Write with detach	Continue local processing or end. This session has ended.
	Any other output operation	Issue another output operation (except evoke), or issue an input operation.
	End-of-Session	Continue local processing or end.

[1] A target program (started by a program start request) cannot issue an evoke function in this session; it can issue an evoke function only in a different session that it has first acquired.

0001 **Description:** On a successful input operation, your program received some data. Your program must continue to receive data until it receives a turnaround indication (which allows your program to send data) or a detach indication.

Action: Issue another input operation. If your program detects a turnaround indication, it can issue an output operation.

0003 **Description:** On a successful input operation, your program received some data with an end-of-group indication.

Action: Issue an input operation to receive the next group of records.

0004 **Description:** On a successful input operation, your program received some data with a function-management-header (FMH) and a turnaround indication. The other program is ready to receive data.

Action: Issue an output operation.

0005 **Description:** On a successful input operation, your program received some data with a function-management-header (FMH).

Action: Your program can issue another input operation to continue receiving data until it receives a turnaround indication or a detach indication.

0007 **Description:** On a successful input operation, your program received a function-management-header (FMH) and an end-of-group indication. Your program should continue to receive data.

Action: Issue another input operation to receive the next group of records.

0008 **Description:** On a successful input operation, your program received a detach indication with the last of the data. The communications transaction with the other program has ended.

Action: If your program started the session, it can issue another evoke function (to start another program), issue a release operation (to perform local processing or to start another session), or end. If a program start request from the other program started the transaction, your program can either issue an end-of-session function or end.

0010 **Description:** On a successful output operation, your program received a request-to-write indication. The other program wants to send data as soon as possible. You should allow the other program to send this data.

Action: Issue an input operation as soon as possible.

0014 **Description:** On a successful input operation, your program received some data with a turnaround indication. In addition, the other program requested confirmation.

Action: Process any data received with the request. If your program detects no errors, respond to the confirm request with a respond-to-confirm (RSPCONFIRM) function, then issue an input or output operation. If your program does detect an error, issue a fail function, or end your program.

0015 **Description:** On a successful input operation, your program received some data. In addition, the other program requested confirmation.

Action: Process any data received with the request. If your program detects no errors, respond to the confirm request with a respond-to-confirm (RSPCONFIRM) function, then issue an input operation. If your program does detect an error, issue a fail function, or end your program.

0017 **Description:** On a successful input operation, your program received some data with an end-of-group indication. In addition, the other program requested confirmation.

Action: Process any data received with the request. If your program detects no errors, respond to the confirm request with a respond-to-confirm (RSPCONFIRM) function, then issue another input operation. If your program does detect an error, issue a fail function, or end your program.

001C **Description:** On a successful input operation, your program received some data with a detach indication. In addition, the other program requested confirmation.

Action: If your program detects no errors, it should respond to the confirm request with a respond-to-confirm (RSPCONFIRM) function, and then:

- If your program started the transaction, it can issue another evoke operation (to start another program), issue a release operation (to perform local processing or to start another session), or end.
- If a program start request from the other program started the transaction, your program can issue an end-of-session function or end.

If your program does detect an error, issue a fail operation. The transaction remains active, and your program and the other program can perform the necessary error recovery. If your program detects an error and wants to end the transaction abnormally, issue an end-of-session function, or end your program.

0044 **Description:** On a successful input operation, your program received some data with a function-management-header (FMH) and a turnaround indication. In addition, the other program requested confirmation.

Action: If your program detects no errors, respond to the confirm request with a respond-to-confirm (RSPCONFIRM) function, then issue an output operation. If your program does detect an error, issue a fail function, or end your program.

0045 **Description:** On a successful input operation, your program received some data with a function-management-header (FMH). In addition, the other program requested confirmation.

Action: If your program detects no errors, respond to the confirm request with a respond-to-confirm (RSPCONFIRM) function, then issue an input operation. If your program does detect an error, issue a fail function, or end your program.

0047 **Description:** On a successful input operation, your program received some data with a function-management-header (FMH) and an end-of-group indication. In addition, the other program requested confirmation.

Action: If your program detects no errors, respond to the confirm request with a respond-to-confirm (RSPCONFIRM) function, then issue an input operation. If your program does detect an error, issue a fail function, or end your program.

Major Code 02

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

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

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

Code	Description/Action
0200	**Description:** On a successful input operation, your program received some data with a turnaround indication. Also, your job is being ended (controlled). The other program is ready to receive data from your program. **Action:** Your program can issue an input or output operation. However, the recommended action is to complete all processing and end your program as soon as possible. The system eventually changes a job ended (controlled) to a job ended (immediate) and forces all processing to stop for your job.
0201	**Description:** On a successful input operation, your program received some data. Also, your job is being ended (controlled). Your program can continue to receive data until it receives a turnaround indication (which allows your program to send data) or a detach indication. **Action:** Your program can issue another input operation. If your program detects the equivalent of a turnaround indication, it can issue an output operation. However, the recommended action is to complete all processing and end your program as soon as possible. The system eventually changes a job ended (controlled) to a job ended (immediate) and forces all processing to stop for your job.
0203	**Description:** On a successful input operation, your program received some data with an end-of-group indication. Also, your job is being ended (controlled). **Action:** Your program can issue an input operation to receive the next group of records. However, the recommended action is to complete all processing and end your program as soon as possible. The system eventually changes a job ended (controlled) to a job ended (immediate) and forces all processing to stop for your job.
0204	**Description:** On a successful input operation, your program received some data with a function-management-header (FMH) and a turnaround indication. Also, your job is being ended (controlled). The other program is ready to receive data. **Action:** Your program can issue an output operation. However, the recommended action is to complete all processing and end your program as soon as possible. The system eventually changes a job ended (controlled) to a job ended (immediate) and forces all processing to stop for your job.
0205	**Description:** On a successful input operation, your program received some data with a function-management-header (FMH). Also, your job is being ended (controlled). **Action:** Your program can issue another input operation to continue receiving data until it receives a turnaround indication or a detach indication. However, the recommended action is to complete all processing and end your program as soon as possible. The system eventually changes a job ended (controlled) to a job ended (immediate) and forces all processing to stop for your job.

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

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

0208 **Description:** On a successful input operation, your program received a detach indication with the last of the data. The communications transaction with the other program has ended. Also, your job is being ended (controlled).

Action: If your program started the session, it can issue another evoke function (to start another program), issue a release operation (to perform local processing or to start another session), or end. If a program start request from the other program started the transaction, your program can either issue an end-of-session function or end. However, the recommended action is to complete all processing and end your program as soon as possible. The system eventually changes a job ended (controlled) to a job ended (immediate) and forces all processing to stop for your job.

0214 **Description:** On a successful input operation, your program received some data with a turnaround indication. In addition, the other program requested confirmation. Also, your job is being ended (controlled).

Action: Process any data received with the request. If your program detects no errors, respond to the confirm request with a respond-to-confirm (RSPCONFIRM) function, then issue an input or output operation. If your program does detect an error, issue a fail function, or end your program. However, the recommended action is to complete all processing and end your program as soon as possible. The system eventually changes a job ended (controlled) to a job ended (immediate) and forces all processing to stop for your job.

0215 **Description:** On a successful input operation, your program received some data. In addition, the other program requested confirmation. Also, your job is being ended (controlled).

Action: Process any data received with the request. If your program detects no errors, respond to the confirm request with a respond-to-confirm (RSPCONFIRM) function, then issue an input operation. If your program does detect an error, issue a fail function, or end your program. However, the recommended action is to complete all processing and end your program as soon as possible. The system eventually changes a job ended (controlled) to a job ended (immediate) and forces all processing to stop for your job.

0217 **Description:** On a successful input operation, your program received some data with an end-of-group indication. In addition, the other program requested confirmation. Also, your job is being ended (controlled).

Action: Process any data received with the request. If your program detects no errors, respond to the confirm request with a respond-to-confirm (RSPCONFIRM) function, then issue another input operation. If your program does detect an error, issue a fail function, or end your program. However, the recommended action is to complete all processing and end your program as soon as possible. The system eventually changes a job ended (controlled) to a job ended (immediate) and forces all processing to stop for your job.

021C **Description:** On a successful input operation, your program received some data with a detach indication. In addition, the other program requested confirmation. Also, your job is being ended (controlled).

Action: If your program detects no errors, it should respond to the confirm request with a respond-to-confirm (RSPCONFIRM) function, and then:

- If your program started the transaction, it can issue another evoke operation (to start another program), issue a release operation (to perform local processing or to start another session), or end.
- If a program start request from the other program started the transaction, your program can issue an end-of-session function or end.

If your program does detect an error, issue a fail operation. The transaction remains active, and your program and the other program can perform the necessary error recovery. If your program detects an error and wants to end the transaction abnormally, issue an end-of-session function, or end your program.

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

0244 **Description:** On a successful input operation, your program received some data with a function-management-header (FMH) and a turnaround indication. In addition, the other program requested confirmation. Also, your job is being ended (controlled).

Action: If your program detects no errors, respond to the confirm request with a respond-to-confirm (RSPCONFIRM) function, then issue an output operation. If your program does detect an error, issue a fail function, or end your program. However, the recommended action is to complete all processing and end your program as soon as possible. The system eventually changes a job ended (controlled) to a job ended (immediate) and forces all processing to stop for your job.

0245 **Description:** On a successful input operation, your program received some data with a function-management-header (FMH). In addition, the other program requested confirmation. Also, your job is being ended (controlled).

Action: If your program detects no errors, respond to the confirm request with a respond-to-confirm (RSPCONFIRM) function, then issue an input operation. If your program does detect an error, issue a fail function, or end your program. However, the recommended action is to complete all processing and end your program as soon as possible. The system eventually changes a job ended (controlled) to a job ended (immediate) and forces all processing to stop for your job.

0247 **Description:** On a successful input operation, your program received some data with a function-management-header (FMH) and an end-of-group indication. In addition, the other program requested confirmation. Also, your job is being ended (controlled).

Action: If your program detects no errors, respond to the confirm request with a respond-to-confirm (RSPCONFIRM) function, then issue an input operation. If your program does detect an error, issue a fail function, or end your program. However, the recommended action is to complete all processing and end your program as soon as possible. The system eventually changes a job ended (controlled) to a job ended (immediate) and forces all processing to stop for your job.

Major Code 03

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

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

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

Code	Description/Action
0300	**Description:** On a successful input operation, your program received a turnaround indication without any data. The session is still active. **Action:** Issue an input or output operation.
0301	**Description:** On a successful input operation, your program received no data. Your program must continue to receive input until it receives a turnaround or detach indication. **Action:** Issue an input operation.
0302	**Description:** On a successful input operation, your program received a fail indication without any data. Either the other program has sent a fail function, or the system has detected a break condition. **Action:** Issue an input operation to receive the reason for the fail from the other program.
0303	**Description:** On a successful input operation, your program received an end-of-group indication without any data. **Action:** Issue another input operation.
0308	**Description:** On a successful input operation, your program received a detach indication without any data. The communications transaction with the other program has ended. If you specified the DDS keyword RCVDETACH, the receive-detach indicator is also set on. **Action:** If your program started the session, it can issue another evoke function (to start another program), issue a release operation (to perform local processing or to start another session), or end. If a program start request from the other program started the transaction, your program can either issue an end-of-session function or end.
0309	**Description:** On a read-from-invited-program-devices operation, your program did not receive any data. Also, your job is being ended (controlled). **Action:** Your program can continue processing. However, the recommended action is to complete all processing and end your program as soon as possible. The system eventually changes a job ended (controlled) to a job ended (immediate) and forces all processing to stop for your job. **Messages:** `CPF4741 (Notify)`
0310	**Description:** On a read-from-invited-program-devices operation, the time interval specified by a timer function in your program or by the WAITRCD value specified for the ICF file expired. **Action:** Issue the intended operation after the specified time interval has ended. For example, if you were using the time interval to control the length of time to wait for data, you can issue another read-from-invited-program-devices operation to receive the data. **Note:** Since no specific program device name is associated with the completion of this operation, the program device name in the common I/O

feedback area is set to *N. Therefore, your program should not make any checks based on the program device name after receiving the 0310 return code.

Messages:

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

0314 **Description:** On a successful input operation, your program received a turn-around indication without any data. In addition, the other program requested confirmation.

Action: If your program detects no errors, respond to the confirm request with a respond-to-confirm (RSPCONFIRM) function, then issue an input or output operation. If your program does detect an error, issue a fail function, or end your program.

0315 **Description:** On a successful input operation, your program did not receive any data. In addition, the other program requested confirmation.

Action: If your program detects no errors, respond to the confirm request with a respond-to-confirm (RSPCONFIRM) function, then issue an input operation. If your program does detect an error, issue a fail function, or end your program.

0317 **Description:** On a successful input operation, your program received an end-of-group indication without any data. In addition, the other program requested confirmation.

Action: If your program detects no errors, respond to the confirm request with a respond-to-confirm (RSPCONFIRM) function, then issue an input operation. If your program does detect an error, issue a fail function, or end your program.

031C **Description:** On a successful input operation, your program received a detach indication without any data. In addition, the other program requested confirmation.

Action: If your program detects no errors, it should respond to the confirm request with a respond-to-confirm (RSPCONFIRM) function, and then:

- If your program started the transaction, it can issue another evoke operation (to start another program), issue a release operation (to perform local processing or to start another session), or end.
- If a program start request from the other program started the transaction, your program can issue an end-of-session function or end.

If your program does detect an error, issue a fail operation. The transaction remains active, and your program and the other program can perform the necessary error recovery. If your program detects an error and wants to end the transaction abnormally, issue an end-of-session function, or end your program.

Major Code 04

Major Code 04 – Output exception occurred.

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

Action: Issue an input operation to receive the data.

Code	Description/Action
0402	**Description:** Your program was sending data when a fail indication was received. Your program is now in receive state. **Action:** Issue an input operation. **Messages:** CPF4806 (Notify)
0412	**Description:** An output exception occurred because your program attempted to send data when it should be receiving data that was sent by the other program. The data from your output operation was not sent. Your program can attempt to send the data later. **Action:** Issue an input operation to receive the data. **Note:** If your program issues another output operation before an input operation, your program receives a return code of 831C. **Messages:** CPF4750 (Notify) CPF5076 (Notify)

Major Codes 08 and 11

Major Codes 08 and 11 – Miscellaneous program errors occurred.

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

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

Code	Description/Action
0800	**Description:** The acquire operation just attempted by your program was not successful. Your program tried to acquire a program device that was already acquired and is still active. **Action:** If the session associated with the original acquire operation is the one needed, your program can begin communicating in that session since it is already available. If you want a different session, issue another acquire operation for the new session by specifying a different program device name in the PGMDEV parameter of the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command that precedes the program. **Messages:** CPD4077 (Diagnostic) CPF5041 (Status) CPF50A0 (Status)
1100	**Description:** The read-from-invited-program-devices operation just attempted by your program was not successful because your program tried this operation when no program devices were invited and no timer function was in effect. **Action:** Issue an invite function (or a combined operation that includes an invite) followed by a read-from-invited-program-devices operation. **Messages:** CPF4740 (Notify)

Major Code 34

Major Code 34 – Input exception occurred.

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

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

Code	Description/Action
3401	**Description:** The input operation issued by your program was not successful because the length of the data record sent by the other system was longer than the length specified for your program's input buffer. The length of the data record received from the other system, if available, is in the actual-record-length field in the I/O feedback area. **Action:** Issue another input operation if your program can specify a record size large enough to receive the data, plus any indicators for a file without a separate indicator area. Otherwise, you should close the file, end your program, correct the record size, then run your program again. **Messages:** `CPF4768 (Notify)`
3441	**Description:** A valid record format name was specified with format selection type *RMTFMT or *RECID. However, although the data received matched one of the record formats in the ICF file, it did not match the format specified on the read operation. **Action:** Correct your program to issue a read operation that does not specify a record format name, or specify the correct record format name to process the data based on the format selection option for the file. **Messages:** `CPF5058 (Notify)`
3451	**Description:** Your program specified a file record size that was not large enough for the indicators to be included with the data sent by the other program (for a file defined with a nonseparate indicator area). Your program did not receive any data. For a file using a nonseparate indicator area, the actual record length field in the device-dependent I/O feedback area contains the number of indicators specified by the record format. **Action:** End the session; close the file; correct the file record size; then open the file again. **Messages:** `CPF4768 (Notify)`

Major Code 80

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

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

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

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

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

Code	Description/Action

8081	**Description:** The operation attempted by your program was not successful because a system error condition was detected.
	
	Action: Your communications configurations may need to be varied off and then on again. Your program can do one of the following:
	• Continue local processing.
	• Close the ICF file, open the file again, and establish the session again.
	• End.
	
	Messages:
	`CPF4170 (Escape)`
	`CPF4510 (Escape)`
	`CPF4566 (Escape)`
	`CPF5257 (Escape)`

8082	**Description:** The operation attempted by your program was not successful because the device supporting intrasystem communications between your program and the other program is not usable. For example, this may have occurred because communications were stopped for the device by a Hold Communications Device (HLDCMNDEV) command. Your program should not issue any operations to the device.
	
	Action: Communications with the remote program cannot resume until the device has been reset to a varied on state. If the device has been held, use the Release Communications Device (RLSCMNDEV) command to reset the device. If the device is in an error state, vary the device off and then on again. Your program can attempt to establish the session again, continue local processing, or end.
	
	Messages:
	`CPF4744 (Escape)`
	`CPF5269 (Escape)`

80B3	**Description:** The open operation issued by your program was not successful because the ICF file is in use by another process. **Action:** Wait for the file to become available, then issue another open operation. Otherwise, your program may continue processing, or it can end.

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

Messages:

CPF4128 (Escape)

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

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

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

- If the update and delete options are not supported for the program device, use an option of input, or output, or both.
- If your program tried selecting user buffers and blocked data together, it should try selecting one or the other, but not both.
- If your program tried to open a non-source file as a source file, either change the file name or change the library name.
- If there was a mismatch on the INDARA keyword, either correct the file or correct your program so that the two match.
- If no program devices were previously acquired for a shared file, acquire one or more program devices for the file.

Messages:

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

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

Action: Close the file. Compile your program again to match the file level of the ICF file, or change or override the file to LVLCHK(*NO); then open the file again.

Messages:

CPF4131 (Escape)

80EF **Description:** Your program attempted an open operation on a file or library for which the user is not authorized.

Action: Close the file. Either change the file or library name on the open operation, or obtain authority for the file or library from your security officer. Then issue the open operation again.

Messages:

CPF4104 (Escape)

80F8 **Description:** The open operation attempted by your program was not successful because one of the following occurred:

- The file is already open.
- The file is marked in error on a previous return code.

Action:

- If the file is already open, close the file and end your program. Remove the duplicate open operation from your program, then issue the open operation again.
- If the file is marked in error, your program can check the job log to see what errors occurred previously, then take the appropriate recovery action for those errors.

Messages:

```
CPF4132 (Escape)
CPF5129 (Escape)
```

Major Code 81

Major Code 81 – Permanent session error (irrecoverable).

Description: An irrecoverable session error occurred during an I/O operation. Your session cannot continue and has ended. Before communications can resume, the session must be established again by using an acquire operation or another program start request. Recovery from this error is unlikely until the problem causing the error is detected and corrected. Operations directed to other sessions associated with the file should work.

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

If your program initiated the session, you can:

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

If your session was initiated by a program start request from the other program, you can:

- Continue processing without the session.
- End.

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

Code **Description/Action**

8140 **Description:** A cancel reply was received from your program or from the operator in response to a notify message, or was the result of a system default, causing the session to be ended. The session is no longer active.

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

Messages:

```
CPF5104 (Escape)
```

81E9 **Description:** An input operation was issued and the format selection option for the ICF file was *RECID, but the data received did not match any record formats in the file. There was no format in the file defined without a RECID

keyword, so there was no default record format to use. The session has ended.

Action: Verify that the data sent by the other program was correct. If the data was not correct, change the other program to send the correct data. If the data was correct, add a RECID keyword definition to the file that matches the data, or define a record format in the file without a RECID keyword so that a default record format can be used on input operations. If your program started the session, use another acquire operation to start the session again. If a program start request started your program, continue local processing or end.

Messages:

```
CPF5291 (Escape)
```

Major Code 82

Major Code 82 – Open or acquire operation failed.

Description: Your attempt to establish a session was not successful. The error may be recoverable or permanent, and recovery from it is unlikely until the problem causing the error is detected and corrected.

Action: You can perform the following general actions for all 82xx return codes. Specific actions are given in each minor code description.

If your program was attempting to start the session, you can:

- Correct the problem and attempt to establish the session again. The next operation could be successful only if the error occurred because of some temporary condition. If the operation is still not successful, your program should end.
- Continue processing without the session.
- End.

If your session was initiated by a program start request from the other program, you can:

- Correct the problem and attempt to connect to the requesting program device again. If the operation is still not successful, your program should end.
- Continue processing without the session.
- End.

Several of the minor codes indicate that an error condition must be corrected by changing a value in the communications configuration or in the file.

- To change a parameter value in the communications configuration, vary the configuration off, make the change to the configuration description, then vary the configuration on.
- To change a parameter value in the file, use the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command.

  Note: When a parameter can be specified both in the ADDICFDEVE or OVRICFDEVE command and in the configuration, the value in the ADDICFDEVE or OVRICFDEVE command overrides the value specified in the configuration (for your program only). Therefore, in some cases, you may choose to make a change with the ADDICFDEVE or OVRICFDEVE command rather than in the configuration.

If no changes are needed in your file or in the configuration (and depending on what the return code description says):

- If the attempted operation was an acquire, issue the acquire operation again.
- If the attempted operation was an open, close the file and issue the open operation again.

Code	Description/Action
8209	**Description:** The open or acquire operation issued by your program was not successful because a prestart job is being canceled. One of the following may have occurred: • An End Job (ENDJOB), End Prestart Job (ENDPJ), End Subsystem (ENDSBS), End System (ENDSYS), or Power Down System (PWRDWNSYS) command was being issued. • The maximum number of prestart jobs (MAXJOBS parameter) was reduced by the Change Prestart Job Entry (CHGPJE) command. • The value for the maximum number of program start requests allowed (specified in the MAXUSE parameter on the ADDPJE or CHGPJE command) was exceeded. • Too many unused prestart jobs exist. • The prestart job had an initialization error. **Action:** Complete all processing and end your program as soon as possible. Correct the system error before starting this job again. **Messages:** CPF4292 (Escape) CPF5313 (Escape)
8233	**Description:** A program device name that was not valid was detected. Either an ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command was not run, or the program device name in your program does not match the program device name specified in the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command for the session being acquired. The session was not started. **Action:** If the error was in your program, change your program to specify the correct program device name. If an incorrect identifier was specified in the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command, specify the correct value in the PGMDEV parameter. **Messages:** CPF4288 (Escape) CPF5068 (Escape)
8281	**Description:** On an unsuccessful open or acquire operation, a system error condition was detected. For example, the file may previously have been in error, or the file could not be opened due to a system error. **Action:** Your communications configurations may need to be varied off and then on again. Your program can do one of the following: • Continue local processing. • Close the ICF file, open the file again, and acquire the program device again. However, if this results in another 8281 return code, your program should close the file and end. • Close the file and end. **Messages:** CPF4168 (Escape) CPF4182 (Escape) CPF4369 (Escape) CPF4370 (Escape) CPF4375 (Escape) CPF5257 (Escape) CPF5274 (Escape) CPF5317 (Escape) CPF5318 (Escape)
8282	**Description:** The open or acquire operation attempted by your program was not successful because the device supporting intrasystem communications between your program and the other program is not usable. For example, this

may have occurred because communications were stopped for the device by a Hold Communications Device (HLDCMNDEV) command. Your program should not issue any operations to the device. The session was not started.

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

Messages:

```
CPF4298 (Escape)
CPF5269 (Escape)
```

82A8 **Description:** The acquire operation attempted by your program was not successful because the maximum number of program devices allowed for the ICF file has been reached. The session was not started.

Action: Your program can recover by releasing a different program device and issuing the acquire operation again. If more program devices are needed, close the file and increase the MAXPGMDEV value for the ICF file.

Messages:

```
CPF4745 (Diagnostic)
CPF5041 (Status)
```

82A9 **Description:** The acquire operation issued by your program to a *REQUESTER device was not successful due to one of the following causes:

- Your program has already acquired the *REQUESTER device.
- The job was started by a program start request with the *REQUESTER device detached.
- The *REQUESTER device was released because an end-of-session was requested.
- The job does not have a *REQUESTER device; that is, the job was not started by a program start request.
- A CPI Communications requesting conversation is already allocated.
- A permanent error occurred on the session.

Action:

- If the *REQUESTER device is already acquired and your program expects to communicate with the *REQUESTER device, use the program device that acquired the *REQUESTER.
- If the *REQUESTER device is not available and your program expects to communicate with the *REQUESTER device, the other program must send a program start request without a detach function.
- If your program released its *REQUESTER device, correct the error that caused your program to release its *REQUESTER device before trying to acquire it.
- If this job does not have a *REQUESTER device, correct the error that caused your program to attempt to acquire a *REQUESTER device.
- If a permanent error caused the acquire operation to fail, verify that your program correctly handles the permanent error return codes (80xx, 81xx) it received on previously issued input and output operations. Because your program was started by a program start request, your program cannot attempt error recovery after receiving a permanent error return code. It is the responsibility of the other program to initiate error recovery.

Messages:

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

82AA **Description:** The open or acquire operation attempted by your program was not successful because the remote location name specified on the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command does not match any remote location configured on the system. The session was not started.

Action: Your program can continue local processing, or close the file and end. Verify that the name of the remote location is specified correctly in the RMTLOCNAME parameter on the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command.

Messages:

```
CPF4103 (Escape)
CPF4363 (Escape)
CPF4364 (Escape)
CPF4747 (Escape)
CPF5304 (Escape)
CPF5378 (Escape)
CPF5379 (Escape)
```

82AB **Description:** The open or acquire operation attempted by your program was not successful because the device description for the remote location was not varied on. The session was not started.

Action: Your program can wait until the communications configuration is varied on and then issue the acquire operation again, it can try the acquire operation again using a different device description, continue local processing, or end.

Messages:

```
CPF4304 (Escape)
CPF5355 (Escape)
```

82EA **Description:** The open or acquire operation attempted by your program was not successful. A format selection of *RECID was specified on the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command, but cannot be used with the ICF file because the RECID DDS keyword is not used on any of the record formats in the file. The session was not started.

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

Messages:

```
CPF4348 (Escape)
CPF5521 (Escape)
```

82EE **Description:** Your program attempted an open or acquire operation to a device that is not supported. Your program tried to acquire a device that is not a valid ICF communications type, or it is trying to acquire the requesting program device in a program that was not started by a program start request. The session was not started.

Action: Your program can continue local processing or end. Verify that the name of the remote location is specified correctly in the RMTLOCNAME parameter on the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command. If your program was attempting to acquire a non-ICF device, use the appropriate interface for that communications type. If your program was attempting to acquire a requesting program device, verify that your program is running in the correct environment.

Messages:

```
CPF4105 (Escape)
CPF4223 (Escape)
CPF4251 (Escape)
```

CPF4760 (Escape)
CPF5038 (Escape)
CPF5550 (Escape)

82EF **Description:** Your program attempted an acquire operation, or an open operation that implicitly acquires a session, to a device that the user is not authorized to, or that is in service mode. The session was not started.

Action: If the operation was an acquire, correct the problem and issue the acquire again. If the operation was an open, close the file, correct the problem, then issue the open operation again. To correct an authority error, obtain authority for the device from your security officer or device owner. If the device is in service mode, wait until machine service function (MSF) is no longer using the device before issuing the operation again.

Messages:

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

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

Action: End your program, correct the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command, then run your program again.

Messages:

CPF4322 (Escape)
CPF5539 (Escape)

Major Code 83

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

Description: A session error occurred, but the session may still be active. Recovery within your program might be possible.

Action: You can perform the following general actions for all 83xx return codes. Specific actions are given in each minor code description.

- Correct the problem and continue processing with the session. If the error occurred because of a resource failure on the system, a second attempt may be successful. If the operation is still not successful, your program should end the session.
- Issue an end-of-session function and continue processing without the session.
- End.

Several of the minor codes indicate that an error condition must be corrected by changing a value in the communications configuration or in the file.

- To change a parameter value in the communications configuration, vary the configuration off, make the change to the configuration description, then vary the configuration on.
- To change a parameter value in the file, use the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command.

  Note: When a parameter can be specified both in the ADDICFDEVE or OVRICFDEVE command and in the configuration, the value in the ADDICFDEVE or OVRICFDEVE command overrides the value specified in the configuration (for your program only). Therefore, in some cases, you may choose to make a change with the ADDICFDEVE or OVRICFDEVE command rather than in the configuration.

If no changes are needed in your file or in the configuration, and depending on what the return code description says, you should notify the system operator that a change is required to correct the error received.

Code	Description/Action
830B	**Description:** Your program attempted an operation that was not valid because the session was not yet acquired or has ended. The session may have ended because of a release operation, an end-of-session function, or a permanent error. Your program may have incorrectly handled a previous error. **Action:** Verify that your program does not attempt any operations without an active session. Also verify that your program correctly handles the permanent error or session-not-acquired return codes (80xx, 81xx, 82xx) it received on previously issued input and output operations. To recover from an incorrectly handled error condition, your program may or may not be able to issue another acquire operation, depending on the return code. **Messages:** CPD4079 (Diagnostic) CPF4739 (Status) CPF5067 (Escape) CPF5068 (Escape) CPF5070 (Escape)
8319	**Description:** The other program sent a negative-response with sense data. **Action:** Issue an input operation to receive the sense data. **Messages:** CPF4773 (Notify)

831A **Description:** One of the following occurred:

- The evoke function attempted by your program was not successful.
- The other program issued an end-of-session function.
- The other program ended abnormally.

Action: Your program can issue an end-of-session function, issue a different evoke function, or end.

Messages:

CPF4805 (Notify)
CPF4808 (Notify)

831B **Description:** Your program tried to specify invalid sense data on a negative-response function, or it tried to send a negative-response that has already been sent to the current chain. The data was not sent.

Action: Correct your program so that it does not issue the same negative-response more than once, and that it sends valid sense data on a negative-response function. Valid sense data must be either 0 or 8 bytes long. To send 8 bytes, the first four bytes must be 0000, 08xx, or 10xx, and the remaining four bytes must be in the ranges 0-9, A-F, or a-f. If your program chooses to send a negative-response without sense data, intrasystem communications automatically sends 08110000 to the other program.

Messages:

CPF4774 (Notify)

831C **Description:** Your program's previous output operation received a return code of 0412, indicating that your program must receive information sent by the other program; however, your program did not handle the return code correctly. The current output operation was not successful because your program should have issued an input operation to receive the information already sent by the other program.

Action: Issue an input operation to receive the previous information.

Messages:

CPF4934 (Notify)

831E **Description:** The operation attempted by your program was not valid, or a combination of operations that was not valid was specified. The session is still active. The error may have been caused by one of the following:

- Your program issued an operation that is not recognizable or not supported by intrasystem communications.
- Your program requested a combination of operations or keywords that was not valid, such as a combined write-then-read operation with the invite function specified.
- Your program issued an input operation, or an output operation with the invite or allow-write function, for a file that was opened for output only.
- Your program issued an output operation for a file that was opened for input only.
- Your program issued a close operation with a temporary close option.

Action: Your program can try a different operation, issue a release operation or end-of-session function, or end. Correct the error in your program before trying to communicate with the other program.

If the file was opened for input only, do not issue any output operations; or, if the file was opened for output only, do not issue any input operations, and do not use the invite or allow-write function on an output operation. If such an operation is needed, then release the session, close the ICF file, and open the file again for input and output.

Messages:

CPF4564 (Escape)
CPF4764 (Notify)
CPF4766 (Notify)
CPF4790 (Notify)
CPF4803 (Notify)
CPF5132 (Escape)
CPF5149 (Escape)

831F **Description:** Your program specified data or a length for the operation that was not valid; however, the session is still active. One of the following caused the error indication:

- On an output operation, your program tried to send a data record that was longer than the MAXRCDLEN value specified for the ICF file.
- The program used a read or write operation that specified a data length greater than the record format in the ICF file.
- If this was a timer function, the format of the timer interval was not HHMMSS.
- If a system-defined format was used to specify the operation, or if the variable-length-data-record (VARLEN) function was used, then the length of the user buffer was not valid.

Action: If you want your program to recover, try the operation again with a smaller data length. If you do not need your program to recover immediately, do one of the following:

- Change the record format length in the ICF file, or change the record length in your program and compile your program again.
- For an input operation, specify a data length equal to or less than the record format length, or do not specify a length at all.
- If the timer function was used, verify that the format of the timer interval is HHMMSS.
- For an output operation that used the variable-length-data-record (VARLEN) function, verify that the length specified is less than the record length specified for the ICF file when it was opened.

Messages:

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

8322 **Description:** Your program tried to issue a negative-response or a request-to-write function. These functions are only valid while your program is in receive state.

Action: Your program can issue an output operation to continue sending data, issue an input operation to begin receiving data, issue an end-of-session function to continue local processing, or end. Correct the error that caused your program to attempt the operation that was not valid.

Messages:

CPF4703 (Notify)
CPF4775 (Notify)

8323 **Description:** Your program attempted to issue a cancel function when data was received for your program. The cancel function is only valid in send state.

Action: Your program can issue an input operation to continue receiving data, issue an end-of-session function, or end. Correct the error that caused your program to attempt the invalid operation.

Messages:

CPF4776 (Notify)

CPF4809 (Notify)

8326 **Description:** Your program attempted to issue a negative-response function or a cancel function to cancel a group of records when no records were previously sent to start a group. The cancel function is only valid within a chain; it is not valid preceding a chain or between chains. The session is still active.

Action: Correct the error that caused your program to attempt the invalid operation.

Messages:

CPF4779 (Notify)
CPF4810 (Notify)

8327 **Description:** The input or output operation issued by your program was not successful because there was no active transaction. Either the transaction has ended, or the transaction was never started.

Action: If your program wants to start a transaction, it can issue an evoke function. Otherwise, it can issue an end-of-session function or end. If a coding error in your program caused the error, correct your program.

Messages:

CPF5098 (Notify)

8329 **Description:** An evoke function that was not valid was detected in this session. Your program was started by a program start request and, therefore, cannot issue any evoke functions in this session.

Action: To recover, your program can try a different operation or function. To issue an evoke function in a different session, first issue an acquire operation (using a different program device name), then try the evoke function. Otherwise, your program can issue an end-of-session function, continue local processing, or end. If a coding error caused your program to attempt an evoke that was not valid, correct your program.

Messages:

CPF5099 (Notify)

832A **Description:** Both your program and the other program were attempting to receive data at the same time.

Action: The other program is waiting to receive data from your program. Issue an output operation. If a coding error in your program caused the error, correct your program.

Messages:

CPF4807 (Notify)

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

Action: Issue an input operation to satisfy the invite function, or issue a cancel-invite function to cancel the invite function; then try the release operation again. Otherwise, issue an end-of-session function to end the session. If a coding error caused your program to attempt a release operation that was not valid, correct your program.

Messages:

CPF4769 (Notify)

832D **Description:** Following an invite function, your program issued a request-to-write indication, a negative-response indication, a cancel reply, or an additional invite function. This operation failed because the original invite function must first be satisfied by an input operation.

Action: Issue an input operation to receive the data that was invited. Otherwise, issue an end-of-session function to end the session. If a coding error caused your program to attempt a request-to-write indication or an additional invite function, correct your program.

Messages:

CPF4924 (Notify)

832F **Description:** The evoke function or release operation issued by your program was not successful because your program attempted the operation while the current transaction was still active. The operation was not performed, but the session is still active.

Action: Use the detach function to end the current transaction before issuing an evoke function or release operation. Correct the error that caused your program to issue an evoke function during an active transaction; then run your program again.

Messages:

CPF5099 (Notify)

8330 **Description:** On a successful input operation, your program received a cancel function with a turnaround indication. The other program has canceled the group of records it was sending and is now ready to receive data from your program. The session is still active.

Action: Normally, your program should discard the canceled data it received from the other program, as the data may be in error. Your program can then issue an output operation.

Messages:

CPF4782 (Notify)

8331 **Description:** On a successful input operation, your program received a cancel function without a turnaround indication. The other program has canceled the group of records it was sending, but it is still in send state, and your program is still in receive state. The session is still active.

Action: Normally, your program should discard the canceled data it received from the other program, as the data may be in error. Your program should then issue another input operation.

Messages:

CPF4783 (Notify)

8334 **Description:** The evoke function attempted by your program was not valid. A program name must be specified on the evoke function.

Action: Correct your program so that it issues the evoke correctly, then try the operation again.

Messages:

CPF4804 (Notify)

83CD **Description:** The input operation issued by your program was not successful because your program attempted a confirm function for a transaction that was started with a synchronization level of *NONE.

Action: Issue an end-of-session function and change your program to start the transaction with a synchronization level of *CONFIRM.

Messages:

CPF5016 (Notify)

83D6 **Description:** The RSPCONFIRM function issued by your program was not valid because the other program did not request confirmation, or because the current transaction was started with a synchronization level of *NONE.

Action: If the other program did not request confirmation, correct the error that caused your program to issue the RSPCONFIRM function. However, if both programs expect to use confirmation processing, the transaction must be started with a synchronization level of *CONFIRM.

Messages:

CPF4792 (Notify)

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

Action: Verify that the name of the record format in your program is correct, then check to see whether the record format is defined in the file definition.

Messages:

CPF5054 (Notify)

83E8 **Description:** Your program attempted to issue a cancel-invite function to a session that was not invited. One of the following may have occurred:

- The invite function was implicitly canceled earlier in your program by a valid output operation.
- The invite function was satisfied earlier in your program by a valid input operation.
- Your program had already canceled the invite function, then tried to cancel it again.
- Your program never invited the session.

The session is still active.

Action: Your program can issue an input or output operation, issue an end-of-session function, continue local processing, or end. However, you should correct the error that caused your program to attempt the cancel-invite to a session that was not invited.

Messages:

CPF4763 (Notify)

83F8 **Description:** Your program attempted to issue an operation to a program device that is marked in error due to a previous I/O or acquire operation. Your program may have handled the error incorrectly.

Action: Release the program device, correct the previous error, then acquire the program device again.

Messages:

CPF5293 (Escape)

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 you are running in the System/36 environment on your AS/400 system, there can be two nonzero reason codes. These two reason codes occur when the OS/400 program cannot determine whether the program start request was to start a job in the System/36 environment or by the OS/400 program. 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 by the OS/400 program. Whenever you receive two reason codes, you should determine which environment the job was to run in and correct the problem for that environment.

Figure B-2 on page B-25 describes reason codes for failed program start requests.

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

Reason Code	Reason Description
401	Program start request received to a device that is not allocated to an active subsystem.
402	Requested device is currently being held by a Hold Communications Device (HLDCMNDEV) command.
403	User profile is not accessible.
404	Job description is not accessible.
405	Output queue is not accessible.
406	Maximum number of jobs defined by subsystem description are already active.
407	Maximum number of jobs defined by communications entry are already active.
408	Maximum number of jobs defined by routing entry are already active.
409	Library on library list is exclusively in use by another job.
410	Group profile cannot be accessed.
411	Insufficient storage in machine pool to start job.
412	System value not accessible.
501	Job description was not found.
502	Output queue was not found.
503	Class was not found.
504	Library on initial library list was not found.
505	Job description or job description library is damaged.
506	Library on library list is destroyed.
507	Duplicate libraries were found on library list.
508	Storage-pool defined size is zero.
602	Transaction program-name value is reserved but not supported.
604	Matching routing entry was not found.
605	Program was not found.
704	Password is not valid.
705	User is not authorized to device.
706	User is not authorized to subsystem description.
707	User is not authorized to job description.
708	User is not authorized to output queue.
709	User is not authorized to program.
710	User is not authorized to class.
711	User is not authorized to library on library list.
712	User is not authorized to group profile.
713	User ID is not valid.
714	Default user profile is not valid.
715	Neither password nor user ID was provided, and no default user profile was specified in the communications entry.
718	No user ID.
722	A user ID was received but a password was not sent.
723	No password was associated with the user ID.
725	User ID does not follow naming convention.
726	User profile is disabled.
801	Program initialization parameters are present but not allowed.
802	Program initialization parameter exceeds 2000 bytes.
803	Subsystem is ending.
804	Prestart job is inactive or is ending.
805	WAIT(NO) was specified on the prestart job entry and no prestart job was available.
806	The maximum number of prestart jobs that can be active on a prestart job entry was exceeded.
807	Prestart job ended when a program start request was being received.
901	Program initialization parameters are not valid.
902	Number of parameters for program not valid.
903	Program initialization parameters required but not present.

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

Reason Code	Reason Description
1001	System logic error. Function check or unexpected return code encountered.
1002	System logic error. Function check or unexpected return code encountered while receiving program initialization parameters.
1501	Character in procedure name not valid.
1502	Procedure not found.
1503	System/36 environment library not found.
1504	Library QSSP not found.
1505	File QS36PRC not found in library QSSP.
1506	Procedure or library name is greater than 8 characters.
1507	Current library not found.
1508	Not authorized to current library.
1509	Not authorized to QS36PRC in current library.
1510	Not authorized to procedure in current library.
1511	Not authorized to System/36 environment library.
1512	Not authorized to file QS36PRC in System/36 environment library.
1513	Not authorized to procedure in System/36 environment library.
1514	Not authorized in library QSSP.
1515	Not authorized to file QS36PRC in QSSP.
1516	Not authorized to procedure in QS36PRC in QSSP.
1517	Unexpected return code from System/36 environment support.
1518	Problem phase program not found in QSSP.
1519	Not authorized to problem phase program in QSSP.
1520	Maximum number of target programs started (100 per System/36 environment).
2501	System logic error. Function check or unexpected return code encountered while processing a program start request.
2502	Temporarily unable to allocate needed resources for a program start request.
2503	No subsystem accepting program start requests for this device.

Appendix C. Using Intrasystem Communications to Test Applications

This appendix discusses the differences between intrasystem communications and other communications types, such as advanced program-to-program communications (APPC), binary synchronous communications equivalence link (BSCEL), Systems Network Architecture Upline Facility (SNUF), asynchronous, retail, and finance communications, in sending and receiving data. There may also be differences in the way the communications types support the start of sessions and transactions, online messages, record length, and return codes.

These differences should be noted, especially if you use intrasystem communications to test new application programs to be run using other communications types.

If your program expects to receive certain return codes and messages, these codes and messages may not be the same for intrasystem communications as they are for another communications type. You may only be able to observe a range of messages, or you may have to refer to the return codes section for a specific communications type to determine the differences for each code.

Using Intrasystem Communications for Advanced Program-to-Program Communications

Advanced program-to-program communications (APPC) allows programs on an AS/400 system to communicate with programs on other systems having compatible communications support. It also provides the capability for two programs to communicate with each other while running on the same AS/400 system. This capability is enabled when LINKTYPE(*LOCAL) is specified when the APPC controller description is created. We recommend this method to test programs which you have written to use APPC. APPC is the AS/400 implementation of the SNA LU session type 6.2 architecture. The following considerations apply when you use intrasystem communications to test programs to be run using APPC.

Confirm Function

When your program uses the confirm function, intrasystem communications sends a positive response to a confirm request if the user does a valid read or write operation. However, APPC requires the user to use the RSPCONFIRM DDS keyword to send a positive response.

Intrasystem communications also supports the confirm function with non-APPC functions, such as the end-of-group function. This results in more return codes for intrasystem communications.

Conversation Types

APPC supports both basic and mapped conversation types; intrasystem communications only supports the equivalent of a mapped conversation. Programs written for either basic or mapped conversations can run using intrasystem communications. However, intrasystem communications does no checking to ensure the two communicating programs are both using the same conversation type.

Evoke Function

When your program uses the evoke function to start another program and you specify *USER for the user ID on the SECURITY keyword, intrasystem communications always passes the user ID on the program start request. However, APPC only passes the user ID if the remote system accepts a user ID that has already been verified.

Fail Function

If your program receives a fail indication, intrasystem communications issues a 0302 return code when your program is in the receive state, and a 0402 return code when your program is in the send state. However, APPC issues 83C7 through 83CC return codes when a fail indication is received in the send or receive state.

Force-Data Function

Intrasystem communications does not buffer data, APPC does. If you use APPC, and your program issues a write operation without specifying a function that forces the data to be sent (for example, an invite function, force-data function, or read operation), the data is buffered so it can be sent later.

Output Operations

If your program attempts to send data when it should be receiving data, an output exception occurs and intrasystem communications sends your program a 0412 return code. If your program issues another output operation, it receives a 831C return code. APPC continues to send the 0412 return code.

Record Length

On input operations, if the length of the data record sent by the other program is greater than the length of your program input buffer, intrasystem communications returns a 3401 return code, and your program can issue another read operation if it can specify a record size large enough to receive the entire record. APPC truncates the data to fit in your program's input buffer, and returns a 3431 return code with the data; the data that was truncated is lost.

Variable Buffer Management (VARBUFMGT)

APPC supports the variable buffer management (VARBUFMGT) DDS keyword, whereas intrasystem communications does not.

Using Intrasystem Communications for Asynchronous Communications

Asynchronous communications is a method of communications that allows an exchange of data with a remote device or system, using either a start-stop line or an X.25 line. The following considerations apply when you use intrasystem communications to test programs to be run using asynchronous communications.

Detach Function

Intrasystem communications requires that your program issue a detach function to end a transaction before ending the session. Asynchronous communications does not support the detach function, and does not require a transaction to be ended before ending the session.

Evoke Function

Intrasystem communications requires that your program issue an evoke function as the first operation after an acquire operation, whereas asynchronous communications does not. Therefore, your first operation following an acquire operation must be an evoke function when you use intrasystem communications to test an application program which is to be run using asynchronous communications.

Fail Function

For both intrasystem and asynchronous communications, your program issues the fail function to indicate that it has detected an error in the data while it was sending or receiving. However, whereas asynchronous communications always sends a 0302 return code, intrasystem communications sends a 0302 return code when your program is in receive state, and a 0402 return code when your program is in send state. Also, asynchronous communications discards all data waiting to be received by your application whenever a fail indication is sent or received.

Function-Management-Header Function

If your program uses the function-management-header function, intrasystem communications sends the function-management-header data to the other program. If your program uses asynchronous communications and issues a write function-management-header function, it affects data translation, changes certain characteristics of data on an asynchronous communications line, or sends packet assembler/disassembler (PAD) messages.

Number of Sessions

Intrasystem communications allows multiple sessions per device; asynchronous communications allows only one session per device.

Read or Write Operations

Asynchronous communications allows your program to issue read and write operations in any order. Intrasystem communications normally requires that your application program be in the send state to issue output operations, and in the receive state to issue input operations; however, if your program is in the send state, you may issue a read operation.

Translation

Asynchronous communications supports translation of data from EBCDIC to ASCII, whereas intrasystem communications does not.

Using Intrasystem Communications with Binary Synchronous Communications Equivalence Link

Binary synchronous communications (BSC) is a data communications line protocol that uses a standard set of transmission control characters and control character sequences to send binary-coded data over a communications line. The ICF support on the AS/400 system that provides binary synchronous communications with another AS/400 system is referred to as **binary synchronous communications equivalence link (BSCEL) support**. The following considerations apply when you use intrasystem communications to test programs to be run using binary synchronous communications equivalence link (BSCEL).

Detach Function

Intrasystem communications requires that your program use a detach function to end a transaction, and the other program receives a minor return code of 08 indicating that the detach function was sent. If your program uses BSCEL and specifies RMTBSCEL(*NO), BSCEL treats a detach function as if it were an end-of-group function, and the receiving program would never receive a 08 minor return code.

End-of-Group Function

Intrasystem communications issues a return code of 0003 or 0303 to the other program when your program issues an end-of-group function (the '03' minor code indicates an end of group). BSCEL issues either a 0300 or 0301 return code to the receiving program, depending on the value specified for the GRPSEP parameter in the device description or on the ADDICFDEVE, the CHGICFDEVE, or the OVRICFDEVE command.

Evoke Function

Intrasystem communications requires that a source program issue an evoke function as the first operation after an acquire operation. However, if your program uses BSCEL and you specify RMTBSCEL(*NO) on a program device entry command, on the CRTDEVBSC command, or the CHGDEVBSC command, the evoke function is optional, and the first input or output operation from your program starts the transaction.

If your program uses intrasystem communications and the evoke function fails, a notify message is sent to your program with a reason code indicating why it failed. If your program uses BSCEL and you specify RMTBSCEL(*YES) and an evoke function fails, both a notify message and an online message are sent to

your program and your program must issue a read operation to receive the online message.

Fail Function

When intrasystem communications support receives a fail function, a return code of 0402 or 0302 is returned to your program, and you may correct the error indicated and continue sending data. When BSCEL receives a fail function (an end-of-transmission, or EOT, indication), a return code of 8197 or 8198 is returned to your program, and the session is ended.

Number of Sessions

Intrasystem communications allows multiple sessions per device; BSCEL allows only one session per device.

Online Messages

Intrasystem communications does not send or receive any online messages; BSCEL does support online messages.

Program Start Requests

When using BSCEL, a source program that specifies RMTBSCEL(*NO) for the communications session can send data in the proper format for a program start request with the program's first output operation. When using intrasystem communications, your source program cannot issue a program start request; the evoke function must be used to start another program.

Receiving Data

If you are using intrasystem communications, a return code of 0300 or 832A is used if both your program and the program with which you are communicating attempt to receive data at the same time. If you use BSCEL, both programs will be waiting to receive data indefinitely.

Receive-Turnaround Indication

If you use intrasystem communications, your program may receive a turnaround indication on the same read operation for which your program receives data. BSCEL sends the turnaround indication as a separate transmission after the data record is sent. Therefore, you may need to issue an additional read operation for BSCEL to receive the turnaround indication.

Record Blocking

Intrasystem communications does not support record blocking (that is, you cannot specify the BLOCK parameter on the program device entry commands); BSCEL does support record blocking.

Record Length

Intrasystem communications supports a maximum record length of 32767 bytes; BSCEL supports a maximum record length of 8192 bytes.

Using Intrasystem Communications for Finance Communications

Finance communications allows programs on an AS/400 system to communicate with programs using the SNA LU session type 0 protocol. The following considerations apply when you use intrasystem communications to test programs to be run using finance communications.

Allow-Write and Request-to-Write Functions

Both intrasystem and finance communications require that your program either send or receive at any given time. However, whereas intrasystem communications uses the allow-write and request-to-write functions as a way of determining which program should send or receive, finance communications does not. If you use finance communications, and neither your program nor the controller has sent a group of records, a contention state exists, in which either program may attempt to send. If both the local and the controller program send at the same time, the controller is designated the sender, and can send a negative-response indication to your program. When writing programs that use finance communications, you need to be aware of these contention error conditions.

Confirm, End-of-Group, Invite, or Read Functions Specified on Write Operations with Data

Intrasystem communications allows your program to specify a confirm function on write operations with data; finance communications does not. However, finance communications supports a function similar to the confirm function. When your program issues a write operation with data, and the end-of-group function is also specified, the data is sent to the finance controller, and the write operation does not complete until a response is received from the controller.

If you specify invite or read functions on a write operation with data, however, the data is sent to the finance controller as if an end-of-group function was specified, but no response from the finance controller is required.

Note: Data sent to a 3694 finance controller never requires a response.

Force-Data Function

Intrasystem communications does not buffer data; finance communications does. If you use finance communications, and your program issues a write operation and this is the first record in a group of records, the data is sent immediately. However, if your program sends subsequent records without specifying a function that closes the group of records (for example, end-of-group or invite functions or a read operation), the data may be buffered to be sent at a later time. Your program can use the force-data function to ensure that data is sent when the write operation is issued.

Number of Sessions

Intrasystem communications allows multiple sessions per device; finance communications allows only one session per device.

Read Operations

Intrasystem communications returns 0000 and 0001 return codes on read operations, finance communications does not. If you use finance communications; your program must receive an entire group of records. Therefore, your program can only receive the following return codes if the read operation is successful: 0003 (the last record in a group of records has been received) or 0007 (a group of records was received with a function management header as the first record).

Sense Data

Finance communications returns sense data to your program in an I/O feedback area that is accessible to your program. Sense data is returned for any operation that fails with an 8319 or 831A return code. However, intrasystem communications requires that the user issue an input operation to receive the sense data.

Write Operations

If you use intrasystem communications, your program receives an error indication on a write operation if the error indication is received before the write operation is issued. When finance communications receives an error indication before your program issues a write operation, your program either receives the error indication on the write operation, or is required to issue an input operation to receive the error indication if the session is invited.

Using Intrasystem Communications for Retail Communications

Retail communications allows programs on an AS/400 system to communicate with programs using SNA LU session type 0 protocol. The following considerations apply when you use intrasystem communications to test programs to be run using retail communications.

Sending and Receiving Data

Intrasystem communications is half-duplex, that is, your program can send or receive data, but cannot do both at the same time. Retail communications allows you to acquire sessions with retail controllers using a Systems Network Architecture (SNA) bind command that specifies a duplex protocol, that is, you can send and receive data at the same time. Therefore, when writing programs that use retail communications, you should note that it does not support the usual rules relating to when your program can send or receive data.

Note: Due to this major difference between intrasystem and retail communications, using intrasystem communications may not be the most effective way to test programs that use retail communications.

Confirm, End-of-Group, Invite, or Read Functions Specified on Write Operations with Data

Intrasystem communications allows your program to specify a confirm function on write operations with data; retail communications does not. However, retail communications supports a function similar to the confirm function. When your program issues a write operation with data, and the end-of-group function is also specified, the data is sent to the retail controller, and the write operation does not complete until a response is received from the controller.

If you specify invite or read functions on a write operation with data, however, the data is sent to the retail controller as if an end-of-group function was specified, but no response from the retail controller is required.

Detach Function

Intrasystem communications allows your program to send data when using the detach function, but retail communications does not. In addition, retail communications requires that any partially sent or partially received group of records be closed before a detach function is allowed. If any data or error indications have been received from the retail controller but have not yet been received by your program, the detach function fails, and the return code 8322 is returned to your program.

Evoke Function

When you use intrasystem communications and issue an evoke function, you must specify the program name. You may also use, for example, the SECURITY keyword and program initialization parameters, and specify read operations or functions such as the invite and function-management-header functions. If the retail controller program specifies program initialization parameters or security information on the evoke function, retail communications ignores this information. If the function-management-header function is specified on an evoke function, retail communications issues an 831E return code.

Force-Data Function

Retail communications buffers data; intrasystem communications does not. If your program issues a write operation without specifying a function that closes the group of records (for example, an end-of-group, force-data, or invite function or a read operation), the data may be buffered so that it can be sent at a later time. Specifying any of these functions ensures that all the data is sent.

Invite Function

If your program sends a group of records and then issues an invite function, intrasystem communications closes the group of records that is being sent. However, retail communications does not; the session is simply invited.

Note: Retail communications does close a group of records that your program is sending, however, when you issue an invite function and it is specified on a write operation with data.

Number of Sessions
Intrasystem communications allows multiple sessions per device; retail communications allows only one session per device.

Read Operations
Intrasystem communications returns a 0001 return code on a read operation; retail communications does not. A retail application program can receive one of the following return codes: 0000 (one record was received in a group of records); 0003 (the last record was received or the only record of a group of records was received); 0005 (the first record of a group of records was received with a function-management-header indication); or 0007 (a group of records was received with function-management-header data as the first record).

In addition, if you partially send a group of records on a write operation, and then issue a read operation, intrasystem communications closes the group of records. However, retail communications only closes a group of records on a read operation if it is specified on a write operation with data.

Sense Data
Retail communications returns sense data to your program in an I/O feedback area that is accessible to your program. Sense data is returned for any operation that fails with an 8319 or 831A return code. However, intrasystem communications requires that the user issue an input operation to receive the sense data.

Write Operations
Intrasystem communications requires your program to receive data when data is available to be received. You cannot also send data while you are receiving. However, retail communications allows your program to issue write operations at any time. The only case in which this is not true is when you receive an error indication from the remote program.

Using Intrasystem Communications for Systems Network Architecture Upline Facility

The **SNA upline facility (SNUF)** is the communications support that allows the AS/400 system to communicate with CICS/VS and IMS/VS application programs on a host system. The following considerations apply when you use intrasystem communications to test programs to be run using Systems Network Architecture upline facility (SNUF).

End-of-Group Function
Intrasystem communications issues a return code of 0003 or 0303 to indicate the end of a group of records. SNUF issues a major return code of 00 with minor return codes 01 or 03, or a major return code 03 with minor return codes 01 or 03, depending on the value specified for the batch parameter on either the Create Device Description (SNUF) (CRTDEVSNUF) command or on the ADDICFDEVE or OVRICFDEVE command.

Evoke Function
Intrasystem communications requires your program to issue an evoke function after acquiring a session to start the target program; SNUF does not.

Fail Function
Intrasystem communications can send and receive fail indications in addition to cancel and negative-response indications. If your program receives a fail function while in the receive state, intrasystem communications returns a 0302 return code. If your program receives a fail function while in send state, intrasystem communications returns a 0402 return code. SNUF cannot receive a fail indication from the host system, but can receive a cancel or negative-response indication, depending on the state of the transaction. If your program issues a fail function while in receive state, SNUF sends a negative-response indication to the host system. If your program issues a fail function while in the send state, SNUF sends a cancel indication to the host system.

Function-Management-Header Data
If a function-management-header indication is received with data, intrasystem communications inserts the characters FMH before the data that is received. SNUF does not insert these characters; instead, the return code indicates that the function-management-header indication is being received.

Number of Sessions
Intrasystem communications allows multiple sessions per device, SNUF allows only one session per device.

System Messages
Intrasystem communications does not send or receive system messages. However, SNUF checks the received data for system messages, and if they are received, SNUF issues a major return code 00 with the following minor return codes: 20, 21, 23, 25, 27, 28, 30, 31, 33, 35, 37, or 38.

Appendix D. Program Examples

This appendix provides sample programs to demonstrate how intrasystem communications is used. Program examples for the ILE C/400, C Set ++, COBOL*/400, and RPG*/400 programming languages are provided with explanations.

In the ILE C/400 program example, both the source and the target programs are provided to show how an application can be involved in an interactive inquiry with a single ICF session. A source program accepts inquiries from a display device and sends a request to a target program. The source program communicates with the display device through a display file, and with the target program through an ICF file.

In the COBOL/400 and RPG/400 program examples, both source and target programs are provided in an example that illustrates how an application can be involved in an interactive inquiry with two ICF sessions. A source program accepts inquiries from a display device and sends a request to one of two target programs. The source program communicates with the display device through a display file, and with the two target programs through a single ICF file. The purpose of the example is to show how a source program can communicate with two sessions from a single ICF file. From the viewpoint of each of the two target programs, the requester is the only session. Therefore, the target programs do not require any unique logic to support the two-session source.

For both source and target programs, the DDS source for the ICF file, program listings, and an explanation of the programs are provided.

Description of the Single-Session Inquiry Program Example

The following explanation describes the transaction between a source program and a target program, and applies to the ILE C/400 programs in this appendix.

The source program is started from a display station, and both the display and the ICF files are opened. The work station is implicitly acquired when the display file opens, but because the ICF file is created with ACQPGMDEV(*NONE), no ICF devices are acquired during open processing.

The ICF program device, ICF00, is explicitly acquired by the source program using the acquire operation. After the acquire, the source program starts the target program, through a CL program, using an evoke function. The target program also explicitly acquires a program device.

The source and target programs use their respective ICF files with program device ICF00, which is defined as the default program device from the acquire operation. Both programs will have only one session active at a time.

A customer inquiry prompt is displayed on the work station by the source program. The source program uses a write-with-invite function to send a number entered by the user to the target program, and then waits for data to be received from the target program. The target program reads the number sent by the source program, and searches a data file for a customer record based on the number received. If a record was found, it is sent to the source program; if not, a fail indication is sent. The source program ends by sending a detach request to the target program, issuing an end-of-session operation, and closing the files that were opened.

ILE C/400 Source Program for a Single-Session Inquiry

The following describes an ILE C/400 source program for a single-session inquiry.

Program Files: The ILE C/400 single-session source program uses the following files:

SRCICFF 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.

DDS Source: The DDS for the ICF file (SRCICFF) is illustrated in Figure D-1.

```
A*************************************************************
A*                                                           *
A*                   ICF FILE: SRCICFF                       *
A*      USED IN INTRASYSTEM ILE C/400 PROGRAM EXAMPLES       *
A*                                                           *
A*************************************************************
A                                     INDARA
A          R PGMSTR
A                                     EVOKE(CEXAMPLES/CTGTPGMCL)
A                                     SECURITY(2 *USER +
A                                              3 *USER)
A*
A          R CUST
A                                     INVITE
A            NUMBER         5
A*
A          R CINFO
A            CUSTNO         5
A            NAME          20
A            ADDR          20
A            CITY          20
A            STATE          2
A            ZIP            5
A            ACCBAL         6
A*
A          R SENDDETACH
A                                     DETACH
A*
A          R ENDSESSION
A                                     EOS
```

Figure D-1. DDS Source for a Single-Session Source Program Using SRCICFF

The DDS source file for the display file (DSPFIL) is illustrated in Figure D-2.

```
A*************************************************************
A*                                                           *
A*                  DISPLAY FILE: DSPFIL                     *
A*        USED IN INTRASYSTEM ILE C/400 PROGRAM EXAMPLES     *
A*                                                           *
A*************************************************************
A                                     INDARA
A                                     DSPSIZ(24 80 *DS3)
A                                     PRINT(QSYSPRT)
A                                     CA03(03 'END OF JOB')
A*
A          R HEADER
A                                     OVERLAY
A                                   2  4TIME
A                                      DSPATR(HI)
A                                   2 29'Customer Master Inquiry'
A                                      DSPATR(HI)
A                                   2 70DATE
A                                      EDTCDE(Y)
A                                      DSPATR(HI)
A*
A          R FOOT
A                                  24  4'F3 - End Job'
A                                      DSPATR(HI)
```

Figure D-2 (Part 1 of 2). DDS Source for a Single-Session Source Program Using DSPFIL

```
A*
A**********************************
A*  PROMPT CUSTOMER NUMBER SCREEN
A**********************************
A*
A          R PROMPT
A                                      OVERLAY
A                                    4  4'Enter Customer Number
A                                       (00001 - - 99999)'
A                                      DSPATR(HI)
A            CNUMBR         5A  I  4 42DSPATR(CS)
A                                      RANGE('00001' '99999')
A*
A********************************
A*  CUSTOMER INFORMATION SCREEN
A********************************
A*
A          R DTLSCR
A                                    6 25'Customer Information'
A            CUSTNO         5A  O  6 35DSPATR(HI)
A                                    8 25'Name'
A            NAME          20A  O  8 35DSPATR(HI)
A                                   10 25'Address'
A            ADDR          20A  O 10 35DSPATR(HI)
A                                   12 25'City'
A            CITY          20A  O 12 35DSPATR(HI)
A                                   14 25'State'
A            STATE          2A  O 14 35DSPATR(HI)
A                                   14 41'Zip Code'
A            ZIP            5A  O 14 50DSPATR(HI)
A                                   16 25'Account Balance'
A            ACCBAL         6A  O 16 42DSPATR(HI)
```

Figure D-2 (Part 2 of 2). DDS Source for a Single-Session Source Program Using DSPFIL

Configuration: The following command is needed to create the intrasystem communications device associated with the ICF file:

```
CRTDEVINTR  DEVD(INTRADEV)
  RMTLOCNAME(INTRARMT)  ONLINE(*NO)
  TEXT("THIS IS AN INTRASYSTEM DEVICE
  DESCRIPTION")
```

ICF File Creation and Program Device Entry Definition: The following command is needed to create the ICF file:

```
CRTICFF FILE(CEXAMPLES/SRCICFF)
  SRCFILE(CEXAMPLES/QDDSSRC) SRCMBR(SRCICFF)
  ACQPGMDEV(*NONE) MAXPGMDEV(1)
  TEXT("SOURCE ICF FILE FOR SINGLE
  SESSION PROGRAM")
```

It is not necessary to add a communications entry to the subsystem because the system automatically defines an entry for the device created above when the program is processed. However, the following command is an example of what you would use if you decided to add a communications entry:

```
ADDCMNE     SBSD(QCMN) DEV(INTRADEV)
```

Note: Subsystem QCMN should be stopped before ADDCMNE is entered, and then started again.

The following command is needed to define the program device entry:

```
OVRICFDEVE PGMDEV(ICF00) RMTLOCNAME(INTRARMT)
  FMTSLT(*PGM)
```

The following CL program could be used to run the source program:

```
PGM PARM(&RMT)
    DCL   VAR(&RMT)   TYPE(*CHAR) LEN(8)
    CHGJOB     OUTQ(CEXAMPLES/INTOUTQ)
               LOG(4 00 *SECLVL)
               LOGCLPGM(*YES)
    OVRICFDEVE PGMDEV(ICF00) RMTLOCNAME(&RMT)
               FMTSLT(*PGM)
    CALL CEXAMPLES/CSRCPGM
ENDPGM
```

The following CL program is used to start the target program evoked by the source program (which calls the program CTGTPGM shown in the example):

```
PGM
    ADDLIBLE   LIB(CEXAMPLES)
    OVRICFDEVE PGMDEV(ICF00)
    RMTLOCNAME(*REQUESTER)
    CALL PGM(CEXAMPLES/CTGTPGM)
ENDPGM
```

Program Explanation: The following explains the structure of the program example illustrated in Figure D-3 on page D-4. The ICF file used in the example 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.

All output operations to the ICF file in the example are done using the write statement with the record format name specified previously with a _Rformat function.

1 The display file descriptions (DSPFIL) are included in the program.

2 The ICF file descriptions (SRCICFF) are included in the program.

3 The routines are defined so the ILE C/400 compiler knows the type of value returned and the type of parameters passed, if any.

4 The ICF file is opened for record I/O with the separate indicator area option specified.

5 The display file is opened for record I/O with the separate indicator area option specified.

6 The separate indicator area is initialized and defined for DSPFIL. The variable dsp_indic is a 99 character array.

7 Program device ICF00 is explicitly acquired with the _Racquire function. A session is implicitly acquired for the work station when DSPFIL is opened.

8 The PGMSTR format name is specified and an evoke operation is performed with the _Rwrite function.

9 The program loops until either F3 is pressed from the work station, which sets an indicator in the display file's separate indicator area, or an error occurs in the transaction with the target program.

10 The get_cust_num() function is called to get a customer number from the user using DSPFIL.

11 The get_cust_info() function is called to send a customer number to the target program, and then to receive the customer information if it was found by the target program.

12 The get_cust_num() function gets a customer number from the user. The program displays the customer number inquiry, and reads the number.

13 The get_cust_info() function sends the customer number to the target program, and then receives the customer information sent by the target program. The customer number is copied into the output buffer of the ICF file for record format CUST defined in SRCICFF. CUST is specified on the _Rformat function, and a write is issued to the ICF file. The record format CINFO defined in SRCICFF to receive the customer information from the target is specified, and a read is issued to the ICF file.

The major return code is checked for a successful operation. If a 00 major return code is received, the customer information is displayed by calling the display_info() function, and control returns to main(). If a 00 major return code was not received, then a check is made to see if a 0302 or a 0402 return code was received, indicating that the target program issued a fail operation because it could not find customer information based on the number sent. If none of the above return codes is received, then an unexpected error has occurred and the program ends.

14 The display_info() function writes the customer information received from the target program to the work station.

15 The end_job() procedure is called when F3 is pressed to issue a detach operation to the target program, followed by an end-of-session operation to end the session.

16 The end_error() procedure is called if an error has occurred in trying to end the session. A detach is not issued since the target may have ended abnormally.

17 The send_eos() procedure issues the end-of-session operation to the ICF file.

18 The pos_resp() function checks for a 00 major return code in the display/ICF I/O feedback area.

19 The fail_rt_cd() function checks for a 0302 or a 0402 return code in the display/ICF I/O feedback area.

20 The get_access_to_fb() function accesses the display/ICF I/O feedback area by first accessing the common I/O feedback area to obtain an offset. The offset is added to the pointer to the common I/O feedback area to get access to the display/ICF I/O feedback area. The _Riofbk function returns a pointer to the common I/O feedback area.

```
                                        * * * * *  P R O L O G  * * * * *
Program name  . . . . . . . . . :   CSRCPGM
  Library name  . . . . . . . . :     CEXAMPLES
Source file . . . . . . . . . . :   QCSRC
  Library name  . . . . . . . . :     CEXAMPLES
Source member name  . . . . . . :   CSRCPGM
Text Description  . . . . . . . :   Source C program for Intra
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 . . . . . . . . . . :   90/08/21 10:23:36
Source description  . . . . . . :   Source C program for Intra
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      |#pragma mapinc("dspf/prompt", "cexamples/dspfil(prompt)", "both", "p z")                          |      1
   2      |#pragma mapinc("dspf/dtlscr", "cexamples/dspfil(dtlscr)", "both", "p z")                          |      2
           1
   3      |#include "dspf/prompt"                                                                          |      3
   4      |#include "dspf/dtlscr"                                                                          |      4
   5      |#pragma mapinc("icff/cust", "cexamples/srcicff(cust)", "output", "p z")                          |      5
   6      |#pragma mapinc("icff/cinfo", "cexamples/srcicff(cinfo)", "input", "p z")                         |      6
           2
   7      |#include "icff/cust"                                                                           |      7
   8      |#include "icff/cinfo"                                                                          |      8
   9      |/*--------------------------------------------------------------------*/                        |      9
  10      |/* SOURCE PROGRAM FOR INTRASYSTEM COMMUNICATIONS                     */                        |     10
  11      |/* This program reads a customer number from the display.  The number  */                        |     11
  12      |/* is sent to the target program, ctgtpgm, which searches a data file  */                        |     12
  13      |/* for information about the customer.  If the information is found,   */                        |     13
  14      |/* then the data is received and printed on the display.  If no infor- */                        |     14
  15      |/* mation is found for the given customer number, then the user is     */                        |     15
  16      |/* prompted for another number.                                       */                        |     16
  17      |/*--------------------------------------------------------------------*/                        |     17
  18      |                                                                                               |     18
  19      |#define NOERROR 0                  /* No error occured */                                      |     19
  20      |#define ERROR 1                    /* An error occured */                                      |     20
  21      |#define NOEND 0                    /* F3 wasn't pressed */                                     |     21
  22      |#define END 1                      /* F3 was pressed, signals end */                           |     22
  23      |#define OFF '0'                    /* Indicator off */                                         |     23
  24      |#define ON '1'                     /* Indicator on */                                          |     24
  25      |#include <stdio.h>                 /* Standard I/O header */                                   |     25
  26      |#include <recio.h>                 /* Record I/O header */                                     |     26
  27      |#include <stddef.h>                /* Standard definitions */                                  |     27
  28      |#include <stdlib.h>                /* General utilities */                                     |     28
  29      |#include <string.h>                /* String handling utilities */                             |     29
  30      |#include <xxfdbk.h>                /* Feedback area structures */                              |     30
  31      |                                                                                               |     31
  32      |                                                                                               |     32
```

Figure D-3 (Part 1 of 6). Source Program Example — CSRCPGM

```
  33       |CEXAMPLES_DSPFIL_PROMPT_both_t  prompt_dsp_i;                                                    |     33
  34       |CEXAMPLES_DSPFIL_DTLSCR_both_t  dtlscr_dsp_o;                                                    |     34
  35       |                                                                                                 |     35
  36       |CEXAMPLES_SRCICFF_CUST_o_t  cust_icf_o;                                                          |     36
  37       |CEXAMPLES_SRCICFF_CINFO_i_t  cinfo_icf_i;                                                        |     37
  38       |                                                                                                 |     38
  39       |_XXIOFB_T  *comm_fdbk;               /* Pointer to common I/O feedback */                        |     39
  40       |_XXIOFB_DSP_ICF_T  *dsp_icf_fdbk;    /* Pointer to dsp/ICF I/O feedback */                       |     40
  41       |_RFILE  *icffptr;                   /* Pointer to the ICF file */                                |     41
  42       |_RFILE  *dspfptr;                   /* Pointer to the display file */                            |     42
  43       |3                                                                                                |     43
  44       |int get_cust_num(char??(99??));                                                                  |     44
  45       |int get_cust_info(char??(99??));                                                                 |     45
  46       |int display_info(char??(99??));                                                                  |     46
  47       |int pos_resp(void);                                                                              |     47
  48       |int fail_rt_cd(void);                                                                            |     48
  49       |void end_job(void);                                                                              |     49
  50       |void end_error(void);                                                                            |     50
  51       |void send_eos(void);                                                                             |     51
  52       |void get_access_to_fb(void);                                                                     |     52
  53       |                                                                                                 |     53
Line  STMT                                                                                                  SEQNBR   INCNO
           *...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....+....9........
  54       |main()                                                                                           |     54
  55       |{                                                                                                |     55
  56       |   char dsp_indic??(99??);           /* Display separate indic area */                           |     56
  57       |                                                                                                 |     57
  58       |        /* Open the ICF file */                                                                  |     58
  59       |4                                                                                                |     59
  60       |   if ((icffptr=_Ropen("CEXAMPLES/SRCICFF", "ar+ indicators=y riofb=y"))                         |     60
  61    1  |                                                              == NULL) {                         |     61
  62    2  |       printf("ICF file failed to open.\n");                                                     |     62
  63    3  |       exit(ERROR);                                                                             |     63
  64       |   }                                                                                             |     64
  65       |                                                                                                 |     65
  66       |        /* Open the display file */                                                             |     66
  67       |5                                                                                                |     67
  68       |   if ((dspfptr=_Ropen("CEXAMPLES/DSPFIL", "ar+ indicators=y riofb=y"))                          |     68
  69    4  |                                                            == NULL) {                           |     69
  70    5  |      printf("Display file failed to open.\n");                                                  |     70
  71    6  |      _Rclose(icffptr);                                                                         |     71
  72    7  |      exit(ERROR);                                                                              |     72
  73       |   }                                                                                             |     73
  74       |                                                                                                 |     74
  75       |        /* Set up separate indicator area space */                                              |     75
  76       |6                                                                                                |     76
  77    8  |   memset(dsp_indic, OFF, 99);                                                                  |     77
  78    9  |   _Rindara(dspfptr, dsp_indic);                                                                |     78
  79       |                                                                                                 |     79
  80       |        /* Acquire a session */                                                                 |     80
  81       |7                                                                                                |     81
  82    10 |   _Racquire(icffptr, "ICF00");                                                                 |     82
  83    11 |   if (pos_resp() == ERROR)                                                                     |     83
  84    12 |       printf("Acquire failed.\n");                                                             |     84
  85       |   else {                                                                                        |     85
  86       |                                                                                                 |     86
  87       |        /* Evoke the other program */                                                           |     87
  88       |8                                                                                                |     88
  89    13 |       _Rformat(icffptr, "PGMSTR");                                                             |     89
  90    14 |       _Rwrite(icffptr, NULL, 0);                                                               |     90
  91       |                                                                                                 |     91
  92       |       /* Check if the evoke was successful */                                                  |     92
  93       |                                                                                                 |     93
  94    15 |       if (pos_resp() == ERROR) {                                                               |     94
  95    16 |           printf("Evoke failed.\n");                                                           |     95
  96    17 |           end_error();                                                                          |     96
  97    18 |           return(ERROR);                                                                        |     97
  98       |       }                                                                                         |     98
  99       |                                                                                                 |     99
```

Figure D-3 (Part 2 of 6). Source Program Example — CSRCPGM

```
100       |       /* While F3 isn't pressed get the customer number */                                  |   100
101       |       /* and display the customer record (if received) */                                  |   101
102       | 9                                                                                          |   102
103    19 |       while (dsp_indic??(2??) == OFF) {                                                    |   103
104       |                                                                                            |   104
105       |           /* read the customer number from the display */                                  |   105
106       | 10                                                                                         |   106
107    20 |           if (get_cust_num(dsp_indic) == NOEND)                                            |   107
Line  STMT                                                                                             SEQNBR   INCNO
          *...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....+....9........
108       |                                                                                            |   108
109       |               /* Attempt to get customer information from the */                          |   109
110       |               /* target and display the customer record if found */                        |   110
111       | 11                                                                                         |   111
112    21 |               if (get_cust_info(dsp_indic) == ERROR) {                                     |   112
113       |                                                                                            |   113
114       |                   /* An error occured in the transaction */                               |   114
115       |                                                                                            |   115
116    22 |                   end_error();                                                             |   116
117    23 |                   return(ERROR);                                                           |   117
118       |               }                                                                            |   118
119       |       }                                                                                    |   119
120       |                                                                                            |   120
121       |       /* F3 was pressed, end the session and the job */                                   |   121
122       |                                                                                            |   122
123    24 |       end_job();                                                                           |   123
124       |   }                                                                                        |   124
125    25 |   _Rclose(icffptr);                                                                        |   125
126    26 |   _Rclose(dspfptr);                                                                        |   126
127       |}                                                                                           |   127
128       |                                                                                            |   128
129       |                                                                                            |   129
130       |/*------------------------------------------------------------------------*/                |   130
131       |/* Get a customer number from the display.                                */                |   131
132       |/*------------------------------------------------------------------------*/                |   132
133       | 12                                                                                         |   133
134       |get_cust_num(char dsp_indic??(99??))                                                        |   134
135       |{                                                                                           |   135
136       |        /* Put out display and get information */                                         |   136
137       |                                                                                            |   137
138     1 |   _Rformat(dspfptr, "HEADER");                                                             |   138
139     2 |   _Rwrite(dspfptr, NULL, 0);                                                               |   139
140     3 |   _Rformat(dspfptr, "FOOT");                                                               |   140
141     4 |   _Rwrite(dspfptr, NULL, 0);                                                               |   141
142     5 |   _Rformat(dspfptr, "PROMPT");                                                             |   142
143     6 |   _Rwrite(dspfptr, NULL, 0);                                                               |   143
144     7 |   memset(dsp_indic, '0', 99);                                                              |   144
145     8 |   _Rreadn(dspfptr, &prompt_dsp_i, sizeof(prompt_dsp_i), __DFT);                            |   145
146       |                                                                                            |   146
147       |        /* Check if F3 (end the job) was pressed */                                        |   147
148       |                                                                                            |   148
149     9 |   if (dsp_indic??(2??) == OFF)                                                             |   149
150    10 |       return(NOEND);                                                                       |   150
151       |   else                                                                                     |   151
152    11 |       return(END);                                                                         |   152
153       |}                                                                                           |   153
154       |                                                                                            |   154
155       |                                                                                            |   155
156       |/*------------------------------------------------------------------------*/                |   156
157       |/* Evoke the target program, and then send the customer number to the   */                |   157
158       |/* target.  The target program should either send a customer record or  */                |   158
159       |/* a fail indication (customer not in data file).  The record is dis-   */                |   159
160       |/* played with a call to display_info.                                   */                |   160
161       |/*------------------------------------------------------------------------*/                |   161
```

Figure D-3 (Part 3 of 6). Source Program Example — CSRCPGM

```
Line  STMT                                                                                                  SEQNBR   INCNO
          *...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....+....9........
 162      | 13                                                                                              |   162
 163      |get_cust_info(char dsp_indic??(99??))                                                            |   163
 164      |{                                                                                                |   164
 165      |        /* Put number in ICF file and send it to the other program */                            |   165
 166      |                                                                                                 |   166
 167    1 |   strncpy(cust_icf_o.NUMBER, prompt_dsp_i.CNUMBR, 5);                                           |   167
 168    2 |   _Rformat(icffptr, "CUST");                                                                    |   168
 169    3 |   _Rwrite(icffptr, &cust_icf_o, sizeof(cust_icf_o));                                            |   169
 170      |                                                                                                 |   170
 171      |        /* Check if the number was sent successfully */                                          |   171
 172      |                                                                                                 |   172
 173    4 |   if (pos_resp() == ERROR) {                                                                    |   173
 174    5 |       printf("Unexpected error in transaction.\n");                                             |   174
 175    6 |       return(ERROR);                                                                            |   175
 176      |   }                                                                                             |   176
 177      |                                                                                                 |   177
 178      |        /* Read the customer information */                                                      |   178
 179      |                                                                                                 |   179
 180    7 |   _Rformat(icffptr, "CINFO");                                                                   |   180
 181    8 |   _Rreadn(icffptr, &cinfo_icf_i, sizeof(cinfo_icf_i), __DFT);                                   |   181
 182      |                                                                                                 |   182
 183      |        /* Check if the record was returned, and if so print it on the */                       |   183
 184      |        /* display, or the fail indicator may have been received from  */                       |   184
 185      |        /* the target program if the record wasn't found.             */                       |   185
 186      |                                                                                                 |   186
 187    9 |   if (pos_resp() == NOERROR)                                                                    |   187
 188      |                                                                                                 |   188
 189      |        /* Display the record, and then return to main with the value  */                       |   189
 190      |        /* returned from display_info.                                 */                       |   190
 191      |                                                                                                 |   191
 192   10 |       return(display_info(dsp_indic));                                                          |   192
 193      |   else                                                                                          |   193
 194   11 |       if (fail_rt_cd() == NOERROR)                                                              |   194
 195   12 |           return(NOERROR);                                                                      |   195
 196      |       else {                                                                                    |   196
 197   13 |           printf("Unexpected error in transaction.\n");                                         |   197
 198   14 |           return(ERROR);                                                                        |   198
 199      |       }                                                                                         |   199
 200      |}                                                                                                |   200
 201      |                                                                                                 |   201
 202      |                                                                                                 |   202
 203      |/*--------------------------------------------------------------------------*/                    |   203
 204      |/* Display the customer information.                                        */                    |   204
 205      |/*--------------------------------------------------------------------------*/                    |   205
 206      | 14                                                                                              |   206
 207      |display_info(char dsp_indic??(99??))                                                             |   207
 208      |{                                                                                                |   208
 209      |        /* Put out header and footing on display */                                              |   209
 210      |                                                                                                 |   210
 211    1 |   _Rformat(dspfptr, "HEADER");                                                                  |   211
 212    2 |   _Rwrite(dspfptr, NULL, 0);                                                                    |   212
 213    3 |   _Rformat(dspfptr, "FOOT");                                                                    |   213
 214    4 |   _Rwrite(dspfptr, NULL, 0);                                                                    |   214
 215      |                                                                                                 |   215
Line  STMT                                                                                                  SEQNBR   INCNO
          *...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....+....9........
 216      |        /* Put out display of customer information */                                            |   216
 217      |                                                                                                 |   217
 218    5 |   _Rformat(dspfptr, "DTLSCR");                                                                  |   218
 219    6 |   strncpy(dtlscr_dsp_o.CUSTNO, cinfo_icf_i.CUSTNO, 5);                                          |   219
 220    7 |   strncpy(dtlscr_dsp_o.NAME, cinfo_icf_i.NAME, 20);                                             |   220
 221    8 |   strncpy(dtlscr_dsp_o.ADDR, cinfo_icf_i.ADDR, 20);                                             |   221
 222    9 |   strncpy(dtlscr_dsp_o.CITY, cinfo_icf_i.CITY, 20);                                             |   222
 223   10 |   strncpy(dtlscr_dsp_o.STATE, cinfo_icf_i.STATE, 2);                                            |   223
 224   11 |   strncpy(dtlscr_dsp_o.ZIP, cinfo_icf_i.ZIP, 5);                                                |   224
 225   12 |   strncpy(dtlscr_dsp_o.ACCBAL, cinfo_icf_i.ACCBAL, 6);                                          |   225
 226   13 |   _Rwrite(dspfptr, &dtlscr_dsp_o, sizeof(dtlscr_dsp_o));                                        |   226
 227   14 |   memset(dsp_indic, '0', 99);                                                                   |   227
 228   15 |   _Rreadn(dspfptr, NULL, 0, __DFT);                                                             |   228
 229      |                                                                                                 |   229
```

Figure D-3 (Part 4 of 6). Source Program Example — CSRCPGM

```
 230        |        /* Check if F3 (end the job) was pressed */                                    |    230
 231        |                                                                                      |    231
 232    16  |   if (dsp_indic??(2??) == OFF)                                                       |    232
 233    17  |       return(NOEND);                                                                 |    233
 234        |   else                                                                               |    234
 235    18  |       return(END);                                                                   |    235
 236        |}                                                                                     |    236
 237        |                                                                                      |    237
 238        |                                                                                      |    238
 239        |/*--------------------------------------------------------------------*/            |    239
 240        |/* F3 was pressed, end the job.                                       */            |    240
 241        |/*--------------------------------------------------------------------*/            |    241
 242        | 15                                                                                   |    242
 243        |void end_job()                                                                        |    243
 244        |{                                                                                     |    244
 245        |        /* Issue a detach to the target program, then end the session */             |    245
 246        |                                                                                      |    246
 247     1  |   _Rformat(icffptr, "SENDDETACH");                                                   |    247
 248     2  |   _Rwrite(icffptr, NULL, 0);                                                         |    248
 249     3  |   send_eos();                                                                        |    249
 250        |}                                                                                     |    250
 251        |                                                                                      |    251
 252        |                                                                                      |    252
 253        |/*--------------------------------------------------------------------*/            |    253
 254        |/* Error, clean up.                                                   */            |    254
 255        |/*--------------------------------------------------------------------*/            |    255
 256        | 16                                                                                   |    256
 257        |void end_error()                                                                      |    257
 258        |{                                                                                     |    258
 259     1  |   send_eos();                                                                        |    259
 260     2  |   _Rclose(icffptr);                                                                  |    260
 261     3  |   _Rclose(dspfptr);                                                                  |    261
 262        |}                                                                                     |    262
 263        |                                                                                      |    263
 264        |                                                                                      |    264
 265        |/*--------------------------------------------------------------------*/            |    265
 266        |/* Issue an end of session operation.                                 */            |    266
 267        |/*--------------------------------------------------------------------*/            |    267
 268        | 17                                                                                   |    268
 269        |void send_eos()                                                                       |    269
Line  STMT                                                                                      SEQNBR   INCNO
        *...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....+....9........
 270        |{                                                                                     |    270
 271     1  |   _Rformat(icffptr, "ENDSESSION");                                                   |    271
 272     2  |   _Rwrite(icffptr, NULL, 0);                                                         |    272
 273        |}                                                                                     |    273
 274        |                                                                                      |    274
 275        |                                                                                      |    275
 276        |/*--------------------------------------------------------------------*/            |    276
 277        |/* Check the major return code for a successful operation - 00.      */            |    277
 278        |/*--------------------------------------------------------------------*/            |    278
 279        | 18                                                                                   |    279
 280        |pos_resp()                                                                            |    280
 281        |{                                                                                     |    281
 282     1  |   get_access_to_fb();                                                                |    282
 283     2  |   if (strncmp(dsp_icf_fdbk->major_ret_code, "00", 2) == 0)                           |    283
 284     3  |       return(NOERROR);                                                               |    284
 285        |   else                                                                               |    285
 286     4  |       return(ERROR);                                                                 |    286
 287        |}                                                                                     |    287
 288        |                                                                                      |    288
 289        |                                                                                      |    289
 290        |/*--------------------------------------------------------------------*/            |    290
 291        |/* Check the major/minor return code for a fail - 0302 or 0402.      */            |    291
 292        |/*--------------------------------------------------------------------*/            |    292
 293        | 19                                                                                   |    293
 294        |fail_rt_cd()                                                                          |    294
 295        |{                                                                                     |    295
 296     1  |   get_access_to_fb();                                                                |    296
 297        |   if ((strncmp(dsp_icf_fdbk->major_ret_code, "03", 2) == 0 ||                        |    297
 298        |        strncmp(dsp_icf_fdbk->major_ret_code, "04", 2) == 0) &&                       |    298
 299     2  |        strncmp(dsp_icf_fdbk->minor_ret_code, "02", 2) == 0)                          |    299
 300     3  |       return(NOERROR);                                                               |    300
 301        |   else                                                                               |    301
 302     4  |       return(ERROR);                                                                 |    302
 303        |}                                                                                     |    303
 304        |                                                                                      |    304
```

Figure D-3 (Part 5 of 6). Source Program Example — CSRCPGM

```
305       |                                                                              |  305
306       |/*--------------------------------------------------------------------------*/|  306
307       |/* Get access to the display/ICF feedback area.                            */|  307
308       |/*--------------------------------------------------------------------------*/|  308
309       | 20                                                                           |  309
310       |void get_access_to_fb()                                                       |  310
311       |{                                                                             |  311
312     1 |    comm_fdbk = _Riofbk(icffptr);                                             |  312
313       |    dsp_icf_fdbk = (_XXIOFB_DSP_ICF_T *)((char *)comm_fdbk +                  |  313
314     2 |                                       comm_fdbk->file_dep_fb_offset);        |  314
315       |}                                                                             |  315
                                  * * * * *  E N D  O F  S O U R C E  * * * * *
```

Figure D-3 (Part 6 of 6). Source Program Example — CSRCPGM

ILE C/400 Target Program for a Single-Session Inquiry

The following describes an ILE C/400 target program for a single-session inquiry.

Program Files: The ILE C/400 single-session target program uses the following files:

TGTICFF 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.

CUSMSTP A physical file that contains customer records to be sent to the source program.

CUSMSTL A logical file used with CUSMSTP to access the customer records.

DDS Source: The DDS for the ICF file (TGTICFF) is illustrated in Figure D-4.

```
A*************************************************************
A*                                                           *
A*                   ICF FILE: TGTICFF                       *
A*      USED BY INTRASYSTEM ILE C/400 PROGRAM EXAMPLES       *
A*                                                           *
A*************************************************************
A                                     INDARA
A          R CUST
A            NUMBER         5
A*
A          R CINFO
A                                     INVITE
A            CUSTNO         5
A            NAME          20
A            ADDR          20
A            CITY          20
A            STATE          2
A            ZIP            5
A            ACCBAL         6
A*
A          R NOCUST
A                                     FAIL
A*
A          R ENDSESSION
A                                     EOS
```

Figure D-4. DDS Source for a Single-Session Target Program Using TGTICFF

The DDS source for the database file (CUSMSTP) is illustrated in Figure D-5.

```
A*************************************************************
A*                                                           *
A*                PHYSICAL FILE: CUSMSTP                     *
A*          USED TO CONTAIN CUSTOMER RECORDS FOR             *
A*         INTRASYSTEM ILE C/400 PROGRAM EXAMPLES            *
A*                                                           *
A*************************************************************
A          R CUSREC                   TEXT('Customer record')
A            PCUST          5         TEXT('Customer number')
A            PNAME         20         TEXT('Customer name')
A            PADDR         20         TEXT('Customer address')
A            PCITY         20         TEXT('Customer city')
A            PSTATE         2         TEXT('Customer state')
A            PZIP           5         TEXT('Customer zip code')
A            PACCBL         6         TEXT('Accounts receivable
                                        balance')
```

Figure D-5. DDS Source for a Single-Session Source Program Using CUSMSTP

The DDS source for the logical file (CUSMSTL) is illustrated in Figure D-6.

```
A*************************************************************
A*                                                           *
A*               LOGICAL FILE: LGCMSTF FOR                   *
A*  CUSMSTP USED IN INTRASYSTEM ILE C/400 PROGRAM EXAMPLES   *
A*                                                           *
A*************************************************************
A                                     UNIQUE
A          R CUSREC                   PFILE(CUSMSTP)
A          K PCUST
```

Figure D-6. DDS Source for a Single-Session Target Program Using LGCMSTF

ICF File Creation and Program Device Entry Definition: The following command is needed to create the ICF file:

```
CRTICFF  FILE(CEXAMPLES/TGTICFF)
  SRCFILE(CEXAMPLES/QDDSSRC)
  SRCMBR(SRCICFF)
  ACQPGMDEV(*NONE)
  TEXT("TARGET ICF FILE FOR SINGLE
  SESSION PROGRAM")
```

The following command is needed to define the program device entry:

```
OVRICFDEVE PGMDEV(ICF00)
  RMTLOCNAME(*REQUESTER)
```

Program Explanation: The following explains the structure of the program example illustrated in Figure D-7 on page D-11. The ICF file used in the example is defined by the user, and uses externally described data formats (DDS). The reference letters in the example below correspond to those in the following program example.

1 The database logical file descriptions (CUSMSTL) are included in the program.

2 The ICF file descriptions (TGTICFF) are included in the program.

3 The routines are defined so the ILE C/400 compiler knows the type of value returned and the type of parameters passed, if any.

4 The ICF file is opened for record I/O with the separate indicator area option specified.

5 The database logical file is opened for record input. If an error occurs, the ICF file is closed and the program ends.

6 The ICF00 program device is explicitly acquired with the _Racquire function.

7 The program loops until either a detach is received, or an error occurs in the transaction with the source program.

8 The process_data() function receives a customer number from the source program, reads the database file using the number as the key, and either returns customer data to the source program or issues a fail operation to tell the other program that customer information could not be found for the given number.

9 The send_eos() procedure issues an end-of-session operation to the ICF file to end the session.

10 The pos_resp() checks for a 00 major return code. The return code is obtained by accessing the common I/O feedback area to get a pointer to the display/ICF I/O feedback area. A pointer to the common I/O feedback area is returned from the QXXIOFBK function. The offset to the display/ ICF I/O feedback area is added to the common I/O feedback area pointer to get a pointer to the display/ICF I/O feedback area.

```
                                        * * * * *  P R O L O G  * * * * *
Program name  . . . . . . . . . :   CTGTPGM
  Library name  . . . . . . . . :     CEXAMPLES
Source file . . . . . . . . . . :   QCSRC
  Library name  . . . . . . . . :     CEXAMPLES
Source member name  . . . . . . :   CTGTPGM
Text Description  . . . . . . . :   Target C program for Intra
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 . . . . . . . . . . :   90/09/11 08:52:00
Source description  . . . . . . :   Target C program for Intra
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      |#pragma mapinc("dbf", "cexamples/cusmstl(*all)", "input", "p z")                                 |      1
   2      |#pragma mapinc("icff/cust", "cexamples/tgticff(cust)", "input", "p z")                           |      2
   3      |#pragma mapinc("icff/cinfo", "cexamples/tgticff(cinfo)", "output", "p z")                        |      3
           1
   4      |#include "dbf"                                                                                 |      4
           2
   5      |#include "icff/cust"                                                                           |      5
   6      |#include "icff/cinfo"                                                                          |      6
   7      |/*--------------------------------------------------------------------*/                        |      7
   8      |/* TARGET PROGRAM FOR INTRASYSTEM COMMUNICATIONS                     */                        |      8
   9      |/* This program receives a customer number from the source program and  */                     |      9
  10      |/* searches a data base file for customer information.  If a record is  */                     |     10
  11      |/* found, it is sent to the source program, otherwise a fail indication */                     |     11
  12      |/* is sent.  This program waits for a detach from the source program    */                     |     12
  13      |/* to end.                                                            */                        |     13
  14      |/*--------------------------------------------------------------------*/                        |     14
  15      |                                                                                               |     15
  16      |#define NOERROR 0                  /* No error occurred */                                     |     16
  17      |#define ERROR 1                    /* An error occurred */                                     |     17
  18      |#include <stdio.h>                 /* Standard I/O header */                                   |     18
  19      |#include <recio.h>                 /* Record I/O header */                                     |     19
  20      |#include <stddef.h>                /* Standard definitions */                                  |     20
  21      |#include <stdlib.h>                /* General utilities */                                     |     21
  22      |#include <string.h>                /* String handling utilities */                             |     22
  23      |#include <xxfdbk.h>                /* Feedback area structures */                              |     23
  24      |                                                                                               |     24
  25      |                                                                                               |     25
  26      |CEXAMPLES_CUSMSTL_CUSREC_i_t  cusrec_dbf_i;                                                    |     26
  27      |                                                                                               |     27
  28      |CEXAMPLES_TGTICFF_CUST_i_t  cust_icf_i;                                                        |     28
  29      |CEXAMPLES_TGTICFF_CINFO_o_t  cinfo_icf_o;                                                      |     29
  30      |                                                                                               |     30
```

Figure D-7 (Part 1 of 3). Target Program Example — CTGTPGM

```
   31       |_XXIOFB_T  *comm_fdbk;                         /* Ptr to common I/O feedback */                |     31
   32       |_XXIOFB_DSP_ICF_T  *dsp_icf_fdbk;              /* Ptr to dsp/ICF I/O feedback */               |     32
   33       |_RFILE  *icffptr;                             /* Ptr to the ICF file */                       |     33
   34       |_RFILE  *dbfptr;                              /* Ptr to the database file */                  |     34
   35       | 3                                                                                            |     35
   36       |int process_data(void);                                                                       |     36
   37       |void send_eos(void);                                                                          |     37
   38       |int pos_resp(void);                                                                           |     38
   39       |                                                                                              |     39
   40       |main()                                                                                        |     40
   41       |{                                                                                             |     41
   42       |       /* Open the the ICF file */                                                           |     42
   43       | 4                                                                                            |     43
   44       |   if ((icffptr = _Ropen("CEXAMPLES/TGTICFF", "ar+ indicators=y riofb=y"))                    |     44
   45     1 |                                                          == NULL)                            |     45
   46     2 |       exit(ERROR);                                                                          |     46
   47       |                                                                                              |     47
   48       |       /* Open the the datbase file */                                                       |     48
   49       | 5                                                                                            |     49
   50     3 |   if ((dbfptr = _Ropen("CEXAMPLES/CUSMSTL", "rr riofb=y")) == NULL) {                       |     50
   51     4 |       _Rclose(icffptr);                                                                      |     51
   52     5 |       exit(ERROR);                                                                          |     52
   53       |   }                                                                                          |     53
 Line  STMT                                                                                                SEQNBR   INCNO
          *...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....+....9........
   54       |                                                                                              |     54
   55       |       /* Acquire a session */                                                               |     55
   56       | 6                                                                                            |     56
   57     6 |   _Racquire(icffptr, "ICF00");                                                               |     57
   58       |                                                                                              |     58
   59       |       /* Check that the acquire was successful and get the record */                        |     59
   60       |                                                                                              |     60
   61     7 |   if (pos_resp() == NOERROR) {                                                               |     61
   62       |                                                                                              |     62
   63       |       /* Keep processing numbers until detach or eos is received */                        |     63
   64       | 7                                                                                            |     64
   65     8 |       while (process_data() == NOERROR)                                                     |     65
   66     9 |           ;                                                                                  |     66
   67    10 |       send_eos();                                                                           |     67
   68       |   }                                                                                          |     68
   69       |                                                                                              |     69
   70       |       /* Close the ICF file */                                                              |     70
   71       |                                                                                              |     71
   72    11 |   _Rclose(icffptr);                                                                          |     72
   73       |}                                                                                             |     73
   74       |                                                                                              |     74
   75       |                                                                                              |     75
   76       |/*--------------------------------------------------------------------*/                      |     76
   77       |/* This routine will get a customer number from the source program and  */                    |     77
   78       |/* attempt to find the corresponding record in the physical file CUSMSTP*/                    |     78
   79       |/* (by using logical file CUSMSTL).  The file is searched using the     */                    |     79
   80       |/* customer number received as the key.  If the record was found, it    */                    |     80
   81       |/* is sent to the source program, if not a fail is sent.                */                    |     81
   82       |/*--------------------------------------------------------------------*/                      |     82
   83       | 8                                                                                            |     83
   84       |process_data()                                                                                |     84
   85       |{                                                                                             |     85
   86       |   _RIOFB_T  *rio_fdbk;                 /* Ptr to partial I/O feedback */                     |     86
   87       |                                                                                              |     87
   88     1 |   _Rformat(icffptr, "CUST");                                                                 |     88
   89     2 |   _Rreadn(icffptr, &cust_icf_i, sizeof(cust_icf_i), __DFT);                                  |     89
   90       |                                                                                              |     90
   91       |       /* Check if the read was successful, and if so find the */                            |     91
   92       |       /* record in the data file searching by key             */                            |     92
   93       |                                                                                              |     93
```

Figure D-7 (Part 2 of 3). Target Program Example — CTGTPGM

```
  94     3 |   if (pos_resp() == NOERROR) {                                                                |     94
  95       |                                                                                               |     95
  96       |        /* Read record from database file */                                                   |     96
  97       |                                                                                               |     97
  98       |       rio_fdbk = _Rreadk(dbfptr, &cusrec_dbf_i, sizeof(cusrec_dbf_i),                         |     98
  99     4 |                          __KEY_EQ, cust_icf_i.NUMBER, 5);                                     |     99
 100       |                                                                                               |    100
 101       |        /* Check to see if the record was found */                                             |    101
 102       |                                                                                               |    102
 103     5 |       if (rio_fdbk->num_bytes != 0) {                                                         |    103
 104       |                                                                                               |    104
 105       |        /* Send the customer information to the source program */                             |    105
 106       |                                                                                               |    106
 107     6 |           memcpy(cinfo_icf_o.CUSTNO, cusrec_dbf_i.PCUST, 5);                                  |    107
Line  STMT                                                                                                SEQNBR   INCNO
          *...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....+....9........
 108     7 |           memcpy(cinfo_icf_o.NAME, cusrec_dbf_i.PNAME, 20);                                   |    108
 109     8 |           memcpy(cinfo_icf_o.ADDR, cusrec_dbf_i.PADDR, 20);                                   |    109
 110     9 |           memcpy(cinfo_icf_o.CITY, cusrec_dbf_i.PCITY, 20);                                   |    110
 111    10 |           memcpy(cinfo_icf_o.STATE, cusrec_dbf_i.PSTATE, 2);                                  |    111
 112    11 |           memcpy(cinfo_icf_o.ZIP, cusrec_dbf_i.PZIP, 5);                                      |    112
 113    12 |           memcpy(cinfo_icf_o.ACCBAL, cusrec_dbf_i.PACCBL, 6);                                 |    113
 114    13 |           _Rformat(icffptr, "CINFO");                                                         |    114
 115    14 |           _Rwrite(icffptr, &cinfo_icf_o, sizeof(cinfo_icf_o));                               |    115
 116       |       }                                                                                       |    116
 117       |       else {                                                                                  |    117
 118       |                                                                                               |    118
 119       |        /* Customer record was not found, send a fail */                                      |    119
 120       |                                                                                               |    120
 121    15 |           _Rformat(icffptr, "NOCUST");                                                        |    121
 122    16 |           _Rwrite(icffptr, NULL, 0);                                                          |    122
 123       |       }                                                                                       |    123
 124       |                                                                                               |    124
 125       |        /* Check for successful return code */                                                |    125
 126       |                                                                                               |    126
 127    17 |       return(pos_resp());                                                                     |    127
 128       |   }                                                                                           |    128
 129       |        /* A detach was received, or a transaction error occurred. */                         |    129
 130       |        /* Return error to main so the program can end.           */                          |    130
 131       |                                                                                               |    131
 132    18 |   return(ERROR);                                                                              |    132
 133       |}                                                                                              |    133
 134       |                                                                                               |    134
 135       |                                                                                               |    135
 136       |/*-------------------------------------------------------------------------*/                   |    136
 137       |/* Issue an end of session operation.                                     */                   |    137
 138       |/*-------------------------------------------------------------------------*/                   |    138
 139       | 9                                                                                             |    139
 140       |void send_eos()                                                                                |    140
 141       |{                                                                                              |    141
 142     1 |   _Rformat(icffptr, "ENDSESSION");                                                            |    142
 143     2 |   _Rwrite(icffptr, NULL, 0);                                                                  |    143
 144       |}                                                                                              |    144
 145       |                                                                                               |    145
 146       |                                                                                               |    146
 147       |/*-------------------------------------------------------------------------*/                   |    147
 148       |/* Check the major return code for a successful operation - 00.         */                   |    148
 149       |/*-------------------------------------------------------------------------*/                   |    149
 150       | 10                                                                                            |    150
 151       |pos_resp()                                                                                     |    151
 152       |{                                                                                              |    152
 153     1 |   comm_fdbk = _Riofbk(icffptr);                                                               |    153
 154       |   dsp_icf_fdbk = (_XXIOFB_DSP_ICF_T *)((char *)comm_fdbk +                                    |    154
 155     2 |                                  comm_fdbk->file_dep_fb_offset);                              |    155
 156     3 |   if (strncmp(dsp_icf_fdbk->major_ret_code, "00", 2) == 0)                                    |    156
 157     4 |        return(NOERROR);                                                                       |    157
 158       |   else                                                                                        |    158
 159     5 |        return(ERROR);                                                                         |    159
 160       |}                                                                                              |    160
                                  * * * * *  E N D  O F  S O U R C E  * * * * *
```

Figure D-7 (Part 3 of 3). Target Program Example — CTGTPGM

Description of the Two-Session Inquiry Program Example

The following explanation describes the transaction between a source program and two target programs, and applies to the COBOL/400 and RPG/400 programs in this appendix.

The source program is started from a display station, and both the display and ICF files are opened. The work station is implicitly acquired when the display file opens, but because the ICF file is created with ACQPGMDEV(*NONE), no ICF devices are acquired during open processing.

The two ICF program devices, ICF00 and ICF01, must be explicitly acquired by the source program using the acquire operation. The source program then starts the two target programs using an evoke function.

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. When the target program is started, the ICF file is implicitly opened if you are using the RPG/400 language support. However, if you are using the COBOL/400 language support, you need to open the ICF file explicitly using the open operation. Because the file is created with the requesting program device specified on the ACQPGMDEV parameter, the requesting program device is acquired when the ICF file is opened.

The main menu, with a record format CIMENU, is written to the display station and the program waits for a request from the display station. Based on the request made from the display station, the source program uses a write-with-invite function to send an inquiry request to one of the target programs. The target program then sends a reply to the inquiry using a read operation. Finally, the source program sends a detach request and ends the session.

COBOL/400 Source Program for a Two-Session Inquiry

The following describes a COBOL/400 source program for a two-session inquiry.

Program Files: The COBOL/400 two-session source program uses the following files:

INTFIL 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 An AS/400 printer file used to print records, both sent and received, as well as major and minor ICF return codes.

DFILE An output file that is used to assist in problem analysis for non-recoverable session errors.

DDS Source: The DDS for the ICF file (INTFIL) is illustrated in Figure D-8.

```
5714PW1 R01M00  880301                SEU SOURCE LISTING                          10/14/87 17:20:41               PAGE 1
SOURCE FILE . . . . . .  QINTSRC/INTLIB
MEMBER  . . . . . .  INTFIL
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
     A*************************************************************
     A*                                                           *
     A*                      ICF FILE                             *
     A*         USED IN SOURCE TWO SESSION PROGRAM                *
     A*                                                           *
     A*************************************************************
     A                                     INDARA
     A                                     RCVFAIL(25 'RECEIVED FAIL')
     A                                     RCVTRNRND(90)
     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(&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
     A          R TIMER
     A                                     TIMER(000030)
```

Figure D-8. DDS Source for a Two-Session Source Program Using INTFIL

The DDS source file for the display file (DSPFIL) is illustrated in Figure D-9.

```
5714PW1 R01M00  880301                SEU SOURCE LISTING                          10/14/87 16:59:50                PAGE 1
SOURCE FILE . . . . . .  QINTSRC/INTLIB
MEMBER  . . . . . .  DSPFIL
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
     A*************************************************************
     A*                                                           *
     A*                     DISPLAY FILE                          *
     A*           USED IN SOURCE TWO SESSION PROGRAM              *
     A*                                                           *
     A*************************************************************
     A*  BEGINNING MENU
     A********************
     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. Order inquiry'
     A                                   5  3'2. Buyer inquiry'
     A                                  11  1'Option:'
     A            OPTION         1N  I 11  9VALUES('1' '2')
     A                                  19  5DFT('CMD KEY 1 - END ')
     A          R DTLMNU                   TEXT('   BUYER INQUIRY SCREEN 1')
     A                                   2  2DFT('ENTER BUYER')
     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('   BUYER INQUIRY SCR. #2')
     A                                   1  3DFT('BUYER DPT LAST ORD   & THIS   +
     A                                        $MTH1     &MTH2     $MTH3     THIS+
     A                                        YTD CNAME')
     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          5      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')
```

Figure D-9 (Part 1 of 2). DDS for Source Program Two-Session Inquiry Using DSPFIL

```
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 - BUYER 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          9Y 2  10 16EDTCDE(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('TARGET PROGRAM TIMED OUT. ENTE-
A                                      R 1 TO TRY AGAIN OR 2 TO END.')
A            TIMRSP         1   I 20 61
```

Figure D-9 (Part 2 of 2). DDS for Source Program Two-Session Inquiry Using DSPFIL

Configuration: The following command is needed to create the intrasystem communications device associated with the ICF file:

```
CRTDEVINTR  DEVD(INTRADEV)
  RMTLOCNAME(INTRARMT)
  ONLINE(*NO)
  TEXT("THIS IS AN INTRASYSTEM DEVICE
  DESCRIPTION")
```

ICF File Creation and Program Device Entry Definition: The command needed to create the ICF file is:

```
CRTICFF  FILE(INTLIB/INTFIL)
  SRCFILE(INTLIB/QINTSRC)
  SRCMBR(INTFIL)
  ACQPGMDEV(*NONE)  MAXPGMDEV(2)
  TEXT("SOURCE ICF FILE FOR TWO SESSION
  PROGRAM")
```

It is not necessary to add a communications entry to the subsystem because the system automatically defines an entry for the device created above when the program is processed. However, the following is an example of what you would use if you decided to add a communications entry:

```
ADDCMNE    SBSD(QCMN) DEV(INTRADEV)
```

Note: Subsystem QCMN should be stopped before ADDCMNE is entered, and then restarted again. The commands needed to define the two program device entries are:

```
OVRICFDEVE PGMDEV(ICF00)
  RMTLOCNAME(INTRARMT)
  FMTSLT(*RECID)
OVRICFDEVE PGMDEV(ICF01)
  RMTLOCNAME(INTRARMT)
  FMTSLT(*RECID)
```

The CL program that could be used to run the source program is:

```
CSDINTCL: PGM PARM(&RMT1 &RMT2)
              DCL   VAR(&RMT1)  TYPE(*CHAR)
                    LEN(8)
              DCL   VAR(&RMT2)  TYPE(*CHAR)
                    LEN(8)
              CHGJOB     OUTQ(INTLIB/INTOUTQ)
                         LOG(4 00 *SECLVL)
                         LOGCLPGM(*YES)
              OVRICFDEVE PGMDEV(ICF00)
                         RMTLOCNAME(&RMT1)
                         FMTSLT(*RECID)
              OVRICFDEVE PGMDEV(ICF01)
                         RMTLOCNAME(&RMT2)
                         FMTSLT(*RECID)
              CALL INTLIB/CSDINT
ENDCSDINTCL: ENDPGM
```

A CL program that could be used as the target program called by the source program (which calls the program CTDINT shown in the example) is:

```
CTDINTCL: PGM
               CHGJOB     OUTQ(INTLIB/INTOUTQ)
                          LOG(4 00 *SECLVL)
                          LOGCLPGM(*YES)
               ADDLIBLE INTLIB
               OVRICFDEVE PGMDEV(RQSDEV)
               RMTLOCNAME(*REQUESTER)
               CALL INTLIB/CTDINT
               RMVLIBLE INTLIB
               MONMSG     MSGID(CPF0000)
ENDCTDINTCL: ENDPGM
```

Program Explanation: The following explains the structure of the program example illustrated in Figure D-10 on page D-20. The ICF file used in the example 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.

All output operations to the ICF file in the example are done using the write statement with the record format name coded as an operand.

1 This section defines the ICF file (INTFIL) and the display file (DSPFIL) used in the program.

INTFIL is the ICF file used to send records to and receive records from each of the two target programs. INTFIL is established using the file-level keyword, INDARA, indicating that 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 INTFIL 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 function key pressed.

2 THE DSP-ERROR SECTION and CMN-ERROR SECTION define the error handling procedures for I/O errors on the DSPFIL and INTFIL. A DSPFIL I/O error causes the program to end, and an error message to be sent to the printer file. The section for INTFIL 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.

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 of nine has been exceeded. If it has, the program goes to section 19 and then ends. If the limit count is less than nine, one is added to the count and control passes to section 17 and then to section 3 to try to open the file.

5 The two 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 programs by passing control to section 16.

When control returns from section 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 section 6.

7 The MENU 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 function key 1, the two 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 section 6.

If the option is not 1, the Buyer Inquiry menu (DTLMNU) is written to the work station and control is passed to section 6.

8 The ITMIN routine is called when the user is requesting an item inquiry (record format ITMMNU). If function key 1 (CMD-KEY = 01) is pressed, control passes to section 19, and then to section 20, the two transactions end, and the program ends. If function key 2 is pressed, the inquiry request is canceled, the main menu (CMENU) is written to the work station, and the program returns to section 16.

The item number is read from the work station and then the request is sent to the target program on program device ICF01.

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 timer is issued for 30 seconds before the read operation. This is provided to allow the local program to have a time out when no response is received from the target program.

The read is an implied read-from-invited-program-devices because no record format is specified in the read statement.

If a fail indication is received (the item number requested was not found), the request is not validated and a new item inquiry menu (ITMMNU) is written to the display device.

Control goes to section 9 to process the item information based on the input data that was received, and the result is written to the display using format ITMSC2.

After returning from section 9, the program returns to section 6.

9 The routine ITMOUT is called when the target program responds to a request for an item record. If the returned response is a fail indication (checked in section 8), the request is invalidated 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 The routine ITMRTN is called to process the next user request. If function key 1 (CMD-KEY = 01) is pressed, the transactions and session are ended in section 19, and control goes to section 20 to end the program.

If function key 2 is pressed, the main menu (CMENU) is written to the work station. If function key 3 is pressed, the Item Inquiry menu is written to the work station, and the program returns to section 6. By pressing Enter, the profit and loss figures are calculated and written to the work station before returning control to section 6.

11 The PROFIT-LOSS routine calculates the profit and loss figures for the second display of the requested item number.

12 The DTLIN routine is called when a request is read from the Buyer Inquiry menu (DTLMNU). If function key 1 (CMD-KEY = 01) is pressed, the transactions and sessions are ended. If function key 2 (CMD-KEY = 02) is pressed, the main menu (CIMENU) is written to the work station and the program returns to section 6.

The buyer 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 section 14 to retrieve the buyer detail information.

Routine DTLRTN in section 14 is called to continue the buyer information processing.

The program then returns to section 6.

13 The routine DTLRTN is called from section 6, and handles the user's request following the display of the buyer information. Function key 1 ends the job, function key 2 displays the main menu (CMENU), and pressing Enter displays the Buyer Inquiry menu (DTLMNU). Control then returns to section 6.

14 The CUSTOMER-DETAIL 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 section 2 for unsuccessful I/O operations). A 0310 return code means the remote program has timed out. (The timer was issued on the write operation.) If no data was received (return codes of 03xx), the request is sent again to the remote program. Finally, if the data returns in the wrong format, control is passed to section 17.

The buyer 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.

15 The EVOKE routine builds the evoke requests to send to the remote programs. Because the DDS keyword for the record format only specifies the field identifiers with the record, this code moves the literal value CTDINTCL to the field PGMID, and INTLIB to the field LIB.

When the program start request is received at the remote program, INTLIB is searched for CTDINTCL and that program then starts. CTDINTCL is a CL program that contains CL statements, as illustrated on D-17.

16 The ERROR-RECOVERY routine ends the transactions and closes the files. The ERR-SW indicator is set again, and control returns to the calling routine.

17 The EXIT-FORMAT-ERR 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.

18 The DETACH routine issues the detach function to the ICF file for each of the two program devices. In the program using the user-supplied format, the write operation is issued using the record format name DETACH.

19 The END-JOB routine releases the program devices and close the files. The program ends.

```
5738CB1 V2R1M0  910524      IBM AS/400 COBOL/400                INTLIB/CSDINT         RCH38321  10/08/90 11:09:19     Page    1
Program  . . . . . . . . . . . . . :   CSDINT
  Library  . . . . . . . . . . . . :     INTLIB
Source file  . . . . . . . . . . . :   QINTSRC
  Library  . . . . . . . . . . . . :     INTLIB
Source member  . . . . . . . . . . :   CSDINT     10/08/90 11:08:48
Generation severity level  . . . . :   29
Text 'description' . . . . . . . . :   COBOL Source Intra Program Example
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
5738CB1 V2R1M0  910524            AS/400 COBOL/400 Source         INTLIB/CSDINT         RCH38321  10/08/90 11:09:19     Page    2
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  COPYNAME   CHG DATE
    1  000100 IDENTIFICATION DIVISION.                                                                  09/30/87
    2  000200 PROGRAM-ID.               CSDINT.                                                         11/15/88
       000300*****************************************************************                          09/30/87
       000400*  THIS PROGRAM ASSIGNS TWO  SESSIONS AS FOLLOWS:                *                          11/15/88
       000500*    'ICF00' TO INQUIRE ABOUT A BUYER'S PURCHASING STATUS        *                          11/15/88
       000600*            BEFORE AN ORDER IS ALLOWED.                         *                          11/15/88
       000700*    'ICF01' TO INQUIRE ABOUT THE AVAILABILITY OF AN ITEM        *                          11/15/88
       000800*            BEING ORDERED (ITEM 000001 THRU 999999).            *                          11/15/88
       000900*  A DISPLAY DEVICE IS USED TO ENTER THE REQUEST ( USING A       *                          09/30/87
       001000*  BUYER  AND  AN ITEM MENU ) THAT IS SENT TO THE TARGET.        *                          10/05/90
       001100*****************************************************************                          09/30/87
    3  001200 ENVIRONMENT DIVISION.                                                                     09/30/87
    4  001300 CONFIGURATION SECTION.                                                                    09/30/87
    5  001400 SOURCE-COMPUTER.          IBM-AS400.                                                      01/15/88
    6  001500 OBJECT-COMPUTER.          IBM-AS400.                                                      01/15/88
    7  001600 SPECIAL-NAMES.            I-O-FEEDBACK IS IO-FEEDBACK                                     09/30/87
    8  001700                           OPEN-FEEDBACK IS OPEN-FBA.                                      09/30/87
    9  001800 INPUT-OUTPUT SECTION.                                                                     09/30/87
   10  001900 FILE-CONTROL.                                                                             09/30/87
       002000* 1                                                                                        09/30/87
       002100*****************************************************************                          09/30/87
       002200*                                                               *                          09/30/87
       002300*                F I L E   S P E C I F I C A T I O N S          *                          09/30/87
       002400*                                                               *                          09/30/87
       002500*    INTFIL :   ICF FILE USED TO SEND A REQUEST TO ONE          *                          10/05/90
       002600*               OF TWO DIFFERENT TARGET PROGRAMS.  TWO          *                          10/05/90
       002700*               SESSIONS ARE ACTIVE AT THE SAME TIME.           *                          10/05/90
       002800*                                                               *                          09/30/87
       002900*    DSPFIL :   DISPLAY FILE USED TO ENTER A REQUEST TO BE       *                          09/30/87
       003000*               SENT TO A TARGET PROGRAM.                       *                          10/05/90
       003100*                                                               *                          09/30/87
       003200*****************************************************************                          09/30/87
   11  003300     SELECT INTFIL ASSIGN TO WORKSTATION-INTFIL-SI                                         11/21/88
   12  003400         ORGANIZATION IS TRANSACTION                                                       09/30/87
   13  003500         CONTROL-AREA IS TR-CTL-AREA                                                       09/30/87
   14  003600         FILE STATUS IS STATUS-IND MAJ-MIN.                                                09/30/87
   15  003700     SELECT DSPFIL ASSIGN TO WORKSTATION-DSPFIL                                            09/30/87
   16  003800         ORGANIZATION IS TRANSACTION                                                       09/30/87
   17  003900         CONTROL-AREA IS DISPLAY-FEEDBACK                                                  09/30/87
   18  004000         FILE STATUS IS STATUS-DSP.                                                        09/30/87
   19  004100     SELECT QPRINT ASSIGN TO PRINTER-QSYSPRT.                                              09/30/87
   20  004200 DATA DIVISION.                                                                            09/30/87
   21  004300 FILE SECTION.                                                                             09/30/87
   22  004400 FD  INTFIL                                                                                11/21/88
   23  004500     LABEL RECORDS ARE STANDARD.                                                           09/30/87
   24  004600 01  INTREC.                                                                              11/21/88
   25  004700     COPY DDS-ALL-FORMATS-I-O OF INTFIL.                                                   11/21/88
   26 +000001       05  INTFIL-RECORD PIC X(196).                                  <-ALL-FMTS
      +000002*  INPUT FORMAT:ITMRSP     FROM FILE INTFIL     OF LIBRARY INTLIB     <-ALL-FMTS
      +000003*                                                                    <-ALL-FMTS
   27 +000004       05  ITMRSP-I      REDEFINES INTFIL-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
```

Figure D-10 (Part 1 of 13). Source Program Example — CSDINT

```
5738CB1 V2R1M0  910524            AS/400 COBOL/400 Source         INTLIB/CSDINT         RCH38321  10/08/90 11:09:19     Page    3
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  COPYNAME   CHG DATE
   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* OUTPUT FORMAT:ITMRSP     FROM FILE INTFIL     OF LIBRARY INTLIB          <-ALL-FMTS
      +000024*                                                                          <-ALL-FMTS
   46 +000025       05  ITMRSP-O      REDEFINES INTFIL-RECORD.                        <-ALL-FMTS
   47 +000026           06 RECITM                PIC X(1).                               <-ALL-FMTS
   48 +000027           06 ITEMNO                PIC S9(6).                              <-ALL-FMTS
   49 +000028           06 DESC                  PIC X(30).                              <-ALL-FMTS
   50 +000029           06 QTYLST                PIC S9(7).                              <-ALL-FMTS
   51 +000030           06 QTYOH                 PIC S9(7).                              <-ALL-FMTS
   52 +000031           06 QTYOO                 PIC S9(7).                              <-ALL-FMTS
   53 +000032           06 QTYBO                 PIC S9(7).                              <-ALL-FMTS
   54 +000033           06 UNITQ                 PIC X(2).                               <-ALL-FMTS
   55 +000034           06 PR01                  PIC S9(5)V9(2).                         <-ALL-FMTS
   56 +000035           06 PR05                  PIC S9(7).                              <-ALL-FMTS
   57 +000036           06 UFRT                  PIC S9(3)V9(2).                         <-ALL-FMTS
   58 +000037           06 SLSTM                 PIC S9(7)V9(2).                         <-ALL-FMTS
   59 +000038           06 SLSTY                 PIC S9(9)V9(2).                         <-ALL-FMTS
   60 +000039           06 CSTTM                 PIC S9(7)V9(2).                         <-ALL-FMTS
   61 +000040           06 CSTTY                 PIC S9(9)V9(2).                         <-ALL-FMTS
   62 +000041           06 PRO                   PIC S9(3)V9(2).                         <-ALL-FMTS
   63 +000042           06 LOS                   PIC S9(7)V9(2).                         <-ALL-FMTS
   64 +000043           06 FILL1                 PIC X(56).                              <-ALL-FMTS
      +000044*  INPUT FORMAT:DTLRSP     FROM FILE INTFIL     OF LIBRARY INTLIB          <-ALL-FMTS
      +000045*                                                                          <-ALL-FMTS
   65 +000046       05  DTLRSP-I      REDEFINES INTFIL-RECORD.                        <-ALL-FMTS
   66 +000047           06 RECCUS                PIC X(1).                               <-ALL-FMTS
   67 +000048           06 CUSTNO                PIC S9(6).                              <-ALL-FMTS
   68 +000049           06 DNAME                 PIC X(30).                              <-ALL-FMTS
   69 +000050           06 DLSTOR                PIC S9(6).                              <-ALL-FMTS
   70 +000051           06 DSLSTM                PIC S9(9).                              <-ALL-FMTS
   71 +000052           06 DSPM01                PIC S9(9).                              <-ALL-FMTS
   72 +000053           06 DSPM02                PIC S9(9).                              <-ALL-FMTS
   73 +000054           06 DSPM03                PIC S9(9).                              <-ALL-FMTS
   74 +000055           06 DSTTYD                PIC S9(11).                             <-ALL-FMTS
   75 +000056           06 IDEPT                 PIC S9(3).                              <-ALL-FMTS
   76 +000057           06 FILL2                 PIC X(57).                              <-ALL-FMTS
      +000058* OUTPUT FORMAT:DTLRSP     FROM FILE INTFIL     OF LIBRARY INTLIB          <-ALL-FMTS
      +000059*                                                                          <-ALL-FMTS
   77 +000060       05  DTLRSP-O      REDEFINES INTFIL-RECORD.                        <-ALL-FMTS
   78 +000061           06 RECCUS                PIC X(1).                               <-ALL-FMTS
   79 +000062           06 CUSTNO                PIC S9(6).                              <-ALL-FMTS
   80 +000063           06 DNAME                 PIC X(30).                              <-ALL-FMTS
5738CB1 V2R1M0  910524            AS/400 COBOL/400 Source         INTLIB/CSDINT         RCH38321  10/08/90 11:09:19     Page    4
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  COPYNAME   CHG DATE
   81 +000064           06 DLSTOR                PIC S9(6).                              <-ALL-FMTS
   82 +000065           06 DSLSTM                PIC S9(9).                              <-ALL-FMTS
   83 +000066           06 DSPM01                PIC S9(9).                              <-ALL-FMTS
   84 +000067           06 DSPM02                PIC S9(9).                              <-ALL-FMTS
   85 +000068           06 DSPM03                PIC S9(9).                              <-ALL-FMTS
   86 +000069           06 DSTTYD                PIC S9(11).                             <-ALL-FMTS
   87 +000070           06 IDEPT                 PIC S9(3).                              <-ALL-FMTS
   88 +000071           06 FILL2                 PIC X(57).                              <-ALL-FMTS
```

Figure D-10 (Part 2 of 13). Source Program Example — CSDINT

```
     +000072*  INPUT FORMAT:DETACH     FROM FILE INTFIL     OF LIBRARY INTLIB            <-ALL-FMTS
     +000073*                                                                           <-ALL-FMTS
     +000074*      05  DETACH-I      REDEFINES INTFIL-RECORD.                          <-ALL-FMTS
     +000075* OUTPUT FORMAT:DETACH     FROM FILE INTFIL     OF LIBRARY INTLIB            <-ALL-FMTS
     +000076*                                                                           <-ALL-FMTS
     +000077*      05  DETACH-O      REDEFINES INTFIL-RECORD.                          <-ALL-FMTS
     +000078*  INPUT FORMAT:EOS        FROM FILE INTFIL     OF LIBRARY INTLIB            <-ALL-FMTS
     +000079*                                                                           <-ALL-FMTS
     +000080*      05  EOS-I         REDEFINES INTFIL-RECORD.                          <-ALL-FMTS
     +000081* OUTPUT FORMAT:EOS        FROM FILE INTFIL     OF LIBRARY INTLIB            <-ALL-FMTS
     +000082*                                                                           <-ALL-FMTS
     +000083*      05  EOS-O         REDEFINES INTFIL-RECORD.                          <-ALL-FMTS
     +000084*  INPUT FORMAT:EVKREQ     FROM FILE INTFIL     OF LIBRARY INTLIB            <-ALL-FMTS
     +000085*                                                                           <-ALL-FMTS
     +000086*      05  EVKREQ-I      REDEFINES INTFIL-RECORD.                          <-ALL-FMTS
     +000087* OUTPUT FORMAT:EVKREQ     FROM FILE INTFIL     OF LIBRARY INTLIB            <-ALL-FMTS
     +000088*                                                                           <-ALL-FMTS
  89 +000089       05  EVKREQ-O      REDEFINES INTFIL-RECORD.                          <-ALL-FMTS
  90 +000090           06 PGMID                PIC X(10).                              <-ALL-FMTS
  91 +000091           06 LIB                  PIC X(10).                              <-ALL-FMTS
     +000092*  INPUT FORMAT:ITMREQ     FROM FILE INTFIL     OF LIBRARY INTLIB            <-ALL-FMTS
     +000093*                                                                           <-ALL-FMTS
  92 +000094       05  ITMREQ-I      REDEFINES INTFIL-RECORD.                          <-ALL-FMTS
  93 +000095           06 ITEMNO               PIC S9(6).                              <-ALL-FMTS
     +000096* OUTPUT FORMAT:ITMREQ     FROM FILE INTFIL     OF LIBRARY INTLIB            <-ALL-FMTS
     +000097*                                                                           <-ALL-FMTS
  94 +000098       05  ITMREQ-O      REDEFINES INTFIL-RECORD.                          <-ALL-FMTS
  95 +000099           06 ITEMNO               PIC S9(6).                              <-ALL-FMTS
     +000100*  INPUT FORMAT:DTLREQ     FROM FILE INTFIL     OF LIBRARY INTLIB            <-ALL-FMTS
     +000101*                                                                           <-ALL-FMTS
  96 +000102       05  DTLREQ-I      REDEFINES INTFIL-RECORD.                          <-ALL-FMTS
  97 +000103           06 CUSTNO               PIC S9(6).                              <-ALL-FMTS
     +000104* OUTPUT FORMAT:DTLREQ     FROM FILE INTFIL     OF LIBRARY INTLIB            <-ALL-FMTS
     +000105*                                                                           <-ALL-FMTS
  98 +000106       05  DTLREQ-O      REDEFINES INTFIL-RECORD.                          <-ALL-FMTS
  99 +000107           06 CUSTNO               PIC S9(6).                              <-ALL-FMTS
     +000108*  INPUT FORMAT:TIMER      FROM FILE INTFIL     OF LIBRARY INTLIB            <-ALL-FMTS
     +000109*                                                                           <-ALL-FMTS
     +000110*      05  TIMER-I       REDEFINES INTFIL-RECORD.                          <-ALL-FMTS
     +000111* OUTPUT FORMAT:TIMER      FROM FILE INTFIL     OF LIBRARY INTLIB            <-ALL-FMTS
     +000112*                                                                           <-ALL-FMTS
     +000113*      05  TIMER-O       REDEFINES INTFIL-RECORD.                          <-ALL-FMTS
 100  004800 FD  DSPFIL                                                                        09/30/87
 101  004900     LABEL RECORDS ARE STANDARD.                                                    09/30/87
 102  005000 01  DSPREC.                                                                       09/30/87
 103  005100     COPY DDS-ALL-FORMATS-I-O OF DSPFIL.                                            09/30/87
 104 +000001       05  DSPFIL-RECORD PIC X(79).                                         <-ALL-FMTS
5738CB1 V2R1M0  910524             AS/400 COBOL/400 Source         INTLIB/CSDINT         RCH38321  10/08/90 11:09:19     Page    5
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  COPYNAME   CHG DATE
     +000002*  INPUT FORMAT:CIMENU     FROM FILE DSPFIL     OF LIBRARY INTLIB            <-ALL-FMTS
     +000003*                          MENU FOR INQUIRY                                 <-ALL-FMTS
 105 +000004       05  CIMENU-I      REDEFINES DSPFIL-RECORD.                          <-ALL-FMTS
 106 +000005           06 CIMENU-I-INDIC.                                              <-ALL-FMTS
 107 +000006                07 IN99            PIC 1  INDIC 99.                        <-ALL-FMTS
 108 +000007                07 IN98            PIC 1  INDIC 98.                        <-ALL-FMTS
 109 +000008                07 IN97            PIC 1  INDIC 97.                        <-ALL-FMTS
 110 +000009           06 OPTION               PIC X(1).                               <-ALL-FMTS
```

Figure D-10 (Part 3 of 13). Source Program Example — CSDINT

```
    +000010* OUTPUT FORMAT:CIMENU     FROM FILE DSPFIL     OF LIBRARY INTLIB            <-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 INTLIB            <-ALL-FMTS
    +000014*                             BUYER INQUIRY SCREEN 1                         <-ALL-FMTS
  111 +000015       05  DTLMNU-I      REDEFINES DSPFIL-RECORD.                          <-ALL-FMTS
  112 +000016           06 DTLMNU-I-INDIC.                                             <-ALL-FMTS
  113 +000017                07 IN99             PIC 1  INDIC 99.                     <-ALL-FMTS
  114 +000018                07 IN98             PIC 1  INDIC 98.                     <-ALL-FMTS
  115 +000019                07 IN97             PIC 1  INDIC 97.                     <-ALL-FMTS
  116 +000020           06 CUSTNO               PIC S9(6).                           <-ALL-FMTS
    +000021* OUTPUT FORMAT:DTLMNU     FROM FILE DSPFIL     OF LIBRARY INTLIB            <-ALL-FMTS
    +000022*                             BUYER INQUIRY SCREEN 1                         <-ALL-FMTS
    +000023*      05  DTLMNU-O      REDEFINES DSPFIL-RECORD.                          <-ALL-FMTS
    +000024*  INPUT FORMAT:DTLSCR     FROM FILE DSPFIL     OF LIBRARY INTLIB            <-ALL-FMTS
    +000025*                             BUYER INQUIRY SCR. #2                          <-ALL-FMTS
  117 +000026       05  DTLSCR-I      REDEFINES DSPFIL-RECORD.                          <-ALL-FMTS
  118 +000027           06 DTLSCR-I-INDIC.                                             <-ALL-FMTS
  119 +000028                07 IN99             PIC 1  INDIC 99.                     <-ALL-FMTS
  120 +000029                07 IN98             PIC 1  INDIC 98.                     <-ALL-FMTS
  121 +000030                07 IN97             PIC 1  INDIC 97.                     <-ALL-FMTS
    +000031* OUTPUT FORMAT:DTLSCR     FROM FILE DSPFIL     OF LIBRARY INTLIB            <-ALL-FMTS
    +000032*                             BUYER INQUIRY SCR. #2                          <-ALL-FMTS
  122 +000033       05  DTLSCR-O      REDEFINES DSPFIL-RECORD.                          <-ALL-FMTS
  123 +000034           06 CUSTN                PIC X(6).                            <-ALL-FMTS
  124 +000035           06 DEPT                 PIC S9(3).                           <-ALL-FMTS
  125 +000036           06 DLSTR                PIC S9(6).                           <-ALL-FMTS
  126 +000037           06 DSLSM                PIC S9(9).                           <-ALL-FMTS
  127 +000038           06 DSPM1                PIC S9(9).                           <-ALL-FMTS
  128 +000039           06 DSPM2                PIC S9(9).                           <-ALL-FMTS
  129 +000040           06 DSPM3                PIC S9(9).                           <-ALL-FMTS
  130 +000041           06 DSTYD                PIC S9(11).                          <-ALL-FMTS
  131 +000042           06 CNAME                PIC X(5).                            <-ALL-FMTS
    +000043*  INPUT FORMAT:ITMMNU     FROM FILE DSPFIL     OF LIBRARY INTLIB            <-ALL-FMTS
    +000044*                           ITEM INQUIRY SCREEN ONE                          <-ALL-FMTS
  132 +000045       05  ITMMNU-I      REDEFINES DSPFIL-RECORD.                          <-ALL-FMTS
  133 +000046           06 ITMMNU-I-INDIC.                                             <-ALL-FMTS
  134 +000047                07 IN99             PIC 1  INDIC 99.                     <-ALL-FMTS
  135 +000048                07 IN98             PIC 1  INDIC 98.                     <-ALL-FMTS
  136 +000049                07 IN97             PIC 1  INDIC 97.                     <-ALL-FMTS
  137 +000050           06 ITEMNO               PIC S9(6).                           <-ALL-FMTS
    +000051* OUTPUT FORMAT:ITMMNU     FROM FILE DSPFIL     OF LIBRARY INTLIB            <-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 INTLIB            <-ALL-FMTS
    +000055*                           ITEM INQUIRY SCREEN TWO                          <-ALL-FMTS
  138 +000056       05  ITMSC2-I      REDEFINES DSPFIL-RECORD.                          <-ALL-FMTS
5738CB1 V2R1M0  910524             AS/400 COBOL/400 Source         INTLIB/CSDINT         RCH38321  10/08/90 11:09:19     Page    6
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  COPYNAME   CHG DATE
  139 +000057           06 ITMSC2-I-INDIC.                                             <-ALL-FMTS
  140 +000058                07 IN99             PIC 1  INDIC 99.                     <-ALL-FMTS
  141 +000059                07 IN98             PIC 1  INDIC 98.                     <-ALL-FMTS
  142 +000060                07 IN97             PIC 1  INDIC 97.                     <-ALL-FMTS
    +000061* OUTPUT FORMAT:ITMSC2     FROM FILE DSPFIL     OF LIBRARY INTLIB            <-ALL-FMTS
    +000062*                           ITEM INQUIRY SCREEN TWO                          <-ALL-FMTS
  143 +000063       05  ITMSC2-O      REDEFINES DSPFIL-RECORD.                          <-ALL-FMTS
  144 +000064           06 DSC                  PIC X(30).                           <-ALL-FMTS
  145 +000065           06 QAVAIL                PIC S9(7).                           <-ALL-FMTS
  146 +000066           06 QTYH                  PIC S9(7).                           <-ALL-FMTS
  147 +000067           06 QTYO                  PIC S9(7).                           <-ALL-FMTS
  148 +000068           06 QTYB                  PIC S9(7).                           <-ALL-FMTS
  149 +000069           06 UNT                   PIC X(2).                            <-ALL-FMTS
  150 +000070           06 PR1                   PIC S9(5)V9(2).                      <-ALL-FMTS
  151 +000071           06 PR5                   PIC S9(7).                           <-ALL-FMTS
  152 +000072           06 UFR                   PIC S9(3)V9(2).                      <-ALL-FMTS
    +000073*  INPUT FORMAT:ITMSC3     FROM FILE DSPFIL     OF LIBRARY INTLIB            <-ALL-FMTS
    +000074*                           ITEM INQUIRY SCREEN 3                            <-ALL-FMTS
  153 +000075       05  ITMSC3-I      REDEFINES DSPFIL-RECORD.                          <-ALL-FMTS
  154 +000076           06 ITMSC3-I-INDIC.                                             <-ALL-FMTS
  155 +000077                07 IN99             PIC 1  INDIC 99.                     <-ALL-FMTS
  156 +000078                07 IN98             PIC 1  INDIC 98.                     <-ALL-FMTS
  157 +000079                07 IN97             PIC 1  INDIC 97.                     <-ALL-FMTS
```

Figure D-10 (Part 4 of 13). Source Program Example — CSDINT

```
     +000080* OUTPUT FORMAT:ITMSC3     FROM FILE DSPFIL     OF LIBRARY INTLIB            <-ALL-FMTS
     +000081*                          ITEM INQUIRY SCREEN 3                             <-ALL-FMTS
  158 +000082       05  ITMSC3-O      REDEFINES DSPFIL-RECORD.                          <-ALL-FMTS
  159 +000083           06 SLSM                  PIC S9(7)V9(2).                        <-ALL-FMTS
  160 +000084           06 SLSY                  PIC S9(9)V9(2).                        <-ALL-FMTS
  161 +000085           06 CSTM                  PIC S9(7)V9(2).                        <-ALL-FMTS
  162 +000086           06 CSTY                  PIC S9(9)V9(2).                        <-ALL-FMTS
  163 +000087           06 PROFIT                PIC S9(3)V9(2).                        <-ALL-FMTS
  164 +000088           06 LOSTS                 PIC S9(7)V9(2).                        <-ALL-FMTS
     +000089*  INPUT FORMAT:TIMOUT     FROM FILE DSPFIL     OF LIBRARY INTLIB            <-ALL-FMTS
     +000090*                          TIME OUT SCREEN                                   <-ALL-FMTS
  165 +000091       05  TIMOUT-I      REDEFINES DSPFIL-RECORD.                          <-ALL-FMTS
  166 +000092           06 TIMOUT-I-INDIC.                                              <-ALL-FMTS
  167 +000093                07 IN99             PIC 1  INDIC 99.                      <-ALL-FMTS
  168 +000094                07 IN98             PIC 1  INDIC 98.                      <-ALL-FMTS
  169 +000095                07 IN97             PIC 1  INDIC 97.                      <-ALL-FMTS
  170 +000096           06 TIMRSP                PIC X(1).                             <-ALL-FMTS
     +000097* OUTPUT FORMAT:TIMOUT     FROM FILE DSPFIL     OF LIBRARY INTLIB            <-ALL-FMTS
     +000098*                          TIME OUT SCREEN                                   <-ALL-FMTS
     +000099*      05  TIMOUT-O      REDEFINES DSPFIL-RECORD.                          <-ALL-FMTS
  171  005200 FD  QPRINT                                                                 09/30/87
  172  005300     LABEL RECORDS ARE OMITTED.                                             09/30/87
  173  005400 01  PRINTREC.                                                             01/14/88
  174  005500     05  RC                        PIC 9999.                               01/15/88
  175  005600     05  ERRMSG                    PIC X(128).                             01/14/88
  176  005700 WORKING-STORAGE SECTION.                                                   09/30/87
  177  005800 77  STATUS-IND                    PIC X(2).                               09/30/87
  178  005900 77  STATUS-DSP                    PIC X(2).                               09/30/87
  179  006000 77  MAJ-MIN-SAV                   PIC X(4).                               09/30/87
  180  006100 77  EOF-PFILE-SW                  PIC X VALUE "0".                        09/30/87
  181  006200 77  ERR-SW                        PIC X VALUE "0".                        09/30/87
  182  006300 77  INDON                         PIC 1 VALUE B"1".                       09/30/87
5738CB1 V2R1M0  910524            AS/400 COBOL/400 Source         INTLIB/CSDINT         RCH38321  10/08/90 11:09:19     Page    7
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  COPYNAME   CHG DATE
  183  006400 77  INDOFF                       PIC 1 VALUE B"0".                       09/30/87
  184  006500 77  OPEN-COUNT                   PIC 9(1) VALUE 0.                       09/30/87
  185  006600 77  LEN                          PIC 9(10)V9(5) COMP.                    09/30/87
  186  006700 77  PROFM                        PIC 9(7)V9(2) COMP-4.                   09/30/87
  187  006800 77  CMD2                         PIC X(31)                               09/30/87
  188  006900       VALUE "CPYF HEXDUMP *LIST PRTFMT(*HEX)".                          09/30/87
  189  007000 01  SUBKEY-VALUE.                                                         09/30/87
  190  007100     05  SUBKEY                    PIC 9(3) VALUE 0.                       09/30/87
  191  007200 01  TR-CTL-AREA.                                                          09/30/87
  192  007300     05  FILLER                    PIC X(2).                               09/30/87
  193  007400     05  PGM-DEV-NME               PIC X(10).                              09/30/87
  194  007500     05  RCD-FMT-NME               PIC X(10).                              09/30/87
  195  007600 01  INTF-INDIC-AREA.                                                      11/21/88
  196  007700     05  IN25                      PIC 1 INDIC 25.                         11/16/88
  197  007800         88 IN25-ON                     VALUE B"1".                       11/16/88
  198  007900         88 IN25-OFF                    VALUE B"0".                       11/16/88
  199  008000     05  IN90                      PIC 1 INDIC 90.                         11/16/88
  200  008100         88 IN90-ON                     VALUE B"1".                       11/16/88
  201  008200         88 IN90-OFF                    VALUE B"0".                       11/16/88
  202  008300 01  DSPF-INDIC-AREA.                                                     09/30/87
  203  008400     05  IN23                      PIC 1 INDIC 23.                         09/30/87
  204  008500         88 IN23-ON                   VALUE B"1".                         09/30/87
  205  008600         88 IN23-OFF                   VALUE B"0".                        09/30/87
  206  008700     05  IN97                      PIC 1 INDIC 97.                         09/30/87
  207  008800         88 IN97-ON                   VALUE B"1".                         09/30/87
  208  008900         88 IN97-OFF                   VALUE B"0".                        09/30/87
  209  009000     05  IN98                      PIC 1 INDIC 98.                         09/30/87
  210  009100         88 IN98-ON                     VALUE B"1".                       09/30/87
  211  009200         88 IN98-OFF                    VALUE B"0".                       09/30/87
  212  009300     05  IN99                      PIC 1 INDIC 99.                         09/30/87
  213  009400         88 IN99-ON                     VALUE B"1".                       09/30/87
  214  009500         88 IN99-OFF                    VALUE B"0".                       09/30/87
  215  009600 01  MAJ-MIN.                                                            09/30/87
  216  009700     05  MAJ                        PIC X(2).                               09/30/87
  217  009800     05  MIN                        PIC X(2).                               09/30/87
  218  009900 01  DISPLAY-FEEDBACK.                                                    09/30/87
  219  010000     05  CMD-KEY                   PIC X(2).                               09/30/87
  220  010100     05  FILLER                    PIC X(10).                              09/30/87
  221  010200     05  RCD-FMT                   PIC X(10).                              09/30/87
     010300*                                                                           11/18/88
  222  010400 PROCEDURE DIVISION.                                                      09/30/87
```

Figure D-10 (Part 5 of 13). Source Program Example — CSDINT

```
      010500 DECLARATIVES.                                                                  09/30/87
      010600* 2                                                                            10/14/87
      010700*************************************************************************      03/16/89
      010800*                                                                       *      03/16/89
      010900* AN ERROR ON THE DISPLAY FILE - DSPFIL - MAKES IT INACTIVE AND         *      03/16/89
      011000* THE JOB IS ENDED.                                                     *      03/16/89
      011100*                                                                       *      03/16/89
      011200*************************************************************************      03/16/89
      011300 DSP-ERROR SECTION.                                                            10/05/87
      011400     USE AFTER STANDARD ERROR PROCEDURE ON DSPFIL.                            10/05/87
      011500*                                                                            10/05/87
      011600 DSPFIL-EXCEPTION.                                                             10/05/87
  223 011700     MOVE "DISPLAY ERROR. JOB TERMINATED" TO ERRMSG.                          11/16/88
  224 011800     WRITE PRINTREC.                                                          11/16/88
5738CB1 V2R1M0  910524              AS/400 COBOL/400 Source         INTLIB/CSDINT         RCH38321  10/08/90 11:09:19     Page    8
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  COPYNAME   CHG DATE
  225 011900     CLOSE INTFIL DSPFIL QPRINT.                                              11/21/88
  226 012000     STOP RUN.                                                                11/16/88
      012100*                                                                            10/13/87
      012200*************************************************************************      03/16/89
      012300*                                                                       *      03/16/89
      012400* THIS SECTION HANDLES ERRORS ON THE INTFIL. A PERMANENT SESSION        *      03/16/89
      012500* ERROR WILL END THE JOB.                                               *      03/16/89
      012600*                                                                       *      03/16/89
      012700*************************************************************************      03/16/89
      012800 INT-ERROR SECTION.                                                            11/21/88
      012900     USE AFTER STANDARD ERROR PROCEDURE ON INTFIL.                            11/21/88
      013000 INTFIL-EXCEPTION.                                                             11/21/88
      013100*                                                                            10/05/90
      013200* RECOVERABLE SESSION ERROR.  CLOSE ICF FILE.                                10/05/90
  227 013300     IF MAJ = "83"                                                            09/30/87
  228 013400        MOVE MAJ-MIN TO RC                                                    01/14/88
  229 013500        MOVE "PROGRAM STARTED AGAIN DUE TO SESSION ERROR"                     09/30/87
      013600        TO ERRMSG                                                             01/14/88
  230 013700        WRITE PRINTREC                                                        09/30/87
  231 013800        MOVE "1" TO ERR-SW                                                    09/30/87
  232 013900        GO TO EXIT-DECLARATIVES.                                              09/30/87
      014000*                                                                           09/30/87
      014100* RECOVERABLE SESSION ERROR.  CLOSE ICF FILE.                                10/05/90
  233 014200     IF MAJ = "03"                                                            11/30/88
  234 014300        MOVE MAJ-MIN TO RC                                                    11/30/88
  235 014400        MOVE "ERROR IGNORED AND PROGRAM RESTARTED"                            11/30/88
      014500        TO ERRMSG                                                             11/30/88
  236 014600        WRITE PRINTREC                                                        11/30/88
  237 014700        MOVE "1" TO ERR-SW                                                    11/30/88
  238 014800        GO TO EXIT-DECLARATIVES.                                              11/30/88
      014900*                                                                           11/30/88
      015000*************************************************************************      03/16/89
      015100*                                                                       *      03/16/89
      015200* WHEN THERE IS A PERMANENT SESSION ERROR DETECTED,                     *      03/16/89
      015300* THE MAJOR-MINOR CODE IS PLACED INTO A DATABASE                        *      10/05/90
      015400* FILE AND THE FILE CAN BE PRINTED IN HEX USING COPYFILE.               *      03/16/89
      015500*                                                                       *      03/16/89
      015600*************************************************************************      03/16/89
      015700*                                                                           09/30/87
      015800 GETFBA.                                                                       09/30/87
  239 015900     MOVE MAJ-MIN TO RC.                                                      01/14/88
  240 016000     MOVE "PROGRAM TERMINATED DUE TO ERROR IN INTFIL FILE"                    11/21/88
      016100        TO ERRMSG.                                                            01/14/88
  241 016200     WRITE PRINTREC.                                                          09/30/87
  242 016300     CLOSE INTFIL DSPFIL QPRINT.                                              11/21/88
  243 016400     STOP RUN.                                                                09/30/87
      016500*                                                                           10/02/87
      016600 EXIT-DECLARATIVES.                                                            09/30/87
      016700     EXIT.                                                                    03/15/89
      016800*                                                                           09/30/87
  244 016900 END DECLARATIVES.                                                         09/30/87
      017000*                                                                           11/18/88
      017100 START-PROGRAM  SECTION.                                                       09/30/87
      017200*                                                                           09/30/87
      017300 START-PROGRAM-PARAGRAPH.                                                      09/30/87
```

Figure D-10 (Part 6 of 13). Source Program Example — CSDINT

```
5738CB1 V2R1M0  910524            AS/400 COBOL/400 Source         INTLIB/CSDINT         RCH38321  10/08/90 11:09:19     Page    9
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  COPYNAME   CHG DATE
      017400* 3                                                                                      09/30/87
  245 017500     OPEN I-O   INTFIL DSPFIL                                                           11/21/88
      017600          OUTPUT QPRINT.                                                                09/30/87
  246 017700     MOVE ZEROS TO INTF-INDIC-AREA.                                                     11/21/88
      017800*                                                                                       09/30/87
      017900*****************************************************************                    09/30/87
      018000*                                                               *                    03/16/89
      018100* THE FOLLOWING TEST IS TO ATTEMPT RECOVERY IF AN ERROR         *                    09/30/87
      018200* OCCURS WHEN OPENING THE ICF FILE.                             *                    10/05/90
      018300*                                                               *                    03/16/89
      018400*****************************************************************                    09/30/87
      018500* 4                                                                                    09/30/87
  247 018600     IF ERR-SW = "1"                                                                    09/30/87
  248 018700        THEN IF OPEN-COUNT IS = 9                                                       09/30/87
  249 018800                THEN PERFORM DETACH-ROUTINE THRU DETACH-EXIT                          09/30/87
  250 018900                     GO TO END-JOB                                                    09/30/87
      019000             ELSE                                                                    09/30/87
  251 019100                ADD 1 TO OPEN-COUNT                                                  09/30/87
  252 019200                PERFORM ERROR-RECOVERY                                               09/30/87
  253 019300                GO TO START-PROGRAM-PARAGRAPH                                        09/30/87
      019400     ELSE                                                                            09/30/87
  254 019500        MOVE 0 TO OPEN-COUNT.                                                        09/30/87
      019600*                                                                                    09/30/87
      019700*****************************************************************                    09/30/87
      019800*                                                               *                    09/30/87
      019900*   THE DISPLAY DEVICE IS IMPLICITLY ACQUIRED WHEN THE          *                    10/15/87
      020000*   FILE IS OPENED.                                             *                    09/30/87
      020100*                                                               *                    09/30/87
      020200*   ALL OF THE ICF PROGRAM DEVICES ARE EXPLICITLY ACQUIRED.     *                    10/05/90
      020300*                                                               *                    09/30/87
      020400*   THE TARGET PROGRAM IS EVOKED TWICE, ONCE FOR EACH SESSION   *                    10/05/90
      020500*   ACQUIRED, TO START TWO TRANSACTIONS.                        *                    10/05/90
      020600*                                                               *                    09/30/87
      020700*   THE MAIN INQUIRY MENU (CIMENU) IS WRITTEN TO THE USER'S     *                    09/30/87
      020800*   DISPLAY.                                                    *                    09/30/87
      020900*                                                               *                    09/30/87
      021000*   EVOKE TARGET PROGRAM "CTDINTCL" IN LIBRARY INTLIB.          *                    10/05/90
      021100*                                                               *                    03/16/89
      021200*****************************************************************                    09/30/87
      021300* 5                                                                                    09/30/87
  255 021400     ACQUIRE "ICF00   " FOR INTFIL.                                                   11/21/88
  256 021500     ACQUIRE "ICF01   " FOR INTFIL.                                                   11/21/88
  257 021600     PERFORM EVOKE-ROUTINE THRU EVOKE-EXIT.                                           09/30/87
      021700*                                                                                    09/30/87
  258 021800     WRITE DSPREC FORMAT IS "CIMENU"                                                  09/30/87
      021900        INDICATORS ARE DSPF-INDIC-AREA.                                               09/30/87
      022000*                                                                                    10/14/87
      022100*****************************************************************                    09/30/87
      022200*                                                               *                    09/30/87
      022300*                    DETERMINE USER'S REQUEST                   *                    09/30/87
      022400*                                                               *                    09/30/87
      022500*   A READ TO THE DISPLAY DEVICE IS ISSUED TO RECEIVE           *                    10/15/87
      022600*   THE USER'S REQUEST. THE TYPE OF REQUEST MADE IS BASED ON THE *                    10/13/87
      022700*   DISPLAY FORMAT CURRENTLY ON THE SCREEN.  THE RECORD FORMAT  *                    10/13/87
      022800*   NAME IS EXTRACTED FROM THE I/O FEEDBACK AREA FOR THE DISPLAY *                    10/13/87
5738CB1 V2R1M0  910524            AS/400 COBOL/400 Source         INTLIB/CSDINT         RCH38321  10/08/90 11:09:19     Page   10
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  COPYNAME   CHG DATE
      022900*   FILE AND USED TO DETERMINE WHAT ACTION SHOULD BE TAKEN NEXT. *                    10/13/87
      023000*                                                               *                    09/30/87
      023100*****************************************************************                    09/30/87
```

Figure D-10 (Part 7 of 13). Source Program Example — CSDINT

```
      023200* 6                                                                  09/30/87
      023300 READRQ.                                                             09/30/87
  259 023400     READ DSPFIL INDICATORS ARE DSPF-INDIC-AREA.                     09/30/87
  260 023500     IF RCD-FMT = "CIMENU"                                           09/30/87
  261 023600         PERFORM MENU-ROUTINE THRU MENU-EXIT                         09/30/87
  262 023700         GO TO READRQ.                                               09/30/87
  263 023800     IF RCD-FMT = "ITMMNU"                                           09/30/87
  264 023900         PERFORM ITMIN-ROUTINE THRU ITMIN-EXIT                       09/30/87
  265 024000         GO TO READRQ.                                               09/30/87
  266 024100     IF RCD-FMT = "ITMSC2"                                           09/30/87
  267 024200         PERFORM ITMRTN-ROUTINE THRU ITMRTN-EXIT                     09/30/87
  268 024300         GO TO READRQ.                                               09/30/87
  269 024400     IF RCD-FMT = "ITMSC3"                                           09/30/87
  270 024500         PERFORM ITMRTN-ROUTINE THRU ITMRTN-EXIT                     09/30/87
  271 024600         GO TO READRQ.                                               09/30/87
  272 024700     IF RCD-FMT = "DTLMNU"                                           09/30/87
  273 024800         PERFORM DTLIN-ROUTINE THRU DTLIN-EXIT                       09/30/87
  274 024900         GO TO READRQ.                                               09/30/87
  275 025000     IF RCD-FMT = "DTLSCR"                                           10/12/87
  276 025100         PERFORM DTLRTN-ROUTINE THRU DTLRTN-EXIT                     10/12/87
  277 025200         GO TO READRQ.                                               10/12/87
  278 025300     WRITE DSPREC FORMAT IS "CIMENU".                                09/30/87
      025400*                                                                    11/18/88
  279 025500     GO TO READRQ.                                                   09/30/87
      025600*                                                                    11/18/88
      025700*****************************************************************    09/30/87
      025800*                                                               *    09/30/87
      025900*                            MAIN MENU                          *    09/30/87
      026000*                                                               *    09/30/87
      026100*    THE MAIN MENU IS READ TO DETERMINE THE REQUEST ENTERED     *    10/12/87
      026200*    BY THE USER. IF CMD 1 (*IN99) IS PRESSED, THE PROGRAM      *    10/12/87
      026300*    IS ENDED.  IF OPTION = 1, AN ITEM INQUIRY MENU IS WRITTEN TO *  10/12/87
      026400*    TO SCREEN.  IF OPTION = 2, A CUSTOMER INQUIRY MENU IS      *    10/12/87
      026500*    WRITTEN TO THE SCREEN.                                     *    10/12/87
      026600*                                                               *    09/30/87
      026700*****************************************************************    09/30/87
      026800* 7                                                                  09/30/87
      026900 MENU-ROUTINE.                                                       09/30/87
  280 027000     IF CMD-KEY = "01"                                               09/30/87
  281 027100        PERFORM DETACH-ROUTINE THRU DETACH-EXIT                      09/30/87
  282 027200        GO TO END-JOB.                                               09/30/87
  283 027300     IF OPTION = "1"                                                 09/30/87
  284 027400        WRITE DSPREC FORMAT IS "ITMMNU"                              09/30/87
      027500     ELSE                                                            09/30/87
  285 027600        WRITE DSPREC FORMAT IS "DTLMNU".                             09/30/87
      027700 MENU-EXIT.                                                          09/30/87
      027800     EXIT.                                                           09/30/87
      027900*                                                                    11/18/88
      028000*****************************************************************    09/30/87
      028100*                                                               *    09/30/87
      028200*                        ITEM INQUIRY                           *    09/30/87
      028300*                                                               *    09/30/87
5738CB1 V2R1M0  910524             AS/400 COBOL/400 Source         INTLIB/CSDINT         RCH38321  10/08/90 11:09:19     Page    11
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  COPYNAME   CHG DATE
      028400*    THE ITEM NUMBER REQUESTED BY THE USER ON THE ITEM INQUIRY   *    09/30/87
      028500*    SCREEN IS CHECKED. THIS IS DETERMINED BY THE                *    09/30/87
      028600*    DISPLAY RECORD FORMAT BEING PROCESSED - IN THIS CASE ITMMNU. *   09/30/87
      028700*                                                               *    03/16/89
      028800*    IF CMD KEY 1 IS PRESSED, THE PROGRAM IS ENDED.  IF CMD KEY 2 *   10/13/87
      028900*    IS PRESSED, THE ITEM INQUIRY REQUEST IS CANCELED, AND THE   *    09/30/87
      029000*    MAIN MENU (CIMENU) IS WRITTEN TO THE SCREEN.                *    09/30/87
      029100*                                                               *    09/30/87
      029200*    IF AN ITEM NUMBER IS ENTERED, A ITEM INQUIRY REQUEST IS     *    09/30/87
      029300*    SENT TO THE APPROPRIATE TARGET PROGRAM.                     *    10/05/90
      029400*                                                               *    09/30/87
      029500*    A CHECK IS MADE FOR THREE CONDITIONS FOLLOWING THE READ.    *    10/14/87
      029600*    1) THE TARGET PROGRAM TIMED OUT, 2) NO DATA RECEIVED, AND   *    10/05/90
      029700*    3) DATA RETURNED IN AN UNEXPECTED RECORD FORMAT.            *    10/14/87
      029800*                                                               *    10/14/87
      029900*    IF THE TIMER RUNS OUT (MAJ-MIN = 0310) A MESSAGE            *    11/21/88
      030000*    IS WRITTEN TO THE SCREEN, ASKING TO TRY AGAIN OR END THE    *    10/14/87
      030100*    PROGRAM.                                                    *    10/14/87
      030200*                                                               *    11/21/88
```

Figure D-10 (Part 8 of 13). Source Program Example — CSDINT

```
      030300*    IF A RECEIVE FAIL INDICATION IS RECEIVED (IN-25 FLAG ON),     *                   11/21/88
      030400*    AFTER THE READ OPERATION TO THE PROGRAM DEVICE,               *                   11/21/88
      030500*    A FRESH ITEM MENU (ITMMNU) IS WRITTEN                         *                   11/21/88
      030600*    TO THE DISPLAY DEVICE.                                        *                   11/21/88
      030700*                                                                  *                   10/14/87
      030800*    IF NO DATA IS RECEIVED OR IF RECEIVE FAIL INDICATION          *                   03/16/89
      030900*    IS RECEIVED ( IN-25 FLAG IS ON), AFTER THE READ OPERATION     *                   11/21/88
      031000*    TO THE PROGRAM DEVICE, THE REQUEST IS SENT AGAIN              *                   10/05/90
      031100*    AND THE READ OPERATION IS ISSUED TO THE PROGRAM DEVICE.       *                   10/05/90
      031200*                                                                  *                   10/14/87
      031300*    IF THE RECORD RETURNS WITH THE WRONG RECORD FORMAT, THE       *                   10/14/87
      031400*    PROGRAM WILL GO TO EXIT-FORMAT-ERR ROUTINE.                   *                   10/14/87
      031500*                                                                  *                   10/14/87
      031600*****************************************************************                   09/30/87
      031700* 8                                                                                  09/30/87
  286 031800 ITMIN-ROUTINE.                                                                      09/30/87
  287 031900     IF CMD-KEY = "01"                                                               09/30/87
  288 032000        PERFORM DETACH-ROUTINE THRU DETACH-EXIT                                      10/12/87
  289 032100        GO TO END-JOB.                                                               10/12/87
  290 032200     IF CMD-KEY = "02"                                                               10/12/87
  291 032300        WRITE DSPREC FORMAT IS "CIMENU"                                              10/12/87
  292 032400        GO TO ITMIN-EXIT.                                                            10/12/87
      032500 XITMIN.                                                                            11/18/88
  293 032600     MOVE CORR ITMMNU-I TO ITMREQ-O.                                                 11/18/88
            *       ** CORRESPONDING items for statement 293:
            *       **     ITEMNO
            *       ** End of CORRESPONDING items for statement 293
  294 032700     MOVE "ICF01   " TO PGM-DEV-NME.                                                 09/30/87
  295 032800     MOVE ZEROS TO INTF-INDIC-AREA.                                                  11/21/88
  296 032900     WRITE INTREC FORMAT IS "ITMREQ"                                                 11/21/88
      033000        TERMINAL IS PGM-DEV-NME.                                                     09/30/87
      033100 TRY-AGAIN.                                                                         10/01/87
  297 033200     MOVE "ICF01   " TO PGM-DEV-NME.                                                 10/08/90
  298 033300     MOVE ZEROS TO INTF-INDIC-AREA.                                                  11/28/88
  299 033400     WRITE INTREC FORMAT IS "TIMER"                                                  11/28/88
      033500        TERMINAL IS PGM-DEV-NME.                                                     11/28/88
5738CB1 V2R1M0  910524            AS/400 COBOL/400 Source         INTLIB/CSDINT         RCH38321  10/08/90 11:09:19     Page    12
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  COPYNAME   CHG DATE
  300 033600     READ INTFIL                                                                     10/05/90
      033700        INDICATORS ARE INTF-INDIC-AREA.                                              10/05/90
  301 033800     IF MAJ-MIN = "0310"                                                             10/01/87
  302 033900        WRITE DSPREC FORMAT IS "TIMOUT"                                              09/30/87
  303 034000        READ DSPFIL INDICATORS ARE DSPF-INDIC-AREA                                   09/30/87
  304 034100        IF TIMRSP = "1" GO TO TRY-AGAIN END-IF                                       01/21/88
  306 034200        IF TIMRSP = "2" GO TO END-JOB END-IF.                                        01/21/88
  308 034300     IF IN25-ON                                                                      11/16/88
  309 034400        WRITE DSPREC FORMAT IS "ITMMNU"                                              11/16/88
  310 034500        GO TO ITMIN-EXIT.                                                            11/16/88
  311 034600     IF RCD-FMT-NME IS NOT EQUAL "ITMRSP" GO TO EXIT-FORMAT-ERR.                     11/28/88
  313 034700     PERFORM ITMOUT-ROUTINE THRU ITMOUT-EXIT.                                        09/30/87
      034800 ITMIN-EXIT.                                                                        09/30/87
      034900     EXIT.                                                                          09/30/87
      035000*                                                                                  11/18/88
      035100*****************************************************************                   09/30/87
      035200*                                                               *                   09/30/87
      035300*                    PROCESS ITEM INFORMATION                   *                   09/30/87
      035400*                                                               *                   09/30/87
      035500*    THE ITEM RECORD RECEIVED FROM THE TARGET PROGRAM AND THE   *                   09/30/87
      035600*    INFORMATION ABOUT THE ITEM IS PROCESSED AND DISPLAYED.     *                   09/30/87
      035700*    IF ITEMNO IS 0 OR LESS, IT IS AN INVALID REQUEST AND A FRESH *                 09/30/87
      035800*    ITEM MENU IS WRITTEN TO THE SCREEN.  IF THE REQUEST IS     *                   09/30/87
      035900*    VALID, VALUES ARE CALCULATED BASED ON THE INFORMATION      *                   09/30/87
      036000*    RECEIVED.                                                  *                   09/30/87
      036100*                                                               *                   09/30/87
      036200*****************************************************************                   09/30/87
```

Figure D-10 (Part 9 of 13). Source Program Example — CSDINT

```
      036300* 9                                                                 09/30/87
  314 036400 ITMOUT-ROUTINE.                                                    09/30/87
  315 036500     MOVE DESC OF ITMRSP-I TO DSC OF ITMSC2-O.                      11/18/88
  316 036600     MOVE QTYLST OF ITMRSP-I TO QAVAIL OF ITMSC2-O.                 11/18/88
  317 036700     MOVE QTYOO OF ITMRSP-I TO QTYO OF ITMSC2-O.                    11/18/88
  318 036800     MOVE QTYOH OF ITMRSP-I TO QTYH OF ITMSC2-O.                    11/18/88
  319 036900     MOVE QTYBO OF ITMRSP-I TO QTYB OF ITMSC2-O.                    11/18/88
  320 037000     MOVE UNITQ OF ITMRSP-I TO UNT OF ITMSC2-O.                     11/18/88
  321 037100     MOVE PR01 OF ITMRSP-I TO PR1 OF ITMSC2-O.                      11/18/88
  322 037200     MOVE PR05 OF ITMRSP-I TO PR5 OF ITMSC2-O.                      11/18/88
  323 037300     MOVE UFRT OF ITMRSP-I TO UFR OF ITMSC2-O.                      11/18/88
  324 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*****************************************************************  09/30/87
      038000*                                                               *  09/30/87
      038100*                   ADDITIONAL ITEM INFORMATION                 *  09/30/87
      038200*                                                               *  09/30/87
      038300*    ADDITIONAL ITEM INFORMATION IS PROCESSED AND THE RESULT     *  09/30/87
      038400*    DISPLAYED ON THE SCREEN WHEN A RESPONSE IS READ FROM THE    *  10/13/87
      038500*    DISPLAY STATION WITH AN ITEM SCREEN RECORD FORMAT.          *  10/14/87
      038600*                                                               *  03/16/89
      038700*    IF CMD KEY 1 IS PRESSED, THE PROGRAM IS ENDED.  IF CMD KEY 2 *  10/14/87
      038800*    IS PRESSED, THE ITEM INQUIRY IS ENDED, AND THE MAIN MENU    *  10/14/87
      038900*    (CIMENU) IS WRITTEN TO THE SCREEN.  IF CMD KEY 3 IS PRESSED, *  10/14/87
      039000*    THE ITEM INQUIRY MENU IS WRITTEN TO THE SCREEN.  BY PRESSING *  10/14/87
5738CB1 V2R1M0  910524             AS/400 COBOL/400 Source         INTLIB/CSDINT         RCH38321  10/08/90 11:09:19     Page    13
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  COPYNAME   CHG DATE
      039100*    ENTER WHEN SCREEN 2 IS DISPLAYED, MORE INFORMATION (PROFIT-  *  10/14/87
      039200*    LOSS) IS WRITTEN TO THE SCREEN.  IF SCREEN 3 IS DISPLAYED,   *  10/14/87
      039300*    PRESSING ENTER WILL CAUSE THE ITEM INQUIRY MENU TO BE        *  10/14/87
      039400*    WRITTEN TO THE SCREEN.                                       *  03/16/89
      039500*                                                                 *  03/16/89
      039600*****************************************************************  09/30/87
      039700* 10                                                               09/30/87
  325 039800 ITMRTN-ROUTINE.                                                    09/30/87
  326 039900     IF CMD-KEY = "01"                                              09/30/87
  327 040000        PERFORM DETACH-ROUTINE THRU DETACH-EXIT                     10/12/87
  328 040100        GO TO END-JOB.                                              10/12/87
  329 040200     IF CMD-KEY = "02"                                              09/30/87
  330 040300        WRITE DSPREC FORMAT IS "CIMENU"                             10/12/87
  331 040400        GO TO ITMRTN-EXIT.                                          10/12/87
  332 040500     IF CMD-KEY = "03"                                              09/30/87
  333 040600        WRITE DSPREC FORMAT IS "ITMMNU"                             10/12/87
  334 040700        GO TO ITMRTN-EXIT.                                          10/12/87
  335 040800     IF RCD-FMT = "ITMSC2"                                          10/12/87
  336 040900        PERFORM PROFIT-LOSS THRU PROFIT-LOSS-EXIT                   10/12/87
  337 041000        WRITE DSPREC FORMAT IS "ITMSC3"                             10/12/87
  338 041100        GO TO ITMRTN-EXIT.                                          10/12/87
  339 041200     WRITE DSPREC FORMAT IS "ITMMNU".                               10/12/87
      041300 ITMRTN-EXIT.                                                       09/30/87
      041400     EXIT.                                                          09/30/87
      041500*                                                                  09/30/87
      041600*****************************************************************  09/30/87
      041700*                                                               *  09/30/87
      041800*    PROFIT AND LOSS FIGURES ARE CALCULATED FOR THE ITEM NUMBER   *  10/13/87
      041900*    REQUESTED.  THESE ARE USED IN SCREEN TWO OF THE ITEM.        *  10/13/87
      042000*                                                               *  09/30/87
      042100*****************************************************************  09/30/87
      042200* 11                                                               03/17/89
  340 042300 PROFIT-LOSS.                                                       09/30/87
      042400*                                                                  03/17/89
  341 042500        SUBTRACT SLSTM OF ITMRSP-I FROM                             11/18/88
      042600           CSTTM OF ITMRSP-I GIVING PROFM.                          11/18/88
  342 042700        MULTIPLY PROFM BY 100 GIVING PROFM.                         09/30/87
  343 042800        IF SLSTM OF ITMRSP-I GREATER THAN 0                         11/18/88
  344 042900           DIVIDE PROFM BY SLSTM OF ITMRSP-I GIVING PROFM.          11/18/88
  345 043000        MULTIPLY QTYLST OF ITMRSP-I BY                              11/18/88
      043100           PR01 OF ITMRSP-I GIVING LOSTS.                           11/18/88
  346 043200        MOVE SLSTM OF ITMRSP-I TO SLSM OF ITMSC3-O.                 11/18/88
  347 043300        MOVE SLSTY  OF ITMRSP-I TO SLSY OF ITMSC3-O.                11/18/88
  348 043400        MOVE CSTTM  OF ITMRSP-I TO CSTM OF ITMSC3-O.                11/18/88
  349 043500        MOVE PROFM TO PROFIT OF ITMSC3-O.                           11/18/88
  350 043600        MOVE CSTTY OF ITMRSP-I TO CSTY OF ITMSC3-O.                 11/18/88
      043700 PROFIT-LOSS-EXIT.                                                  09/30/87
      043800     EXIT.                                                          09/30/87
      043900*                                                                  11/18/88
```

Figure D-10 (Part 10 of 13). Source Program Example — CSDINT

```
      044000*****************************************************************                   09/30/87
      044100*                                                               *                   09/30/87
      044200*                       CUSTOMER INQUIRY                        *                   09/30/87
      044300*                                                               *                   09/30/87
      044400*    THE REQUEST FROM THE CUSTOMER INQUIRY MENU IS PROCESSED.   *                   09/30/87
      044500*    IF CMD KEY 1 IS PRESSED, THE PROGRAM IS ENDED.  IF CMD KEY 2  *                   10/13/87
5738CB1 V2R1M0  910524             AS/400 COBOL/400 Source         INTLIB/CSDINT         RCH38321  10/08/90 11:09:19     Page    14
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  COPYNAME   CHG DATE
      044600*    IS PRESSED, THE MAIN MENU (CIMENU) IS WRITTEN TO THE SCREEN.  *                   10/14/87
      044700*                                                               *                   09/30/87
      044800*    IF A CUSTOMER NUMBER IS ENTERED, THE CUSTOMER INQUIRY      *                   09/30/87
      044900*    REQUEST IS SENT, THEN DTOUT-ROUTINE THRU DTOUT-EXIT EXECUTE.  *                   03/16/89
      045000*                                                               *                   09/30/87
      045100*****************************************************************                   09/30/87
      045200* 12                                                                                09/30/87
  351  045300 DTLIN-ROUTINE.                                                                    09/30/87
  352  045400     IF CMD-KEY = "01"                                                             09/30/87
  353  045500        PERFORM DETACH-ROUTINE THRU DETACH-EXIT                                    10/12/87
  354  045600        GO TO END-JOB.                                                             10/12/87
  355  045700     IF CMD-KEY = "02"                                                             10/12/87
  356  045800        WRITE DSPREC FORMAT IS "CIMENU"                                            10/12/87
  357  045900        GO TO DTLIN-EXIT.                                                          10/12/87
      046000 EVDTL.                                                                             09/30/87
  358  046100     MOVE "ICF00   " TO PGM-DEV-NME.                                               09/30/87
  359  046200     MOVE CORR DTLMNU-I TO DTLREQ-O.                                               11/18/88
           *       ** CORRESPONDING items for statement 359:
           *       **     CUSTNO
           *       ** End of CORRESPONDING items for statement 359
  360  046400     WRITE INTREC FORMAT IS "DTLREQ"                                               11/21/88
      046500        TERMINAL IS PGM-DEV-NME.                                                   09/30/87
  361  046600     PERFORM CUSTOMER-DETAIL THRU CUSTOMER-DETAIL-EXIT.                            10/08/90
      046700 DTLIN-EXIT.                                                                        09/30/87
      046800     EXIT.                                                                         09/30/87
      049000*                                                                                  10/14/87
      049100*****************************************************************                   10/12/87
      049200*                                                               *                   03/16/89
      049300*    THIS ROUTINE HANDLES THE USER'S REQUEST FOLLOWING THE      *                   03/16/89
      049400*    THE DISPLAY OF THE CUSTOMER INFORMATION.  CMD KEY 1 WILL   *                   03/16/89
      049500*    EXIT THE JOB, CMD KEY 2 WILL DISPLAY THE MAIN MENU, AND    *                   03/16/89
      049600*    "ENTER" WILL BRING UP THE CUSTOMER INQUIRY MENU.           *                   03/16/89
      049700*                                                               *                   03/16/89
      049800*****************************************************************                   10/12/87
      049900* 13                                                                                10/08/90
  362  050000 DTLRTN-ROUTINE.                                                                   10/12/87
  363  050100     IF CMD-KEY = "01"                                                             10/12/87
  364  050200        PERFORM DETACH-ROUTINE THRU DETACH-EXIT                                    10/12/87
  365  050300        GO TO END-JOB.                                                             10/12/87
  366  050400     IF CMD-KEY = "02"                                                             10/12/87
  367  050500        WRITE DSPREC FORMAT IS "CIMENU"                                            10/12/87
  368  050600        GO TO DTLRTN-EXIT.                                                         10/12/87
  369  050700     WRITE DSPREC FORMAT IS "DTLMNU".                                              10/12/87
      050800 DTLRTN-EXIT.                                                                       10/12/87
      050900     EXIT.                                                                         10/12/87
      051000*                                                                                  10/12/87
      051100*****************************************************************                   03/16/89
      051200*                                                               *                   03/16/89
      051300*    THE READ OPERATION TO THE PROGRAM DEVICE IS ISSUED.        *                   03/16/89
      051400*    A CHECK IS MADE FOR THREE CONDITIONS FOLLOWING THE READ.   *                   03/16/89
      051500*    1) THE TARGET PROGRAM TIMED OUT, 2) NO DATA RECEIVED, AND  *                   10/05/90
      051600*    3) DATA RETURNED IN AN UNEXPECTED RECORD FORMAT.           *                   03/16/89
      051700*                                                               *                   03/16/89
      051800*    IF THE TARGET PROGRAM TIMES OUT (MAJ-MIN = 0310), A MESSAGE   *                   03/16/89
      051900*    IS WRITTEN TO THE SCREEN, ASKING TO TRY AGAIN OR END THE   *                   03/16/89
5738CB1 V2R1M0  910524             AS/400 COBOL/400 Source         INTLIB/CSDINT         RCH38321  10/08/90 11:09:19     Page    15
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  COPYNAME   CHG DATE
      052000*    PROGRAM.                                                   *                   03/16/89
      052100*                                                               *                   03/16/89
      052200*    IF A RECEIVE FAIL INDICATION IS RECEIVED (IN-25 FLAG ON),  *                   03/16/89
      052300*    AFTER THE READ OPERATION TO THE PROGRAM DEVICE,            *                   03/16/89
      052400*    A FRESH CUSTOMER MENU (CIMENU) IS WRITTEN                  *                   03/16/89
      052500*    TO THE DISPLAY DEVICE.                                     *                   03/16/89
      052600*                                                               *                   03/16/89
      052700*    IF NO DATA IS RECEIVED AFTER THE READ OPERATION TO THE     *                   03/16/89
      052800*    PROGRAM DEVICE (MAJ-MIN = 03__) THE REQUEST IS SENT AGAIN  *                   10/05/90
      052900*    TO THE TARGET PROGRAM AND THE READ OPERATION IS ISSUED TO  *                   03/16/89
      053000*    THE ICF PROGRAM DEVICE.                                    *                   10/05/90
      053100*                                                               *                   03/16/89
      053200*    IF THE RECORD RETURNS WITH THE WRONG RECORD FORMAT, THE    *                   03/16/89
      053300*    PROGRAM WILL GO TO EXIT-FORMAT-ERR ROUTINE.                *                   03/16/89
      053400*                                                               *                   03/16/89
      053500*****************************************************************                   03/16/89
```

Figure D-10 (Part 11 of 13). Source Program Example — CSDINT

```
      053600*                                                                   03/17/89
      053700* 14                                                                10/08/90
  370 053800 CUSTOMER-DETAIL.                                                   09/30/87
  371 053900     MOVE ZEROS TO INTF-INDIC-AREA.                                 11/21/88
  372 054000     MOVE "ICF00   " TO PGM-DEV-NME.                                10/08/90
  373 054100     WRITE INTREC FORMAT IS "TIMER"                                 11/28/88
      054200        TERMINAL IS PGM-DEV-NME.                                    11/28/88
  374 054300     READ INTFIL                                                    10/05/90
      054400        INDICATORS ARE INTF-INDIC-AREA.                             10/05/90
  375 054500     IF MAJ-MIN = "0310"                                            10/01/87
  376 054600        WRITE DSPREC FORMAT IS "TIMOUT"                             09/30/87
  377 054700        READ DSPFIL INDICATORS ARE DSPF-INDIC-AREA                  09/30/87
  378 054800        IF TIMRSP = "1" GO TO CUSTOMER-DETAIL END-IF                01/21/88
  380 054900        IF TIMRSP = "2" GO TO END-JOB END-IF.                       01/21/88
  382 055000     IF IN25-ON                                                     11/16/88
  383 055100        WRITE DSPREC FORMAT IS "CIMENU"                             11/16/88
  384 055200        GO TO CUSTOMER-DETAIL-EXIT.                                 11/16/88
  385 055300     IF MAJ = "03"                                                  09/30/87
  386 055400        MOVE ZEROS TO INTF-INDIC-AREA                               11/21/88
  387 055500        WRITE INTREC FORMAT IS "DTLREQ"                             11/21/88
      055600           TERMINAL IS PGM-DEV-NME                                  09/30/87
  388 055700        GO TO CUSTOMER-DETAIL.                                      09/30/87
  389 055800     MOVE CUSTNO OF DTLRSP-I TO CUSTN OF DTLSCR-O.                  11/18/88
  390 055900     MOVE DNAME OF DTLRSP-I TO CNAME OF DTLSCR-O.                   03/15/89
  391 056000     MOVE DLSTOR OF DTLRSP-I TO DLSTR OF DTLSCR-O.                  11/18/88
  392 056100     MOVE DSLSTM OF DTLRSP-I TO DSLSM OF DTLSCR-O.                  11/18/88
  393 056200     MOVE DSPM01 OF DTLRSP-I TO DSPM1 OF DTLSCR-O.                  11/18/88
  394 056300     MOVE DSPM02 OF DTLRSP-I TO DSPM2 OF DTLSCR-O.                  11/18/88
  395 056400     MOVE DSTTYD OF DTLRSP-I TO DSTYD OF DTLSCR-O.                  11/18/88
  396 056500     MOVE IDEPT OF DTLRSP-I TO DEPT OF DTLSCR-O.                    11/18/88
  397 056600     WRITE DSPREC FORMAT IS "DTLSCR".                               10/12/87
      056700 CUSTOMER-DETAIL-EXIT.                                               09/30/87
      056800     EXIT.                                                          09/30/87
5738CB1 V2R1M0  910524             AS/400 COBOL/400 Source         INTLIB/CSDINT         RCH38321  10/08/90 11:09:19     Page    16
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  COPYNAME   CHG DATE
      056900/                                                                   09/30/87
      057000*************************************************************       03/16/89
      057100*                                                           *       03/16/89
      057200*    THE EVOKE-ROUTINE IS CALLED TO EVOKE THE TARGET PROGRAM.*       03/16/89
      057300*    THE SAME TARGET PROGRAM (INTLIB/CTDINTCL) IS EVOKED TWICE,*     03/16/89
      057400*    CREATING TWO DIFFERENT JOBS.  THE PROGRAM DEVICE IS USED TO*     03/16/89
      057500*    IDENTIFY THEM.                                         *       03/16/89
      057600*                                                           *       03/16/89
      057700*************************************************************       03/16/89
      057800*                                                                   03/17/89
      057900* 15                                                                10/08/90
  398 058000 EVOKE-ROUTINE.                                                     09/30/87
  399 058100        MOVE "CTDINTCL" TO PGMID OF EVKREQ-O.                       11/18/88
  400 058200        MOVE "INTLIB" TO LIB  OF EVKREQ-O.                          03/16/89
  401 058300        MOVE "ICF00   " TO PGM-DEV-NME                              09/30/87
  402 058400        WRITE INTREC FORMAT IS "EVKREQ"                             11/21/88
      058500           TERMINAL IS PGM-DEV-NME.                                 09/30/87
  403 058600        MOVE "ICF01   " TO PGM-DEV-NME                              09/30/87
  404 058700        WRITE INTREC FORMAT IS "EVKREQ"                             11/21/88
      058800           TERMINAL IS PGM-DEV-NME.                                 09/30/87
      058900 EVOKE-EXIT.                                                        09/30/87
      059000     EXIT.                                                          09/30/87
      059100*                                                                  09/30/87
      059200*************************************************************       03/16/89
      059300*                                                           *       03/16/89
      059400*    THE TRANSACTION AND SESSION ARE ENDED WITH EACH OF THE *       03/16/89
      059500*    TARGET PROGRAMS.                                       *       03/16/89
      059600*                                                           *       03/16/89
      059700*************************************************************       03/16/89
      059800* 16                                                                10/08/90
  405 059900 ERROR-RECOVERY.                                                    09/30/87
  406 060000     PERFORM DETACH-ROUTINE THRU DETACH-EXIT.                       09/30/87
  407 060100     CLOSE INTFIL DSPFIL                                            11/21/88
      060200           QPRINT.                                                  09/30/87
  408 060300     MOVE "0" TO ERR-SW.                                            09/30/87
      060400 ERROR-RECOVERY-EXIT.                                               09/30/87
      060500     EXIT.                                                          09/30/87
      060600*************************************************************       03/16/89
      060700*                                                           *       03/16/89
      060800* EXIT-FORMAT-ERR IS PERFORMED WHEN A READ TO INTFIL RETURNS WITH *  03/16/89
      060900* AN UNEXPECTED RCD-FMT-NME IN THE I-O-FEEDBACK AREA FOR INTFIL.  *  03/16/89
      061000* AN ERROR MESSAGE IS PRINTED AND THE PROGRAM ENDS.               *  03/16/89
      061100*                                                           *       03/16/89
      061200*************************************************************       03/16/89
```

Figure D-10 (Part 12 of 13). Source Program Example — CSDINT

```
      061300* 17                                                                  10/08/90
  409 061400 EXIT-FORMAT-ERR.                                                     10/01/87
  410 061500     MOVE MAJ-MIN TO RC.                                              01/14/88
  411 061600     MOVE "RECORD FORMAT IS INCORRECT ON READ           "             10/01/87
      061700        TO ERRMSG.                                                    01/14/88
  412 061800     WRITE PRINTREC.                                                  10/01/87
  413 061900     CLOSE  INTFIL DSPFIL QPRINT.                                     11/21/88
  414 062000     STOP RUN.                                                        10/01/87
      062100*                                                                     09/30/87
      062200*****************************************************************    03/16/89
      062300*                                                               *    03/16/89
5738CB1 V2R1M0  910524            AS/400 COBOL/400 Source         INTLIB/CSDINT         RCH38321  10/08/90 11:09:19     Page    17
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  COPYNAME   CHG DATE
      062400*    THIS ROUTINE IS CALLED TO END THE TRANSACTIONS WITH THE       *    03/16/89
      062500*    TARGET PROGRAMS.                                              *    03/16/89
      062600*                                                                  *    03/16/89
      062700*****************************************************************    03/16/89
      062800* 18                                                                  10/08/90
      062900 DETACH-ROUTINE.                                                       09/30/87
  415 063000     MOVE "ICF00   " TO PGM-DEV-NME                                   09/30/87
  416 063100     WRITE INTREC FORMAT IS "DETACH"                                  11/21/88
      063200        TERMINAL IS PGM-DEV-NME.                                      09/30/87
  417 063300     MOVE "ICF01   " TO PGM-DEV-NME                                   09/30/87
  418 063400     WRITE INTREC FORMAT IS "DETACH"                                  11/21/88
      063500        TERMINAL IS PGM-DEV-NME.                                      09/30/87
      063600 DETACH-EXIT.                                                          09/30/87
      063700     EXIT.                                                            09/30/87
      063800*                                                                     09/30/87
      063900*****************************************************************    03/16/89
      064000*                                                               *    03/16/89
      064100*    THIS ROUTINE IS CALLED TO RELEASE THE PROGRAM DEVICES, END    *    03/16/89
      064200*    THE SESSIONS, AND END THE PROGRAM.                            *    03/16/89
      064300*                                                                  *    03/16/89
      064400*****************************************************************    03/16/89
      064500* 19                                                                  10/08/90
      064600*                                                                     09/30/87
  419 064700 END-JOB.                                                          09/30/87
  420 064800     DROP "ICF00   " FROM INTFIL.                                     11/21/88
  421 064900     DROP "ICF01   " FROM INTFIL.                                     11/21/88
  422 065000     CLOSE  INTFIL DSPFIL QPRINT.                                     11/21/88
  423 065100     STOP RUN.                                                        09/30/87
                      * * * * *  E N D   O F   S O U R C E  * * * * *
5738CB1 V2R1M0  910524            AS/400 COBOL/400 Messages       INTLIB/CSDINT         RCH38321  10/08/90 11:09:19     Page    18
 STMT
*   25  MSGID: LBL0600  SEVERITY: 10  SEQNBR:  004700
       Message . . . . :   No INPUT fields found for format DETACH.
*   25  MSGID: LBL0600  SEVERITY: 10  SEQNBR:  004700
       Message . . . . :   No OUTPUT fields found for format DETACH.
*   25  MSGID: LBL0600  SEVERITY: 10  SEQNBR:  004700
       Message . . . . :   No INPUT fields found for format EOS.
*   25  MSGID: LBL0600  SEVERITY: 10  SEQNBR:  004700
       Message . . . . :   No OUTPUT fields found for format EOS.
*   25  MSGID: LBL0600  SEVERITY: 10  SEQNBR:  004700
       Message . . . . :   No INPUT fields found for format EVKREQ.
*   25  MSGID: LBL0600  SEVERITY: 10  SEQNBR:  004700
       Message . . . . :   No INPUT fields found for format TIMER.
*   25  MSGID: LBL0600  SEVERITY: 10  SEQNBR:  004700
       Message . . . . :   No OUTPUT fields found for format TIMER.
*  103  MSGID: LBL0600  SEVERITY: 10  SEQNBR:  005100
       Message . . . . :   No OUTPUT fields found for format CIMENU.
*  103  MSGID: LBL0600  SEVERITY: 10  SEQNBR:  005100
       Message . . . . :   No OUTPUT fields found for format DTLMNU.
*  103  MSGID: LBL0600  SEVERITY: 10  SEQNBR:  005100
       Message . . . . :   No OUTPUT fields found for format ITMMNU.
*  103  MSGID: LBL0600  SEVERITY: 10  SEQNBR:  005100
       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)
     11          0            11               0               0                0
 Source records read . . . . . . . . . :   629
 Copy records read . . . . . . . . . . :   212
 Copy members processed  . . . . . . . :   2
 Sequence errors . . . . . . . . . . . :   0
 Highest severity message issued . . :   10
  LBL0901 00  Program CSDINT created in library INTLIB.
                     * * * * *  E N D   O F   C O M P I L A T I O N  * * * * *
```

Figure D-10 (Part 13 of 13). Source Program Example — CSDINT

COBOL/400 Target Program for a Two-Session Inquiry

The following describes the COBOL/400 target program for a two-session inquiry.

Program Files: The COBOL/400 two-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 source program.

QPRINT An AS/400 printer file used to print records, both sent and received, as well as major and minor ICF return codes.

DDS Source: The DDS for the ICF file (CFILE) is illustrated in Figure D-11 on page D-34.

The DDS source for the database file (PFILE) is illustrated in Figure D-12 on page D-34.

ICF File Creation and Program Device Entry Definition: The command needed to create the ICF file is:

```
CRTICFF  FILE(INTLIB/CFILE)
  SRCFILE(INTLIB/QINTSRC)  SRCMBR(CFILE)
  ACQPGMDEV(RQSDEV)
  TEXT("TARGET ICF FILE FOR TWO 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 example illustrated in Figure D-13 on page D-35. The ICF file used in the example is defined by the user, and uses externally described data formats (DDS). The reference letters in the example below correspond to those in the following program example.

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 program.

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.

2 This section defines the error handling for the program. The CFILE routine first checks the major/minor return code to determine if the error is recoverable.

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 session associated with the target program is automatically acquired when the file is opened.

4 The RECEIVE-DATA routine reads data from the program device (CFILE) through a perform statement until a turnaround indication is received. The program then goes to section 5 to read the database file. When a turnaround indication is received, indicator 40 is set on, as defined by the RCVTRNRND DDS keyword in the DDS source file for the ICF file.

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, a fail indication is sent to the requesting program using a write operation with a fail operation.

If an error occurs on the write operation, control passes to section 2.

If no error occurs on the write, control goes back to section 4.

6 A read operation is issued to the program device.

If a detach indication is received, the program goes to section 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.

7 This routine is called to end the program.

The following message is written to the printer file:

```
CTDINT HAS COMPLETED NORMALLY
```

The files are closed. The program device is automatically released as a result of the close operation and the program ends.

```
5714PW1 R01M00  880301                SEU SOURCE LISTING                          10/14/87 17:20:35               PAGE 1
SOURCE FILE . . . . . .  QINTSRC/INTLIB
MEMBER  . . . . . .  PFILE
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
   A*************************************************************
   A*                                                           *
   A*                          ICF FILE                         *
   A*           USED IN TARGET TWO 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
   A            PARTDS        80
   A            FILL4         64
```

Figure D-11. DDS Source for a Two-Session Target Program Using CFILE

```
5714PW1 R01M00  880301                SEU SOURCE LISTING                          10/16/87 07:43:14               PAGE 1
SOURCE FILE . . . . . .  QINTSRC/INTLIB
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 D-12. DDS Source for a Two-Session Target Program Using PFILE

```
5738CB1 V2R1M0  910524         IBM AS/400 COBOL/400               INTLIB/CTDINT         RCH38321  10/05/90 16:14:43     Page    1
Program  . . . . . . . . . . . . . :   CTDINT
  Library  . . . . . . . . . . . . :     INTLIB
Source file  . . . . . . . . . . . :   QINTSRC
  Library  . . . . . . . . . . . . :     INTLIB
Source member  . . . . . . . . . . :   CTDINT     10/05/90 15:28:14
Generation severity level  . . . . :   29
Text 'description' . . . . . . . . :   COBOL Target Intra Example Program
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
5738CB1 V2R1M0  910524           AS/400 COBOL Source            INTLIB/CTDINT         RCH38321  10/05/90 16:14:43     Page    2
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  COPYNAME   CHG DATE
    1  000100 IDENTIFICATION DIVISION.                                                              10/01/87
    2  000200 PROGRAM-ID.               CTDINT.                                                     11/15/88
       000300*****************************************************************                     10/01/87
       000400*  THIS PROGRAM WILL HANDLE THE REQUEST FOR EITHER A BUYER         *                   11/16/88
       000500*  NUMBER OR AN ITEM NUMBER.  THIS IS ACCOMPLISHED BY MAKING       *                   10/01/87
       000600*  THE DATABASE FILE STRUCTURE (KEY LENGTH, KEY POSITION, RECORD   *                   10/05/90
       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           *                   10/05/90
       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                                                                                    03/17/89
   10  002500 FILE-CONTROL.                                                                         10/01/87
   11  002600     SELECT PFILE ASSIGN TO DATABASE-PFILE                                             10/01/87
   12  002700         ORGANIZATION IS INDEXED                                                       10/01/87
   13  002800         ACCESS IS RANDOM                                                              10/01/87
   14  002900         RECORD KEY IS EXTERNALLY-DESCRIBED-KEY                                        10/01/87
   15  003000            WITH DUPLICATES.                                                           10/01/87
   16  003100     SELECT CFILE ASSIGN TO WORKSTATION-CFILE-SI                                       10/01/87
   17  003200         ORGANIZATION IS TRANSACTION                                                   10/01/87
   18  003300         FILE STATUS IS STATUS-IND MAJ-MIN.                                            10/01/87
   19  003400     SELECT QPRINT ASSIGN TO PRINTER-QSYSPRT.                                          10/01/87
   20  003500 DATA DIVISION.                                                                        10/01/87
   21  003600 FILE SECTION.                                                                         10/01/87
   22  003700 FD  PFILE                                                                             10/01/87
   23  003800     LABEL RECORDS ARE STANDARD.                                                       10/01/87
   24  003900 01  PREC.                                                                            10/01/87
   25  004000     COPY DDS-ALL-FORMATS OF PFILE.                                                    10/01/87
   26 +000001       05  PFILE-RECORD PIC X(150).                                 <-ALL-FMTS
      +000002*    I-O FORMAT:DBREC      FROM FILE PFILE      OF LIBRARY INTLIB    <-ALL-FMTS
      +000003*                                                                   <-ALL-FMTS
```

Figure D-13 (Part 1 of 4). Target Program Example — CTDINT (User-Defined Formats)

```
      +000004*THE KEY DEFINITIONS FOR RECORD FORMAT  DBREC                              <-ALL-FMTS
      +000005*  NUMBER              NAME              RETRIEVAL     TYPE    ALTSEQ      <-ALL-FMTS
      +000006*   0001   DBSEQ                          ASCENDING      AN      NO        <-ALL-FMTS
   27 +000007       05  DBREC         REDEFINES PFILE-RECORD.                          <-ALL-FMTS
   28 +000008           06 RECCUS               PIC X(1).                             <-ALL-FMTS
   29 +000009           06 DBSEQ                PIC X(6).                             <-ALL-FMTS
   30 +000010           06 DBDATA               PIC X(130).                           <-ALL-FMTS
   31 +000011           06 DBFILL               PIC X(13).                            <-ALL-FMTS
   32  004100 FD  CFILE                                                                 10/01/87
   33  004200     LABEL RECORDS ARE STANDARD.                                           10/01/87
   34  004300 01  ICFREC.                                                              10/01/87
   35  004400     COPY DDS-ALL-FORMATS-I-O OF CFILE.                                    10/01/87
5738CB1 V2R1M0  910524        IBM AS/400 COBOL Source            INTLIB/CTDINT         RCH38321  10/05/90 16:14:43     Page    3
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  COPYNAME   CHG DATE
   36 +000001       05  CFILE-RECORD PIC X(150).                                      <-ALL-FMTS
      +000002*  INPUT FORMAT:SNDPART    FROM FILE CFILE      OF LIBRARY INTLIB        <-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 INTLIB        <-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 INTLIB        <-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 INTLIB        <-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
   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
   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* FAIL (FAIL) INDICATOR                                                   11/16/88
   73  006500     05  IN20                         PIC 1 INDIC 20.                   11/21/88
   74  006600         88 IN20-ON                          VALUE B"1".               11/16/88
   75  006700         88 IN20-OFF                         VALUE B"0".               11/16/88
      006800* RECEIVE FAIL (RCVFAL) INDICATOR                                         11/16/88
   76  006900     05  IN25                         PIC 1 INDIC 25.                   11/21/88
   77  007000         88 IN25-ON                          VALUE B"1".               11/16/88
   78  007100         88 IN25-OFF                         VALUE B"0".               11/16/88
      007200* RECEIVE TURNAROUND (RCVTRNRND) INDICATOR                                11/16/88
```

Figure D-13 (Part 2 of 4). Target Program Example — CTDINT (User-Defined Formats)

```
5738CB1 V2R1M0  910524           AS/400 COBOL Source           INTLIB/CTDINT        RCH38321  10/05/90 16:14:43     Page    4
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  COPYNAME   CHG DATE
   79  007300     05  IN40                       PIC 1 INDIC 40.                                  11/16/88
   80  007400         88 IN40-ON                       VALUE B"1".                                  11/16/88
   81  007500         88 IN40-OFF                      VALUE B"0".                                  11/16/88
       007600* RECEIVE DETACH (RCVDETACH) INDICATOR                                                 10/01/87
   82  007700     05  IN44                       PIC 1 INDIC 44.                                  10/01/87
   83  007800         88 IN44-ON                       VALUE B"1".                                  10/01/87
   84  007900         88 IN44-OFF                      VALUE B"0".                                  10/01/87
   85  008000 01  MAJ-MIN.                                                                         10/01/87
   86  008100     05  MAJ                        PIC X(2).                                        10/01/87
   87  008200     05  MIN                        PIC X(2).                                        10/01/87
5738CB1 V2R1M0  910524           AS/400 COBOL Source           INTLIB/CTDINT        RCH38321  10/05/90 16:14:43     Page    5
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  COPYNAME   CHG DATE
       008300/                                                                                    10/01/87
   88  008400 PROCEDURE DIVISION.                                                                 10/01/87
       008500 DECLARATIVES.                                                                       10/01/87
       008600 ERR-SECTION SECTION.                                                                10/01/87
       008700*****************************************************************                    10/01/87
       008800* 2                                                                                  03/17/89
       008900*                                                                                    10/01/87
       009000     USE AFTER STANDARD ERROR PROCEDURE ON CFILE.                                    10/01/87
       009100 CFILE-EXCEPTION.                                                                     10/01/87
       009200*****************************************************************                    10/01/87
       009300*                                                               *                    03/16/89
       009400* PRINT A MESSAGE SAYING CTDINT PROGRAM ENDED ABNORMALLY.       *                    03/16/89
       009500* CLOSE ALL THE FILES AND END THE PROGRAM. THIS ROUTINE IS CALLED *                    03/16/89
       009600* WHEN A NON-RECOVERABLE ERROR OCCURS IN ICF FILE.              *                    10/05/90
       009700*                                                               *                    03/16/89
       009800*****************************************************************                    10/01/87
       009900 GETFBA.                                                                              10/01/87
   89  010000     MOVE MAJ-MIN TO RC.                                                             01/14/88
   90  010100     MOVE "CTDINT HAS COMPLETED ABNORMALLY" TO ERRMSG.                               11/15/88
   91  010200     WRITE PRINTREC.                                                                 10/01/87
   92  010300     CLOSE PFILE                                                                     10/01/87
       010400           CFILE                                                                     10/01/87
       010500           QPRINT.                                                                   10/01/87
   93  010600     STOP RUN.                                                                       10/01/87
       010700*                                                                                    10/01/87
       010800 EXIT-DECLARATIVES.                                                                   10/01/87
       010900     EXIT.                                                                           10/01/87
       011000*                                                                                    10/01/87
   94  011100 END DECLARATIVES.                                                                   10/01/87
       011200*****************************************************************                    10/01/87
5738CB1 V2R1M0  910524           AS/400 COBOL Source           INTLIB/CTDINT        RCH38321  10/05/90 16:14:43     Page    6
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  COPYNAME   CHG DATE
       011300/                                                                                    10/01/87
       011400 START-PROGRAM  SECTION.                                                             10/01/87
       011500 START-PROGRAM-PARAGRAPH.                                                            10/01/87
       011600* 3                                                                                  03/17/89
   95  011700     OPEN OUTPUT QPRINT                                                              10/01/87
       011800          I-O   CFILE                                                               10/01/87
       011900          INPUT PFILE.                                                              10/01/87
       012000*****************************************************************                    10/01/87
       012100*                                                               *                    10/01/87
       012200*  READ THE REQUEST FROM THE SOURCE PROGRAM.  INDICATOR 40      *                    10/01/87
       012300*  INDICATES RCVTRNRND OCCURRED.  INDICATOR 44 INDICATES THAT   *                    10/05/90
       012400*  DETACH INDICATOR HAS BEEN RECEIVED FROM THE OTHER PROGRAM.   *                    03/16/89
       012500*                                                               *                    10/01/87
       012600*  THIS PROGRAM CHECKS FOR ERRORS ON EVERY ICF FILE             *                    10/05/90
       012700*  OPERATION.  A MAJOR CODE GREATER THAN 03 INDICATES AN ERROR. *                    03/16/89
       012800*                                                               *                    10/01/87
       012900*****************************************************************                    10/01/87
       013000* 4                                                                                  03/17/89
       013100 RECEIVE-DATA.                                                                       10/01/87
   96  013200     PERFORM READ-CFILE THRU READ-CFILE-EXIT.                                        10/01/87
   97  013300     GO TO SEND-DATA.                                                                10/05/90
   98  013400     GO TO RECEIVE-DATA.                                                             03/17/89
```

Figure D-13 (Part 3 of 4). Target Program Example — CTDINT (User-Defined Formats)

```
       013500*****************************************************************                   10/01/87
       013600*                                                               *                   10/01/87
       013700*  A REQUEST FROM THE SOURCE PROGRAM RESULTS IN READING A SINGLE   *                   10/01/87
       013800*  RECORD CONTAINING THE REQUESTED BUYER OR ORDER NUMBER.  THE     *                   03/16/89
       013900*  RESPONSE WILL BE RETURNED IN A SINGLE RECORD CONTAINING EITHER  *                   10/01/87
       014000*  THE ITEM OR BUYER INFORMATION, DEPENDING ON THE DATA BASE       *                   03/16/89
       014100*  CONTENT.                                                      *                   10/01/87
       014200*                                                               *                   10/01/87
       014300*  THE RESPONSE IS SENT TO THE SOURCE PROGRAM BY WRITING TO THE    *                   10/01/87
       014400*  PROGRAM DEVICE FILE USING FORMAT SNDPART.                     *                   10/15/87
       014500*                                                               *                   11/21/88
       014600*  WHEN THE REQUESTED BUYER OR ITEM NUMBER IS NOT FOUND,          *                   03/16/89
       014700*  OR WHEN A DISK ERROR OCCURRED WHILE READING THE DATABASE,      *                   03/16/89
       014800*  A FAIL INDICATION IS SENT TO THE SOURCE PROGRAM.               *                   03/16/89
       014900*                                                               *                   11/21/88
       015000*****************************************************************                   10/01/87
       015100*                                                                                   10/01/87
       015200* 5                                                                                 03/17/89
       015300 SEND-DATA.                                                                         10/01/87
    99 015400     MOVE RECID2 OF RCVPART-I TO DBSEQ.                                            10/01/87
   100 015500     READ PFILE INVALID KEY                                                        11/16/88
   101 015600                 SET IN20-ON TO TRUE.                                              11/16/88
   102 015700     MOVE RECCUS TO RECTYP OF SNDPART-O.                                           10/01/87
   103 015800     MOVE DBSEQ TO ITEMNO OF SNDPART-O.                                            10/01/87
   104 015900     MOVE DBDATA TO EDATA OF SNDPART-O                                             10/01/87
   105 016000     WRITE ICFREC FROM PREC FORMAT IS "SNDPART"                                    10/01/87
       016100        INDICATORS ARE CMNF-INDIC-AREA.                                           10/01/87
   106 016200     GO TO RECEIVE-DATA.                                                           10/01/87
       016300*****************************************************************                   10/01/87
       016400*                                                               *                   10/01/87
       016500* THIS ROUTINE ISSUES READ OPERATIONS TO THE PROGRAM DEVICE.      *                   03/16/89
       016600* DETACH INDICATION IS CHECKED AND IF IT IS SET, THE PROGRAM      *                   03/16/89
       016700* IS ENDED (IN44-ON).                                           *                   03/16/89
5738CB1 V2R1M0  910524            AS/400 COBOL Source            INTLIB/CTDINT         RCH38321  10/05/90 16:14:43     Page    7
 STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..IDENTFCN  S  COPYNAME   CHG DATE
       016800*                                                               *                   10/01/87
       016900*****************************************************************                   10/01/87
       017000* 6                                                                                 03/17/89
       017100 READ-CFILE.                                                                        10/01/87
   107 017200     MOVE ZEROS TO CMNF-INDIC-AREA.                                                10/01/87
   108 017300     READ CFILE FORMAT IS "RCVPART"                                                10/01/87
       017400        INDICATORS ARE CMNF-INDIC-AREA.                                           10/01/87
   109 017500     IF IN44-ON                                                                    10/01/87
   110 017600        GO TO END-PROGRAM.                                                         10/01/87
       017700 READ-CFILE-EXIT.                                                                   10/01/87
       017800     EXIT.                                                                         03/15/89
       017900*                                                                                  10/01/87
       018000*****************************************************************                   10/01/87
       018100*                                                               *                   03/16/89
       018200* ROUTINE TO END THE JOB AND CLOSE THE FILES.                   *                   03/16/89
       018300*                                                               *                   03/16/89
       018400*****************************************************************                   10/01/87
       018500*                                                                                   10/01/87
       018600* 7                                                                                 03/17/89
   111 018700 END-PROGRAM.                                                                       10/01/87
   112 018800     MOVE MAJ-MIN TO RC.                                                           01/14/88
   113 018900     MOVE "CTDINT HAS COMPLETED NORMALLY" TO ERRMSG.                               11/15/88
   114 019000     WRITE PRINTREC.                                                               10/01/87
   115 019100     CLOSE PFILE                                                                   10/01/87
       019200           CFILE                                                                   10/01/87
       019300           QPRINT.                                                                 10/01/87
   116 019400     STOP RUN.                                                                     10/01/87
                        * * * * *  E N D  O F  S O U R C E  * * * * *
5738CB1 V2R1M0  910524            AS/400 COBOL Messages          INTLIB/CTDINT         RCH38321  10/05/90 16:14:43     Page    8
 STMT
*   89  MSGID: LBL0335  SEVERITY: 00  SEQNBR:  009900
       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)
      1          1            0               0               0               0
 Source records read . . . . . . . . . :   194
 Copy records read . . . . . . . . . . :   38
 Copy members processed  . . . . . . . :   2
 Sequence errors . . . . . . . . . . . :   0
 Highest severity message issued . . :   0
  LBL0901 00  Program CTDINT created in library INTLIB.
                     * * * * *  E N D  O F  C O M P I L A T I O N  * * * * *
```

Figure D-13 (Part 4 of 4). Target Program Example — CTDINT (User-Defined Formats)

RPG/400 Source Program for a Two-Session Inquiry

The following describes an RPG/400 source program for a two-session inquiry.

Program Files: The RPG/400 two-session source program uses the following files:

INTFIL An 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 An AS/400 printer file used to print records, both sent and received, as well as major and minor ICF return codes.

DDS Source: The DDS for the ICF file (INTFIL) is illustrated in Figure D-14.

```
5714PW1 R01M00  880301                SEU SOURCE LISTING                          10/14/87 17:20:41               PAGE 1
SOURCE FILE . . . . . .  QINTSRC/INTLIB
MEMBER  . . . . . .  INTFIL
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
   A*************************************************************
   A*                                                           *
   A*                      ICF FILE                             *
   A*         USED IN SOURCE TWO SESSION PROGRAM                *
   A*                                                           *
   A*************************************************************
   A                                    INDARA
   A                                    RCVFAIL(25 'RECEIVED FAIL')
   A                                    RCVTRNRND(90)
   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(&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
   A          R TIMER
   A                                    TIMER(000030)
```

Figure D-14. DDS Source for a Two-Session Source Program Using INTFIL

The DDS source file for the display file (DSPFIL) is shown in Figure D-15.

```
5714PW1 R01M00  880301                SEU SOURCE LISTING                          10/14/87 16:59:50               PAGE 1
SOURCE FILE . . . . . .  QINTSRC/INTLIB
MEMBER  . . . . . .  DSPFIL
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
     A*****************************************************************
     A*                                                               *
     A*                       DISPLAY FILE                            *
     A*           USED IN SOURCE TWO SESSION PROGRAM                  *
     A*                                                               *
     A*****************************************************************
     A*  BEGINNING MENU
     A*******************
     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. Order inquiry'
     A                                   5  3'2. Buyer inquiry'
     A                                  11  1'Option:'
     A            OPTION         1N  I 11  9VALUES('1' '2')
     A                                  19  5DFT('CMD KEY 1 - END ')
     A          R DTLMNU                   TEXT('   BUYER INQUIRY SCREEN 1')
     A                                   2  2DFT('ENTER BUYER')
     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('   BUYER INQUIRY SCR. #2')
     A                                   1  3DFT('BUYER DPT LAST ORD   & THIS   +
     A                                        $MTH1     &MTH2     $MTH3     THIS+
     A                                        YTD CNAME')
     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          5      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
```

Figure D-15 (Part 1 of 2). DDS Source for Source Program Two-Session Inquiry Using DSPFIL

```
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 - BUYER 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          9Y 2  10 16EDTCDE(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('TARGET PROGRAM TIMED OUT. ENTE-
A                                       R 1 TO TRY AGAIN OR 2 TO END.')
A            TIMRSP         1   I 20 61
```

Figure D-15 (Part 2 of 2). DDS Source for Source Program Two-Session Inquiry Using DSPFIL

Configuration: The following command is needed to create the intrasystem communications device associated with the ICF file:

```
CRTDEVINTR DEVD(INTRADEV)
  RMTLOCNAME(INTRARMT) ONLINE(*NO)
  TEXT("THIS IS AN INTRASYSTEM DEVICE
  DESCRIPTION")
```

ICF File Creation and Program Device Entry Definition: The command needed to create the ICF file is:

```
CRTICFF  FILE(INTLIB/INTFIL)
  SRCFILE(INTLIB/QINTSRC)
  SRCMBR(INTFIL) ACQPGMDEV(*NONE)
  MAXPGMDEV(2)  WAITRCD(30)
  TEXT("SOURCE ICF FILE FOR TWO SESSION
  PROGRAM")
```

It is not necessary to add a communications entry to the subsystem since the system automatically defines an entry for the device created above at run time. However, if you decided to have one, the following is an example:

```
ADDCMNE SBSD(QCMN) DEV(INTRADEV)
```

The commands needed to define the two program device entries are:

```
OVRICFDEVE PGMDEV(ICF00)
  RMTLOCNAME(INTRARMT)
  FMTSLT(*RECID)
OVRICFDEVE PGMDEV(ICF01)
  RMTLOCNAME(INTRARMT)
  FMTSLT(*RECID)
```

The following is an example of a CL program that might be used to run the source program shown in the example above:

```
RSDINTCL: PGM PARM(&RMT1 &RMT2)
             DCL   VAR(&RMT1)   TYPE(*CHAR)
                   LEN(8)
             DCL   VAR(&RMT2)   TYPE(*CHAR)
                   LEN(8)
          CHGJOB     OUTQ(INTLIB/INTOUTQ)
                     LOG(4 00 *SECLVL)
                     LOGCLPGM(*YES)
          OVRICFDEVE PGMDEV(ICF00)
                     RMTLOCNAME(&RMT1)
                     FMTSLT(*RECID)
          OVRICFDEVE PGMDEV(ICF01)
                     RMTLOCNAME(&RMT2)
                     FMTSLT(*RECID)
          CALL INTLIB/RSDINT
ENDRSDINTCL: ENDPGM
```

The following is an example of a CL program that might be used as the target program that your source program evokes (which calls the program RTDINT shown in the example):

```
RTDINTCL: PGM
            CHGJOB OUTQ(INTLIB/INTOUTQ)
                   LOG(4 00 *SECLVL)
                   LOGCLPGM(*YES)
            ADDLIBLE INTLIB
            OVRICFDEVE PGMDEV(RQSDEV)
            RMTLOCNAME(*REQUESTER)
            CALL INTLIB/RTDINT
            RMVLIBLE INTLIB
ENDRTDINTCL: ENDPGM
```

Program Explanation: The following explains the structure of the program example illustrated in Figure D-16 on page D-44. The ICF file used in the example 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.

The ICF file used in the example is externally described.

All output operations to the ICF file in the example are done using the WRITE statement.

1 The file specifications define the ICF file (INTFIL) and the display file (DSPFIL) used in the program.

INTFIL is the ICF file used to send records to and receive records from each of the two 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: The continuation lines on the file specification define the following:

- The data structure names, IOFB and IODS, used for the feedback area (INFDS) for INTFIL and DSPFIL respectively.
- The number of program devices that can be attached to the files (two for INTFIL).
- The program device name in the 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 referred to in the program is RECID, found in positions 261 through 268 of the feedback area.

3 The two 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 EVKSR in section 13.

When control returns from the EVKSR subroutine, 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 two 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 the read to the display program device section (4).

If the option is not 1, the Buyer Inquiry menu (DTLMNU) is written to the work station and control is passed to section 4.

6 The item number requested by the user from the Order Inquiry Display (record format ITMMNU) is processed here. If function key 1 is pressed (indicator 99), control passes to the I/O operation error section (section 12), the two 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 section 4.

The item number is read from the work station and then the request is sent to the target program on program device ICF01.

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 timer is issued for 30 seconds before the read operation. This is provided to allow the local program to have a time-out when no response is received from the target program.

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. The subroutine ERRCHK in section 14 gets control, and further checks are made.

The return codes are checked after an I/O request. If there are any errors, control is passed to section 12. If not, the program returns to section 4.

7 The information received from the target program is processed. If the information received is a fail indication, it means the requested item number was not found and the request is not valid. A new Item Inquiry menu (ITMMNU) is written to the work station, and control goes to section 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 section 4.

8 This section processes the user requests for additional information (record format ITMSC2). If function key 2 (indicator 98) is pressed, the main menu (record format CIMENU) writes to the work station and control goes to section 4.

If the Enter key is pressed, 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 section 4. If function key 1 (indicator 99) was pressed, control goes to section 12.

If function key 3 (indicator 97) is pressed, the Order Inquiry menu (ITMMNU) is written to the work station and the program returns to section 4.

9 This section processes inquiry read requests from the Buyer menu (DTLMNU). If function key 2 (indicator 98) is pressed, the main menu (CIMENU) is written to the work station and the program returns to section 4. If function key 1 (indicator 99) is pressed, control goes to section 12.

The buyer 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.

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.

Note: A timer operation is issued before the above read is issued to ensure that the operation will finish even if the target program is unable to respond.

If the information received is a fail indication (indicator 25) from the target program, it means the requested item was not found and the request is not valid. The main menu (record format CIMENU) is written to the work station. The program then returns to section 4.

The return codes (or indicators) are checked after an I/O request. If there are any errors, control is passed to section 12.

10 The information supplied by the target program in response to a request for a buyer detail is processed.

The detail information is written to the work station using record format DTLSCR.

The program then returns to section 4.

11 Control is passed here if the buyer detail record format (DLTSCR) is displayed. If function key 1 (indicator 99) is pressed, control goes to section 12. If function key 2 (indicator 98) is pressed, the main menu (CIMENU) is written to the work station and control is returned to section 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 two 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 The EVKSR subroutine in this section builds the evoke requests to send to the remote programs. Because the DDS keyword for the record format only specifies the field identifiers with the record, this code moves the literal value RTDINTCL to the field PGMID, and INTLIB to the field LIB.

When the program start request is received at the remote program, INTLIB is searched for RTDINTCL and that program is then started. RTDINTCL is a CL program that contains CL statements as illustrated on D-41.

14 The subroutine ERRCHK 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 is displayed to the user. The message asks whether you want to try the read operation again or end the job. In this example, the time interval is specified in section 9.

15 The subroutine *PSSR is called if there are I/O operation errors that are not handled by the subroutine ERRCHK in section 14. It checks to see 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.

```
5738RG1 V2R1M0  910524            IBM AS/400 RPG/400                 INTLIB/RSDINT        10/05/90  16:12:28      Page       1
Compiler . . . . . . . . . . . . :   IBM AS/400 RPG/400
Command Options:
  Program  . . . . . . . . . . . :   INTLIB/RSDINT
  Source file  . . . . . . . . . :   INTLIB/QINTSRC
  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 . . . . . . . . . . . . :   RSDINT
  File . . . . . . . . . . . . . :   QINTSRC
  Library  . . . . . . . . . . . :   INTLIB
  Last Change  . . . . . . . . . :   10/05/90  15:22:18
  Description  . . . . . . . . . :   RPG Source Intra Program Example
5738RG1 V2R1M0  910524            IBM AS/400 RPG/400                 INTLIB/RSDINT        10/05/90  16:12:28      Page       2
SEQUENCE                                                                    IND    DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE     LINE   ID
                     S o u r c e   L i s t i n g
     100  H       1                                                                          10/13/87
     200  H*****************************************************************               10/13/87
     300  H*  THIS PROGRAM ASSIGNS TWO  SESSIONS AS FOLLOWS:             *                 11/15/88
     400  H*    'ICF00' TO INQUIRE ABOUT A BUYER'S  CREDIT STANDING      *                 11/15/88
     500  H*            BEFORE AN ORDER IS PROCESSED.                    *                 11/15/88
     600  H*    'ICF01' TO INQUIRE ABOUT THE AVAILABILITY OF AN ITEM     *                 11/15/88
     700  H*            BEING ORDERED (ITEM 000001 THRU 999999).         *                 11/15/88
     800  H*  A DISPLAY DEVICE IS USED TO ENTER THE REQUEST ( USING A    *                 10/13/87
     900  H*  BUYER  AND  AN ITEM MENU ) THAT IS SENT TO THE TARGET      *                 10/05/90
    1000  H*  PROGRAM.                                                   *                 10/05/90
    1100  H*****************************************************************               10/13/87
    1200  F*****************************************************************               10/13/87
    1300  F*                                                              *                 03/17/89
    1400  F*                F I L E   S P E C I F I C A T I O N S         *                 03/17/89
    1500  F*                                                              *                 03/17/89
    1600  F*    INTFIL :   ICF FILE USED TO SEND A REQUEST TO ONE         *                 10/05/90
    1700  F*               OF TWO DIFFERENT TARGET PROGRAMS.  TWO         *                 10/05/90
    1800  F*               SESSIONS ARE ACTIVE AT THE SAME TIME.          *                 10/05/90
    1900  F*                                                              *                 03/17/89
    2000  F*    DSPFIL :   DISPLAY FILE USED TO ENTER A REQUEST TO BE     *                 03/17/89
    2100  F*               SENT TO A TARGET PROGRAM.                      *                 10/05/90
    2200  F*                                                              *                 03/17/89
    2300  F*    THE FOLLOWING INFORMATION IS SPECIFIED AS PART OF THE     *                 03/17/89
    2400  F*    FILE SPECIFICATION:                                       *                 03/17/89
    2500  F*               INFDS : I/O FEEDBACK AREA                      *                 03/17/89
    2600  F*               NUM   : SPECIFIES THE MAXIMUM NUMBER OF        *                 03/17/89
    2700  F*                       PROGRAM DEVICES THAT CAN BE ATTACHED   *                 03/17/89
    2800  F*                       TO THIS FILE.  A VALUE OF 2 IS         *                 03/17/89
    2900  F*                       SPECIFIED FOR THE ICF FILE.            *                 10/05/90
    3000  F*                       THIS DEFINES THE FILE AS A             *                 03/17/89
    3100  F*                       MULTIPLE DEVICE FILE.                  *                 03/17/89
    3200  F*               ID    : 10 CHARACTER PROGRAM DEVICE NAME       *                 03/17/89
    3300  F*                       FIELD WHICH SPECIFIES WHICH PROGRAM    *                 03/17/89
    3400  F*                       DEVICE TO DIRECT THE OPERATION.        *                 03/17/89
    3500  F*                                                              *                 03/17/89
    3600  F*                                                              *                 03/17/89
    3700  F*****************************************************************               10/13/87
    3800   *1                                                                              10/13/87
    3900  FINTFIL  CF  E                    WORKSTN                                       11/21/88
    4000  F                                           KINFDS IOFB                         10/13/87
    4100  F                                           KINFSR *PSSR                        10/14/87
    4200  F                                           KNUM        2                       11/15/88
    4300  F                                           KID    CMID                         10/13/87
          RECORD FORMAT(S):  LIBRARY INTLIB FILE INTFIL.
                   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
```

Figure D-16 (Part 1 of 13). Source Program Example — RSDINT

```
5738RG1 V2R1M0  910524            IBM AS/400 RPG/400                 INTLIB/RSDINT        10/05/90  16:12:28      Page       3
SEQUENCE                                                                                 IND    DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE     LINE   ID
                    EXTERNAL FORMAT ITMREQ RPG NAME ITMREQ
                    EXTERNAL FORMAT DTLREQ RPG NAME DTLREQ
                    EXTERNAL FORMAT TIMER RPG NAME TIMER
    4400  FDSPFIL  CF  E                    WORKSTN                                        10/13/87
    4500  F                                            KINFDS IODS                        10/13/87
           RECORD FORMAT(S):  LIBRARY INTLIB 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
    4600  FQPRINT  O   F     132            PRINTER                                        02/13/89
    4700  I*****************************************************************               10/13/87
    4800  I*                                                              *               03/17/89
    4900  I*            I N P U T   S P E C I F I C A T I O N S           *               03/17/89
    5000  I*                                                              *               03/17/89
    5100  I*  IODS   :  REDEFINES THE I/O FEEDBACK AREA OF THE DISPLAY    *               03/17/89
    5200  I*            FILE.  THIS AREA CONTAINS THE NAME OF THE LAST    *               03/17/89
    5300  I*            RECORD PROCESSED.  THIS FIELD IS CALLED RECID.    *               03/17/89
    5400  I*  IOFB   :  REDEFINES THE I/O FEEDBACK AREA FOR THE ICF FILE. *               10/05/90
    5500  I*                                                              *               03/17/89
    5600  I*****************************************************************               10/13/87
    5700  I* 2                                                                            10/13/87
 A000000   INPUT  FIELDS FOR RECORD ITMRSP FILE INTFIL 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 INTFIL FORMAT DTLRSP.
 B000001                                      1   1 RECCUS
 B000002                                      2   70CUSTNO
 B000003                                      8  37 DNAME
 B000004                                     38  430DLSTOR
 B000005                                     44  520DSLSTM
5738RG1 V2R1M0  910524            IBM AS/400 RPG/400                 INTLIB/RSDINT        10/05/90  16:12:28      Page       4
SEQUENCE                                                                                 IND    DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE     LINE   ID
 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 INTFIL FORMAT DETACH.
 D000000   INPUT  FIELDS FOR RECORD EOS FILE INTFIL FORMAT EOS.
 E000000   INPUT  FIELDS FOR RECORD EVKREQ FILE INTFIL FORMAT EVKREQ.
 F000000   INPUT  FIELDS FOR RECORD ITMREQ FILE INTFIL FORMAT ITMREQ.
 F000001                                      1   60ITEMNO
 G000000   INPUT  FIELDS FOR RECORD DTLREQ FILE INTFIL FORMAT DTLREQ.
 G000001                                      1   60CUSTNO
 H000000   INPUT  FIELDS FOR RECORD TIMER FILE INTFIL FORMAT TIMER.
 I000000   INPUT  FIELDS FOR RECORD CIMENU FILE DSPFIL FORMAT CIMENU.
 I000000         MENU FOR INQUIRY
 I000001                                      3   3 *IN97
 I000002                                      2   2 *IN98
 I000003                                      1   1 *IN99
 I000004                                      4   4 OPTION
```

Figure D-16 (Part 2 of 13). Source Program Example — RSDINT

```
J000000   INPUT  FIELDS FOR RECORD DTLMNU FILE DSPFIL FORMAT DTLMNU.
J000000            BUYER INQUIRY SCREEN 1
J000001                                       3   3 *IN97
J000002                                       2   2 *IN98
J000003                                       1   1 *IN99
J000004                                       4   90CUSTNO
K000000   INPUT  FIELDS FOR RECORD DTLSCR FILE DSPFIL FORMAT DTLSCR.
K000000            BUYER INQUIRY SCR. #2
K000001                                       3   3 *IN97
K000002                                       2   2 *IN98
K000003                                       1   1 *IN99
L000000   INPUT  FIELDS FOR RECORD ITMMNU FILE DSPFIL FORMAT ITMMNU.
L000000         ITEM INQUIRY SCREEN ONE
L000001                                       3   3 *IN97
L000002                                       2   2 *IN98
L000003                                       1   1 *IN99
L000004                                       4   90ITEMNO
M000000   INPUT  FIELDS FOR RECORD ITMSC2 FILE DSPFIL FORMAT ITMSC2.
M000000         ITEM INQUIRY SCREEN TWO
M000001                                       3   3 *IN97
M000002                                       2   2 *IN98
M000003                                       1   1 *IN99
N000000   INPUT  FIELDS FOR RECORD ITMSC3 FILE DSPFIL FORMAT ITMSC3.
N000000         ITEM INQUIRY SCREEN 3
N000001                                       3   3 *IN97
N000002                                       2   2 *IN98
N000003                                       1   1 *IN99
O000000   INPUT  FIELDS FOR RECORD TIMOUT FILE DSPFIL FORMAT TIMOUT.
O000000         TIME OUT SCREEN
O000001                                       3   3 *IN97
O000002                                       2   2 *IN98
O000003                                       1   1 *IN99
O000004                                       4   4 TIMRSP
   5800  IIODS        DS                                                    10/13/87
5738RG1 V2R1M0  910524             IBM AS/400 RPG/400                  INTLIB/RSDINT        10/05/90  16:12:28      Page       5
SEQUENCE                                                                        IND    DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE     LINE   ID
    5900  I                                       1 240 FILL01                  10/13/87
    6000  I                                     261 268 RECID                   10/13/87
    6100  I                                     271 415 FILL02                  10/13/87
    6200  IIOFB        DS                                                     10/13/87
    6300  I                                    *ROUTINE LOC                     10/14/87
    6400  I                                    *STATUS  ERR                     10/14/87
    6500  I                                       1 240 FILL03                  10/13/87
    6600  I                                      38  47 FMTNM                   10/05/90
    6700  I                                     273 282 CMID                    10/13/87
    6800  I                                     401 404 MAJMIN                  10/13/87
    6900  I                                     401 402 MAJCOD                  10/13/87
    7000  I                                     403 404 MINCOD                  10/13/87
    7100  I                                     261 268 RECID2                  10/13/87
    7200  I                                     271 415 FILL04                  10/13/87
    7300  C*****************************************************************   10/13/87
    7400  C*                                                               *   03/17/89
    7500  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     *   03/17/89
    7600  C*                                                               *   03/17/89
    7700  C*   THE DISPLAY PROGRAM DEVICE IS IMPLICITLY ACQUIRED WHEN THE *   03/17/89
    7800  C*   FILE IS OPENED.                                             *   03/17/89
    7900  C*                                                               *   03/17/89
    8000  C*   ALL OF THE ICF PROGRAM DEVICES ARE EXPLICITLY ACQUIRED.     *   10/05/90
    8100  C*                                                               *   03/17/89
    8200  C*   THE TARGET PROGRAM IS EVOKED TWICE TO ESTABLISH TWO         *   10/05/90
    8300  C*   DIFFERENT TRANSACTIONS.                                     *   10/05/90
    8400  C*                                                               *   03/17/89
    8500  C*   THE MAIN INQUIRY MENU (CIMENU) IS WRITTEN TO THE USER'S     *   03/17/89
    8600  C*   DISPLAY.                                                    *   03/17/89
    8700  C*                                                               *   03/17/89
    8800  C*****************************************************************   10/13/87
    8900   * 3                                                                10/13/87
    9000  C           ENTRY     TAG                                           10/13/87
    9100  C           'ICF00   'ACQ  INTFIL                  1ST SESSION      11/21/88
    9200  C           'ICF01   'ACQ  INTFIL                  2ND SESSION      11/21/88
    9300  C                     MOVEL'ICF00   'CMID          1ST PROGRAM      10/13/87
    9400  C                     EXSR EVKSR                   CALL EVOKE       10/13/87
    9500  C                     MOVEL'ICF01   'CMID          2ND PROGRAM      10/13/87
    9600  C                     EXSR EVKSR                   CALL EVOKE       10/13/87
    9700  C           MAIN      TAG                                           10/13/87
    9800  C                     WRITECIMENU                                   10/13/87
```

Figure D-16 (Part 3 of 13). Source Program Example — RSDINT

```
 9900 C*************************************************************                  10/13/87
10000 C*                                                           *                  03/17/89
10100 C*                     DETERMINE USER'S REQUEST              *                  03/17/89
10200 C*                                                           *                  03/17/89
10300 C*   A READ TO THE DISPLAY DEVICE IS ISSUED TO RECEIVE THE   *                  03/17/89
10400 C*   USER'S REQUEST. THE TYPE OF REQUEST MADE IS BASED ON THE *                  03/17/89
10500 C*   DISPLAY FORMAT CURRENTLY ON THE SCREEN.  THE RECORD FORMAT *                03/17/89
10600 C*   NAME IS EXTRACTED FROM THE I/O FEEDBACK AREA AND USED TO  *                  03/17/89
10700 C*   DETERMINE WHAT ACTION SHOULD BE TAKEN NEXT.             *                  03/17/89
10800 C*                                                           *                  03/17/89
10900 C*************************************************************                  10/13/87
11000  * 4                                                                            10/13/87
5738RG1 V2R1M0  910524            IBM AS/400 RPG/400                 INTLIB/RSDINT      10/05/90  16:12:28      Page       6
SEQUENCE                                                                        IND    DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE     LINE   ID
11100 C           READRQ    TAG                                                            10/13/87
11200 C                     SETOF                    8889  TIMEOUT IND     1 2           10/13/87
11300 C                     READ DSPFIL                 87                    3           10/13/87
11400 C           RECID     CABEQ'CIMENU  'MENU           MAIN MENU ?                    10/13/87
11500 C           RECID     CABEQ'ITMMNU  'ITMIN          ITEM MENU ?                    10/13/87
11600 C           RECID     CABEQ'ITMSC2  'ITMRTN         ITM SCR?                       10/13/87
11700 C           RECID     CABEQ'ITMSC3  'ITMRTN         ITM SCR?                       10/13/87
11800 C           RECID     CABEQ'DTLMNU  'DTLIN          DETAIL SCR?                    10/13/87
11900 C           RECID     CABEQ'DTLSCR  'DTLRTN         CUST SCR?                      10/13/87
12000 C                     WRITECIMENU                   MAIN MENU IF                   10/13/87
12100 C                     GOTO READRQ                   THERE IS ERR                   10/13/87
12200 C*************************************************************                  10/13/87
12300 C*                                                           *                  03/17/89
12400 C*                            MAIN MENU                      *                  03/17/89
12500 C*                                                           *                  03/17/89
12600 C*    THE MAIN MENU IS READ TO DETERMINE THE REQUEST ENTERED  *                  03/17/89
12700 C*    BY THE USER. IF CMD 1 (*IN99) IS PRESSED, THE PROGRAM   *                  03/17/89
12800 C*    IS ENDED.  IF OPTION = 1, AN ITEM INQUIRY MENU IS WRITTEN *                03/17/89
12900 C*    TO SCREEN.  IF OPTION = 2, A BUYER'S INQUIRY MENU IS    *                  03/17/89
13000 C*    WRITTEN TO THE SCREEN.                                 *                  03/17/89
13100 C*                                                           *                  03/17/89
13200 C*************************************************************                  10/13/87
13300  * 5                                                                            10/13/87
13400 C           MENU      TAG                                                            10/13/87
13500 C           *IN99     CABEQ'1'       END            JOB ENDS                        10/13/87
13600 C           OPTION    IFEQ '1'                                             B001      10/13/87
13700 C                     WRITEITMMNU                   ITEM MENU               001      10/13/87
13800 C                     ELSE                                                 X001      10/13/87
13900 C                     WRITEDTLMNU                   CUST MENU               001      10/13/87
14000 C                     END                                                  E001      10/13/87
14100 C                     GOTO READRQ                                                    10/13/87
14200 C*************************************************************                  03/17/89
14300 C*                                                           *                  03/17/89
14400 C*                        ITEM INQUIRY                       *                  03/17/89
14500 C*                                                           *                  03/17/89
14600 C*    THE ITEM NUMBER REQUESTED BY THE USER ON THE ITEM INQUIRY *                03/17/89
14700 C*    SCREEN IS CHECKED.  THIS IS DETERMINED BY THE          *                  03/17/89
14800 C*    DISPLAY RECORD FORMAT BEING PROCESSED - IN THIS CASE   *                  03/17/89
14900 C*    ITMMNU.                                                *                  03/17/89
15000 C*                                                           *                  03/17/89
15100 C*    IF CMD 1 (*IN99) IS PRESSED, THE PROGRAM IS ENDED.  IF *                  03/17/89
15200 C*    CMD 2 IS PRESSED, THE ITEM INQUIRY REQUEST IS CANCELED, *                 03/17/89
15300 C*    AND THE MAIN MENU (CIMENU) IS WRITTEN TO THE SCREEN.   *                  03/17/89
15400 C*                                                           *                  03/17/89
15500 C*    IF AN ITEM NUMBER IS ENTERED, AN ITEM INQUIRY REQUEST IS *                03/17/89
15600 C*    SENT TO THE APPROPRIATE TARGET PROGRAM.                *                  03/17/89
15700 C*                                                           *                  03/17/89
15800 C*    IF A FAIL INDICATION IS RECEIVED, A FRESH ITEM MENU IS *                  03/17/89
15900 C*    WRITTEN TO THE DISPLAY DEVICE.                         *                  03/17/89
16000 C*                                                           *                  03/17/89
16100 C*    IF AN ERROR OCCURS, THE ERROR IS PRINTED AND THE JOB   *                  03/17/89
16200 C*    IS ENDED.                                              *                  03/17/89
16300 C*                                                           *                  03/17/89
16400 C*************************************************************                  10/13/87
```

Figure D-16 (Part 4 of 13). Source Program Example — RSDINT

```
5738RG1 V2R1M0  910524            IBM AS/400 RPG/400                 INTLIB/RSDINT       10/05/90  16:12:28      Page       7
SEQUENCE                                                                                  IND    DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE     LINE   ID
   16500   * 6                                                                                    10/13/87
   16600  C           ITMIN     TAG                                                             10/13/87
   16700  C           *IN99     CABEQ'1'       END             EXIT ON CMD3                     10/13/87
   16800  C           *IN98     IFEQ '1'                                                B001    10/13/87
   16900  C                     WRITECIMENU                       MAIN MENU              001    10/13/87
   17000  C                     GOTO READRQ                                              001    10/13/87
   17100  C                     END                                                      E001    10/13/87
   17200  C                     MOVEL'ICF01   'CMID                                              10/13/87
   17300  C           XITMIN    TAG                                                             10/13/87
   17400  C                     WRITEITMREQ                       INQ W/INVITE                   11/28/88
   17500  C           MAJCOD    CABGE'04'      ERROR            ERROR RTN                      10/13/87
   17600  C           TRY89     TAG                                                             10/13/87
   17700  C                     SETOF                          89                  3           10/13/87            OQ
   17800  C                     WRITETIMER                        START TIMER                    11/28/88
   17900  C                     MOVEL'        'CMID                                              12/01/88
   18000  C                     READ INTFIL                8910RECV ITM INFO     2 3           11/30/88
   18100  C   89                EXSR ERRCHK                       CHCK ERR INFO                  10/13/87
   18200  C           *IN25     IFEQ '1'                          RECEIVE FAIL           B001    11/16/88
   18300  C                     WRITEITMMNU                       ITEM MENU              001    11/16/88
   18400  C                     GOTO READRQ                                              001    11/16/88
   18500  C                     END                                                      E001    11/16/88
   18600  C           MAJMIN    CABGE'0300'    ITMIN            NODATATRYAGN                   10/13/87
   18700  C           MAJCOD    CABGE'04'      ERROR            ERROR RTN                      10/13/87
   18800  C           RECID2    CABNE'ITMRSP'  RECERR           PRINT MSG                      10/13/87
   18900  C*************************************************************                      03/17/89
   19000  C*                                                           *                      03/17/89
   19100  C*                    PROCESS ITEM INFORMATION               *                      03/17/89
   19200  C*                                                           *                      03/17/89
   19300  C*    THE ITEM RECORD RECEIVED FROM THE TARGET PROGRAM AND THE  *                     03/17/89
   19400  C*    INFORMATION ABOUT THE ITEM IS PROCESSED AND DISPLAYED.    *                     03/17/89
   19500  C*    IF A FAIL INDICATION IS RECEIVED FROM THE TARGET PROGRAM, *                     10/05/90
   19600  C*    ITEM REQUESTED WAS NOT FOUND, AND A FRESH ITEM MENU       *                     03/17/89
   19700  C*    IS DISPLAYED IF ITEMNO IS 0 OR LESS, IT IS AN INVALID     *                     03/17/89
   19800  C*    REQUEST AND A FRESH ITEM MENU IS WRITTEN TO THE SCREEN.   *                     03/17/89
   19900  C*    IF THE REQUEST IS VALID, VALUES ARE CALCULATED BASED ON   *                     03/17/89
   20000  C*    THE INFORMATION RECEIVED.                                 *                     03/17/89
   20100  C*                                                           *                      03/17/89
   20200  C*************************************************************                      03/17/89
   20300   * 7                                                                                  10/13/87
   20400  C           ITMOUT    TAG                                                             10/13/87
   20500  C           ITEMNO    IFLE 000000                                             B001    10/13/87
   20600  C                     WRITEITMMNU                       ITEM MENU              001    10/13/87
   20700  C                     GOTO READRQ                       READ DISPLY            001    10/13/87
   20800  C                     ELSE                                                     X001    10/13/87
   20900  C                     Z-ADD0         QAVAIL  70         QTY AVAIL.             001    10/13/87
   21000  C                     ADD  QTYOH     QAVAIL                                    001    10/13/87
   21100  C                     SUB  QTYOO     QAVAIL                                    001    10/13/87
   21200  C                     ADD  QTYBO     QAVAIL                                    001    10/13/87
   21300  C                     MOVELDESC      DSC                                       001    10/13/87
   21400  C                     MOVE QTYOO     QTYO                                      001    10/13/87
   21500  C                     MOVE QTYOH     QTYH                                      001    10/13/87
   21600  C                     MOVE QTYBO     QTYB                                      001    10/13/87
   21700  C                     MOVE UNITQ     UNT                                       001    10/13/87
   21800  C                     MOVE PR01      PR1                                       001    10/13/87
5738RG1 V2R1M0  910524            IBM AS/400 RPG/400                 INTLIB/RSDINT       10/05/90  16:12:28      Page       8
SEQUENCE                                                                                  IND    DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE     LINE   ID
   21900  C                     MOVE PR05      PR5                                       001    10/13/87
   22000  C                     MOVE UFRT      UFR                                       001    10/13/87
   22100  C                     WRITEITMSC2                       DSP DETAIL             001    10/13/87
   22200  C                     GOTO READRQ                                              001    10/13/87
   22300  C*                    END                                                              10/13/87
```

Figure D-16 (Part 5 of 13). Source Program Example — RSDINT

```
  22400 C*************************************************************                 10/13/87
  22500 C*                                                           *                 03/17/89
  22600 C*                   ADDITIONAL ITEM INFORMATION             *                 03/17/89
  22700 C*                                                           *                 03/17/89
  22800 C*    ADDITIONAL ITEM INFORMATION IS PROCESSED AND THE RESULT*                 03/17/89
  22900 C*    DISPLAYED ON THE SCREEN WHEN A RESPONSE IS READ        *                 03/17/89
  23000 C*    FROM THE DISPLAY STATION WITH AN ITEM SCREEN RECORD    *                 03/17/89
  23100 C*    FORMAT.                                                *                 03/17/89
  23200 C*                                                           *                 03/17/89
  23300 C*    IF CMD 1 (*IN99) IS PRESSED, THE PROGRAM IS ENDED.     *                 03/17/89
  23400 C*                                                           *                 03/17/89
  23500 C*    IF CMD 2 (*IN98) IS PRESSED, THE ITEM INQUIRY IS       *                 03/17/89
  23600 C*    ENDED, AND THE MAIN MENU (CIMENU) IS WRITTEN TO THE    *                 03/17/89
  23700 C*    SCREEN.                                                *                 03/17/89
  23800 C*                                                           *                 03/17/89
  23900 C*    IF CMD 3 (*IN97) IS PRESSED, THE ITEM INQUIRY MENU IS  *                 03/17/89
  24000 C*    WRITTEN ON THE SCREEN.                                 *                 03/17/89
  24100 C*                                                           *                 03/17/89
  24200 C*    IF 'ENTER' IS PRESSED WHILE SCREEN 2 FOR ITEM REQUESTED IS*              03/17/89
  24300 C*    CURRENTLY DISPLAYED, MORE INFORMATION IS CALCULATED AND *                03/17/89
  24400 C*    DISPLAYED.                                             *                 03/17/89
  24500 C*                                                           *                 03/17/89
  24600 C*    IF 'ENTER' IS PRESSED WHILE SCREEN 3 FOR ITEM REQUESTED IS*              03/17/89
  24700 C*    CURRENTLY DISPLAYED, THEN THE ITEM INQUIRY MENU        *                 03/17/89
  24800 C*    IS WRITTEN TO THE SCREEN.                              *                 03/17/89
  24900 C*                                                           *                 03/17/89
  25000 C*************************************************************                 03/17/89
  25100  * 8                                                                          10/13/87
  25200 C           ITMRTN    TAG                                                 001   10/13/87
  25300 C           *IN99     CABEQ'1'       END             JOB ENDS             001   10/13/87
  25400 C           *IN98     IFEQ '1'                                           B002   10/13/87
  25500 C                     WRITECIMENU                    MAIN MENU            002   10/13/87
  25600 C                     GOTO READRQ                                         002   10/13/87
  25700 C                     END                                                E002   10/13/87
  25800 C           *IN97     IFEQ '1'                       CMD 3 ?              B002   10/13/87
  25900 C           RECID     IFEQ 'ITMSC2  '                ITM SCR 2 ?          B003   10/13/87
  26000 C                     WRITEITMMNU                    YES,THEN ITS         003   10/13/87
  26100 C                     GOTO READRQ                    ITEM MENU            003   10/13/87
  26200 C                     END                                                E003   10/13/87
  26300 C                     END                                                E002   10/13/87
  26400 C           RECID     IFEQ 'ITMSC3  '                ITM SCR 3 ?          B002   10/13/87
  26500 C                     WRITEITMMNU                    YES,THEN ITS         002   10/13/87
  26600 C                     GOTO READRQ                    ITEM MENU            002   10/13/87
  26700 C                     END                                                E002   10/13/87
  26800 C           SLSTM     SUB  CSTTM     PROFM   92      PROF MONTH           001   10/13/87
  26900 C                     MULT 100       PROFM                               001   10/13/87
  27000 C           SLSTM     COMP 0                        46                3   001   10/13/87
  27100 C     N46   PROFM     DIV  SLSTM     PROFM           PROF PCT             001   10/13/87
  27200 C           QTYLST    MULT PR01      LOSTS           LOST SALES           001   10/13/87
5738RG1 V2R1M0  910524            IBM AS/400 RPG/400                INTLIB/RSDINT       10/05/90  16:12:28      Page       9
SEQUENCE                                                                    IND    DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE     LINE   ID
  27300 C                     MOVE SLSTM     SLSM                                001   10/13/87
  27400 C                     MOVE SLSTY     SLSY                                001   10/13/87
  27500 C                     MOVE CSTTM     CSTM                                001   10/13/87
  27600 C                     MOVE PROFM     PROFIT                              001   10/13/87
  27700 C                     MOVE CSTTY     CSTY                                001   10/13/87
  27800 C                     WRITEITMSC3                    DET ITM INF          001   10/13/87
  27900 C                     GOTO READRQ                                         001   10/13/87
  28000 C*************************************************************                 10/13/87
  28100 C*                                                           *                 03/17/89
  28200 C*                      BUYER INQUIRY                        *                 03/17/89
  28300 C*                                                           *                 03/17/89
  28400 C*    THE REQUEST FROM THE BUYER INQUIRY MENU IS PROCESSED.  *                 03/17/89
  28500 C*                                                           *                 03/17/89
  28600 C*    IF CMD 1 (*IN99) IS PRESSED, THE PROGRAM IS ENDED.     *                 03/17/89
  28700 C*                                                           *                 03/17/89
  28800 C*    IF CMD 2 (*IN98) IS PRESSED, THE BUYER INQUIRY IS ENDED,*                03/17/89
  28900 C*    AND THE MAIN MENU (CIMENU) IS WRITTEN TO THE SCREEN.   *                 03/17/89
  29000 C*                                                           *                 03/17/89
  29100 C*    IF A BUYER NUMBER IS ENTERED, THE BUYER INQUIRY        *                 03/17/89
  29200 C*    REQUEST IS SENT TO THE TARGET PROGRAM.                 *                 03/17/89
  29300 C*                                                           *                 03/17/89
  29400 C*    A READ TO THE ICF PROGRAM DEVICE IS ISSUED TO RECEIVE  *                 10/05/90
  29500 C*    THE INFORMATION FROM THE TARGET PROGRAM.               *                 03/17/89
  29600 C*                                                           *                 03/17/89
  29700 C*    IF A FAIL INDICATION IS RECEIVED, A FRESH MAIN MENU IS *                 03/17/89
  29800 C*    WRITTEN TO THE DISPLAY DEVICE.                         *                 03/17/89
  29900 C*                                                           *                 03/17/89
```

Figure D-16 (Part 6 of 13). Source Program Example — RSDINT

```
   30000  C*    IF AN ERROR OCCURS, THE ERROR IS PRINTED AND THE JOB IS   *                 03/17/89
   30100  C*    ENDED.                                                    *                 03/17/89
   30200  C*                                                              *                 03/17/89
   30300  C*************************************************************                 10/13/87
   30400   * 9                                                                            10/13/87
   30500  C           DTLIN     TAG                                                 001   10/13/87
   30600  C           *IN99     CABEQ'1'       END             JOB ENDS             001   10/13/87
   30700  C           *IN98     IFEQ '1'                                           B002   10/13/87
   30800  C                     WRITECIMENU                    MAIN MENU            002   10/13/87
   30900  C                     GOTO READRQ                                         002   10/13/87
   31000  C                     END                                                E002   10/13/87
   31100  C           EVDTL     TAG                                                 001   10/13/87
   31200  C                     MOVEL'ICF00   'CMID                                 001   10/13/87
   31300  C                     WRITEDTLREQ                    CUST INQ             001   10/13/87
   31400  C           MAJCOD    CABGE'04'      ERROR           ERROR RTN            001   10/13/87
   31500  C           TRY88     TAG                                                 001   10/13/87
   31600  C                     SETOF                        88                3    001   10/13/87
   31700  C                     WRITETIMER                     START TIMER          001   11/28/88
   31800  C                     MOVEL'        'CMID                                 001   12/01/88
   31900  C                     READ INTFIL                8810RCV CUS INF       2 3  001   12/01/88
   32000  C   88                EXSR ERRCHK                    CHECK ERR            001   10/13/87
   32100  C           *IN25     IFEQ '1'                       RECEIVE FAIL        B002   11/16/88
   32200  C                     SETOF                        66                3    002   03/17/89
   32300  C                     WRITECIMENU                    MAIN MENU            002   11/16/88
   32400  C                     GOTO READRQ                                         002   11/16/88
   32500  C                     END                                                E002   11/16/88
   32600  C           MAJMIN    CABGE'0300'    EVDTL           NODATATRYAGN         001   10/13/87
5738RG1 V2R1M0  910524            IBM AS/400 RPG/400                INTLIB/RSDINT      10/05/90  16:12:28      Page      10
SEQUENCE                                                                      IND    DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE     LINE   ID
   32700  C           MAJCOD    CABGE'04'      ERROR           ERROR RTN            001   10/13/87
   32800  C           RECID2    CABNE'DTLRSP'  RECERR          PRINT MSG            001   10/13/87
   32900  C*************************************************************                 10/13/87
   33000  C*                                                            *                 03/17/89
   33100  C*                     PROCESS BUYER INFORMATION              *                 03/17/89
   33200  C*                                                            *                 03/17/89
   33300  C*     THE BUYER DATA RECEIVED FROM THE TARGET PROGRAM         *                 03/17/89
   33400  C*     IS PROCESSED AND THE INFORMATION IS WRITTEN TO THE      *                 03/17/89
   33500  C*     SCREEN.                                                *                 03/17/89
   33600  C*                                                            *                 03/17/89
   33700  C*************************************************************                 10/13/87
   33800   * 10                                                                           10/13/87
   33900  C           DTOUT     TAG                                                 001   10/13/87
   34000  C                     MOVE CUSTNO    CUSTN                                001   10/13/87
   34100  C                     MOVELDNAME     CNAME                                001   03/17/89
   34200  C                     MOVE DLSTOR    DLSTR                                001   10/13/87
   34300  C                     MOVE DSLSTM    DSLSM                                001   10/13/87
   34400  C                     MOVE DSPM01    DSPM1                                001   10/13/87
   34500  C                     MOVE DSPM02    DSPM2                                001   10/13/87
   34600  C                     MOVE DSTTYD    DSTYD                                001   10/13/87
   34700  C                     MOVE IDEPT     DEPT                                 001   10/13/87
   34800  C                     WRITEDTLSCR                    BLD CUS SCR          001   10/13/87
   34900  C                     GOTO READRQ                                         001   10/13/87
   35000  C*************************************************************                 10/13/87
   35100  C*                                                            *                 03/17/89
   35200  C*    THIS ROUTINE HANDLES THE USER'S REQUEST FOLLOWING THE   *                 03/17/89
   35300  C*    DISPLAY OF THE BUYER INFORMATION.  CMD KEY 1 WILL       *                 03/17/89
   35400  C*    EXIT THE JOB, CMD KEY 2 WILL DISPLAY THE MAIN MENU, AND *                 03/17/89
   35500  C*    "ENTER" WILL BRING UP THE BUYER INQUIRY MENU.           *                 03/17/89
   35600  C*                                                            *                 03/17/89
   35700  C*************************************************************                 10/13/87
   35800   * 11                                                                           10/13/87
   35900  C           DTLRTN    TAG                                                 001   10/13/87
   36000  C           *IN99     CABEQ'1'       END             JOB ENDS             001   10/13/87
   36100  C           *IN98     IFEQ '1'                                           B002   10/13/87
   36200  C                     WRITECIMENU                    MAIN MENU            002   10/13/87
   36300  C                     GOTO READRQ                                         002   10/13/87
   36400  C                     END                                                E002   10/13/87
   36500  C                     WRITEDTLMNU                    BUYER INQ            001   11/21/88
   36600  C                     GOTO READRQ                                         001   10/13/87
   36700  C*                                                                              10/13/87
   36800  C*************************************************************                 10/13/87
   36900  C*                                                            *                 03/17/89
   37000  C*    WHEN AN I/O OPERATION ERROR IS DETECTED, A MESSAGE IS   *                 03/17/89
   37100  C*    PRINTED AND THE TRANSACTION AND SESSION ARE ENDED       *                 03/17/89
   37200  C*    WITH EACH OF THE TARGET PROGRAMS.                       *                 03/17/89
   37300  C*                                                            *                 03/17/89
```

Figure D-16 (Part 7 of 13). Source Program Example — RSDINT

```
   37400 C*************************************************************                  10/13/87
   37500  * 12                                                                          10/13/87
   37600 C           RECERR    TAG                                                     001   10/13/87
   37700 C                     EXCPTRECER                  WRONG RECID                 001   03/17/89
   37800 C                     GOTO END                    END PROGRAM                 001   10/13/87
   37900 C           ERROR     TAG                                                     001   10/13/87
   38000 C                     EXCPTMMERR                                              001   03/17/89
5738RG1 V2R1M0  910524            IBM AS/400 RPG/400                INTLIB/RSDINT       10/05/90  16:12:28      Page      11
SEQUENCE                                                                        IND   DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE   NUM   UPDATE     LINE   ID
   38100 C           END       TAG                                                     001   10/13/87
   38200 C                     MOVEL'ICF00   'CMID                                     001   10/13/87
   38300 C                     WRITEDETACH                 DET 1ST SES                 001   03/17/89
   38400 C                     MOVEL'ICF01   'CMID                                     001   10/13/87
   38500 C                     WRITEDETACH                 DET 2ND SES                 001   03/17/89
   38600 C           ABORT     TAG                                                     001   10/13/87
   38700 C           'ICF00   'REL  INTFIL                86  REL 1ST SES       2     001   11/21/88
   38800 C           'ICF01   'REL  INTFIL                86  REL 2ND SES       2     001   11/21/88
   38900 C           FORCE     TAG                                                     001   10/13/87
   39000 C                     SETON                     LR                    1       001   10/13/87
   39100 C                     RETRN                                                   001   10/13/87
   39200 C                     END                                                     E001  10/13/87
   39300 C*************************************************************                  10/13/87
   39400 C*                                                           *                  03/17/89
   39500 C*    THIS SUBROUTINE IS CALLED TO EVOKE THE TARGET PROGRAM.  *                  03/17/89
   39600 C*    THE SAME TARGET PROGRAM (ICFLIB/RTDMULCL) IS EVOKED     *                  03/17/89
   39700 C*    TWO DIFFERENT TIMES CREATING TWO JOBS. THE PROGRAM DEVICE *                03/17/89
   39800 C*    IDENTIFIES WHICH SESSION SHOULD BE EVOKED.  THE PROGRAM  *                 03/17/89
   39900 C*    DEVICE WAS SPECIFIED IN CMID PRIOR TO CALLING THIS       *                 03/17/89
   40000 C*    ROUTINE.                                                 *                 03/17/89
   40100 C*                                                           *                  03/17/89
   40200 C*************************************************************                  10/13/87
   40300  * 13                                                                          10/13/87
   40400 C           EVKSR     BEGSR                                                   10/13/87
   40500 C                     MOVE *BLANK    PGMID        BLANK OUT                   10/13/87
   40600 C                     MOVE *BLANK    LIB          BLANK OUT                   10/13/87
   40700 C                     MOVEL'RTDINTCL'PGMID        PROGR NAME                  12/12/88
   40800 C                     MOVEL'INTLIB  'LIB          LIBRARY                     02/13/89
   40900 C                     WRITEEVKREQ                                             10/13/87
   41000 C           MAJCOD    CABGE'04'      END          TO END PGM                  10/13/87
   41100 C                     ENDSR                                                   10/13/87
   41200 C*************************************************************                  10/13/87
   41300 C*                                                           *                  03/17/89
   41400 C*    THIS SUBROUTINE IS CALLED TO PERFORM FURTHER CHECKS ON   *                 03/17/89
   41500 C*    FILE ERRORS RESULTING FROM THE READ OPERATION ISSUED TO  *                 03/17/89
   41600 C*    THE PROGRAM DEVICE. THIS ROUTINE CHECKS FOR THE TIME     *                 03/17/89
   41700 C*    OUT INDICATION. IF IT IS, A MESSAGE IS SENT TO THE USER  *                 03/17/89
   41800 C*    DISPLAY SCREEN REQUESTING ACTION, OTHERWISE PROGRAM ENDS. *                03/17/89
   41900 C*    ALSO, IF A FAIL INDICATION IS RECEIVED, A FRESH MAIN MENU *                03/17/89
   42000 C*    IS WRITTEN TO THE DISPLAY DEVICE.                        *                 03/17/89
   42100 C*                                                           *                  03/17/89
   42200 C*************************************************************                  10/13/87
   42300  * 14                                                                          10/13/87
   42400 C           ERRCHK    BEGSR                                                   10/13/87
   42500 C           MAJMIN    IFEQ '0310'                 TIMER EXPD?             B001  10/13/87
   42600 C           CHKAGN    TAG                                                     001   10/13/87
   42700 C                     WRITETIMOUT                 DISPLAY MSG             001   10/13/87
   42800 C                     READ DSPFIL                86READ REPLY          3    001   10/13/87
   42900 C   88      TIMRSP    CABEQ'1'       TRY88        CUST INQUIR             001   10/13/87
   43000 C   89      TIMRSP    CABEQ'1'       TRY89        ITEM INQUIR             001   10/13/87
   43100 C           TIMRSP    IFEQ '2'                    END PROGRAM             B002  10/13/87
   43200 C                     WRITEEOS                    END SESSION             002   10/13/87
   43300 C                     GOTO FORCE                  END PROGRAM             002   10/13/87
   43400 C                     END                                                 E002  10/13/87
5738RG1 V2R1M0  910524            IBM AS/400 RPG/400                INTLIB/RSDINT       10/05/90  16:12:28      Page      12
SEQUENCE                                                                        IND   DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE   NUM   UPDATE     LINE   ID
   43500 C                     GOTO CHKAGN                 ASK AGAIN               001   10/13/87
   43600 C                     END                                                 E001  10/13/87
   43700 C           *IN25     IFEQ '1'                    RECEIVE FAIL            B001  11/16/88
   43800 C                     WRITECIMENU                 MAIN MENU               001   11/16/88
   43900 C                     GOTO READRQ                                         001   11/16/88
   44000 C                     END                                                 E001  11/16/88
   44100 C                     GOTO ERROR                  ABEND                   10/13/87
   44200 C                     ENDSR                                               10/13/87
```

Figure D-16 (Part 8 of 13). Source Program Example — RSDINT

```
    44300  C*************************************************************                  10/14/87
    44400  C*                                                           *                  03/17/89
    44500  C*    THIS IS THE PROGRAM ERROR SUBROUTINE THAT RECEIVES      *                  03/17/89
    44600  C*    CONTROL WHEN AN ERROR OCCURS AFTER AN I/O OPERATION     *                  03/17/89
    44700  C*    IS ISSUED TO THE PROGRAM DEVICE AND THERE IS A NON-     *                  03/17/89
    44800  C*    ZERO VALUE IN THE RPG STATUS FIELD (ERR).               *                  03/17/89
    44900  C*    THIS ROUTINE CHECKS FOR STATUS VALUES THAT RELATE TO    *                  03/17/89
    45000  C*    ICF OPERATIONS.                                         *                  10/05/90
    45100  C*    IF THE PROGRAM DEVICE IS ALREADY ACQUIRED, THE ERROR IS *                  03/17/89
    45200  C*    IGNORED, OTHERWISE, THE PROGRAM IS TERMINATED.          *                  03/17/89
    45300  C*                                                           *                  03/17/89
    45400  C*************************************************************                  10/14/87
    45500   * 15                                                                           10/14/87
    45600  C           *PSSR     BEGSR                                                     10/14/87
    45700  C                     MOVE '      '  RETURN  6        DEFAULT                  10/14/87
    45800  C           ERR       CABEQ01285     ENDPSR           ALREADY ACQ?             10/14/87
    45900  C                     MOVE '*CANCL'  RETURN           JOB ENDS                 10/14/87
    46000  C           ENDPSR    ENDSRRETURN                    BACK TO MAIN              02/13/89
    46100  C*************************************************************                  10/13/87
    46200  OQPRINT  E 1              MMERR                                                 10/13/87
    46300  O                                   21 'COMMUNICATION ERROR.'                    10/13/87
    46400  O                                   34 'MAJOR/MINOR:'                            10/13/87
    46500  O                         MAJCOD    37                                         10/13/87
    46600  O                                   38 '/'                                     10/13/87
    46700  O                         MINCOD    40                                         10/13/87
    46800  O                                   49 'FORMAT:'                               10/13/87
    46900  O                         FMTNM     60                                         10/13/87
    47000  O                                   69 'PGMDEV:'                               10/13/87
    47100  O                         CMID      80                                         10/13/87
    47200  O        E 1              RECER                                                10/13/87
    47300  O                                   20 'UNMATCH RECD FORMAT'                   10/13/87
    47400  O                                   31 '-JOB ENDS.'                            10/13/87
    47500  O                         MAJCOD    37                                         10/13/87
    47600  O                                   38 '/'                                     10/13/87
    47700  O                         MINCOD    40                                         10/13/87
    47800  O                                   49 'FORMAT:'                               10/13/87
    47900  O                         RECID2    60                                         10/13/87
    48000  O                                   69 'PGMDEV:'                               10/13/87
    48100  O                         CMID      80                                         10/13/87
* 6103     48101   OVERFLOW INDICATOR OA ASSIGNED TO FILE QPRINT.
  P000000   OUTPUT FIELDS FOR RECORD DETACH FILE INTFIL FORMAT DETACH.
  Q000000   OUTPUT FIELDS FOR RECORD EOS FILE INTFIL FORMAT EOS.
  R000000   OUTPUT FIELDS FOR RECORD EVKREQ FILE INTFIL FORMAT EVKREQ.
  R000001                           PGMID     10  CHAR   10
 5738RG1 V2R1M0  910524            IBM AS/400 RPG/400                  INTLIB/RSDINT        10/05/90  16:12:28      Page      13
 SEQUENCE                                                                                IND    DO    LAST       PAGE   PROGRAM
 NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE     LINE   ID
  R000002                           LIB       20  CHAR   10
  S000000   OUTPUT FIELDS FOR RECORD ITMREQ FILE INTFIL FORMAT ITMREQ.
  S000001                           ITEMNO     6  ZONE  6,0
  T000000   OUTPUT FIELDS FOR RECORD DTLREQ FILE INTFIL FORMAT DTLREQ.
  T000001                           CUSTNO     6  ZONE  6,0
  U000000   OUTPUT FIELDS FOR RECORD TIMER FILE INTFIL FORMAT TIMER.
  V000000   OUTPUT FIELDS FOR RECORD CIMENU FILE DSPFIL FORMAT CIMENU.
  V000000         MENU FOR INQUIRY
  W000000   OUTPUT FIELDS FOR RECORD DTLMNU FILE DSPFIL FORMAT DTLMNU.
  W000000           BUYER INQUIRY SCREEN 1
  X000000   OUTPUT FIELDS FOR RECORD DTLSCR FILE DSPFIL FORMAT DTLSCR.
  X000000           BUYER INQUIRY SCR. #2
  X000001                           CUSTN      6  CHAR    6
  X000002                           DEPT       9  ZONE  3,0
  X000003                           DLSTR     15  ZONE  6,0
  X000004                           DSLSM     24  ZONE  9,0
  X000005                           DSPM1     33  ZONE  9,0
  X000006                           DSPM2     42  ZONE  9,0
  X000007                           DSPM3     51  ZONE  9,0
  X000008                           DSTYD     62  ZONE 11,0
  X000009                           CNAME     67  CHAR    5
  Y000000   OUTPUT FIELDS FOR RECORD ITMMNU FILE DSPFIL FORMAT ITMMNU.
  Y000000         ITEM INQUIRY SCREEN ONE
  Z000000   OUTPUT FIELDS FOR RECORD ITMSC2 FILE DSPFIL FORMAT ITMSC2.
  Z000000         ITEM INQUIRY SCREEN TWO
  Z000001                           DSC       30  CHAR   30
  Z000002                           QAVAIL    37  ZONE  7,0
  Z000003                           QTYH      44  ZONE  7,0
  Z000004                           QTYO      51  ZONE  7,0
  Z000005                           QTYB      58  ZONE  7,0
  Z000006                           UNT       60  CHAR    2
  Z000007                           PR1       67  ZONE  7,2
  Z000008                           PR5       74  ZONE  7,0
  Z000009                           UFR       79  ZONE  5,2
```

Figure D-16 (Part 9 of 13). Source Program Example — RSDINT

```
  1000000   OUTPUT FIELDS FOR RECORD ITMSC3 FILE DSPFIL FORMAT ITMSC3.
  1000000         ITEM INQUIRY SCREEN 3
  1000001                           SLSM       9  ZONE  9,2
  1000002                           SLSY      20  ZONE 11,2
  1000003                           CSTM      29  ZONE  9,2
  1000004                           CSTY      40  ZONE 11,2
  1000005                           PROFIT    45  ZONE  5,2
  1000006                           LOSTS     54  ZONE  9,2
  2000000   OUTPUT FIELDS FOR RECORD TIMOUT FILE DSPFIL FORMAT TIMOUT.
  2000000         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      3900   RPG PROVIDES SEPARATE INDICATOR AREA FOR FILE INTFIL.
 5738RG1 V2R1M0  910524            IBM AS/400 RPG/400                 INTLIB/RSDINT        10/05/90  16:12:28      Page      14
                      C r o s s  R e f e r e n c e
 File and Record References:
        FILE/RCD    DEV/RCD     REFERENCES (D=DEFINED)
    02  DSPFIL      WORKSTN        4400D   11300    42800
          CIMENU                   4400D I000000     9800    12000    16900
                                  25500    30800    32300    36200    43800
                                 V000000
          DTLMNU                   4400D J000000    13900    36500  W000000
          DTLSCR                   4400D K000000    34800  X000000
          ITMMNU                   4400D L000000    13700    18300    20600
                                  26000    26500  Y000000
          ITMSC2                   4400D M000000    22100  Z000000
          ITMSC3                   4400D N000000    27800  1000000
          TIMOUT                   4400D O000000    42700  2000000
    01  INTFIL      WORKSTN        3900D    9100     9200    18000    31900
                                  38700    38800
          DETACH                   3900D C000000    38300    38500  P000000
          DTLREQ                   3900D G000000    31300  T000000
          DTLRSP                   3900D B000000
          EOS                      3900D D000000    43200  Q000000
          EVKREQ                   3900D E000000    40900  R000000
          ITMREQ                   3900D F000000    17400  S000000
          ITMRSP                   3900D A000000
          TIMER                    3900D H000000    17800    31700  U000000
    03  QPRINT      PRINTER        4600D   46200    47200    48101
 Field References:
        FIELD       ATTR    REFERENCES (M=MODIFIED D=DEFINED)
        *IN25       A(1)      18200    32100    43700
        *IN97       A(1)    I000001  J000001  K000001  L000001  M000001
                            N000001  O000001    25800
        *IN98       A(1)    I000002  J000002  K000002  L000002  M000002
                            N000002  O000002    16800    25400    30700
                              36100
        *IN99       A(1)    I000003  J000003  K000003  L000003  M000003
                            N000003  O000003    13500    16700    25300
                              30600    36000
        *PSSR       BEGSR      3900    45600D
* 7031  ABORT       TAG       38600D
        CHKAGN      TAG       42600D   43500
        CMID        A(10)      6700D    9300M    9500M   17200M   17900M
                              31200M   31800M   38200M   38400M   47100
                              48100
        CNAME       A(5)      34100M X000009D
        CSTM        P(9,2)    27500M 1000003D
        CSTTM       P(9,2)  A000014D   26800    27500
 5738RG1 V2R1M0  910524            IBM AS/400 RPG/400                 INTLIB/RSDINT        10/05/90  16:12:28      Page      15
        CSTTY       P(11,2) A000015D   27700
        CSTY        P(11,2)   27700M 1000004D
        CUSTN       A(6)      34000M X000001D
        CUSTNO      P(6,0)  B000002D G000001D J000004D   34000  T000001D
        DEPT        P(3,0)    34700M X000002D
        DESC        A(30)   A000003D   21300
        DLSTOR      P(6,0)  B000004D   34200
        DLSTR       P(6,0)    34200M X000003D
        DNAME       A(30)   B000003D   34100
        DSC         A(30)     21300M Z000001D
        DSLSM       P(9,0)    34300M X000004D
        DSLSTM      P(9,0)  B000005D   34300
        DSPM01      P(9,0)  B000006D   34400
        DSPM02      P(9,0)  B000007D   34500
* 7031  DSPM03      P(9,0)  B000008D
        DSPM1       P(9,0)    34400M X000005D
        DSPM2       P(9,0)    34500M X000006D
        DSPM3       P(9,0)  X000007D
        DSTTYD      P(11,0) B000009D   34600
        DSTYD       P(11,0)   34600M X000008D
```

Figure D-16 (Part 10 of 13). Source Program Example — RSDINT

```
        DTLIN       TAG       11800    30500D
        DTLRTN      TAG       11900    35900D
* 7031  DTOUT       TAG       33900D
        END         TAG       13500    16700    25300    30600    36000
                              37800    38100D   41000
        ENDPSR      ENDSR     45800    46000D
* 7031  ENTRY       TAG        9000D
        ERR         Z(5,0)     6400D   45800
        ERRCHK      BEGSR     18100    32000    42400D
        ERROR       TAG       17500    18700    31400    32700    37900D
                              44100
        EVDTL       TAG       31100D   32600
        EVKSR       BEGSR      9400     9600    40400D
* 7031  FILL01      A(240)     5900D
* 7031  FILL02      A(145)     6100D
* 7031  FILL03      A(240)     6500D
* 7031  FILL04      A(145)     7200D
* 7031  FILL1       A(56)   A000018D
* 7031  FILL2       A(57)   B000011D
        FMTNM       A(10)      6600D   46900
        FORCE       TAG       38900D   43300
        IDEPT       P(3,0)  B000010D   34700
        IODS        DS(415)    4400     5800D
        IOFB        DS(415)    3900     6200D
        ITEMNO      P(6,0)  A000002D F000001D L000004D   20500  S000001D
        ITMIN       TAG       11500    16600D   18600
* 7031  ITMOUT      TAG       20400D
        ITMRTN      TAG       11600    11700    25200D
        LIB         A(10)     40600M   40800M R000002D
* 7031  LOC         A(8)       6300D
* 7031  LOS         P(9,2)  A000017D
        LOSTS       P(9,2)    27200M 1000006D
* 7031  MAIN        TAG        9700D
        MAJCOD      A(2)       6900D   17500    18700    31400    32700
                              41000    46500    47500
        MAJMIN      A(4)       6800D   18600    32600    42500
        MENU        TAG       11400    13400D
 5738RG1 V2R1M0  910524             IBM AS/400 RPG/400                  INTLIB/RSDINT        10/05/90  16:12:28      Page      16
        MINCOD      A(2)       7000D   46700    47700
        MMERR       EXCPT     38000    46200
        OPTION      A(1)    I000004D   13600
        PGMID       A(10)     40500M   40700M R000001D
* 7031  PRO         P(5,2)  A000016D
        PROFIT      P(5,2)    27600M 1000005D
        PROFM       P(9,2)    26800D   26900M   27100    27100M   27600
        PR01        P(7,2)  A000009D   21800    27200
        PR05        P(7,0)  A000010D   21900
        PR1         P(7,2)    21800M Z000007D
        PR5         P(7,0)    21900M Z000008D
        QAVAIL      P(7,0)    20900D   21000M   21100M   21200M Z000002D
        QTYB        P(7,0)    21600M Z000005D
        QTYBO       P(7,0)  A000007D   21200    21600
        QTYH        P(7,0)    21500M Z000003D
        QTYLST      P(7,0)  A000004D   27200
        QTYO        P(7,0)    21400M Z000004D
        QTYOH       P(7,0)  A000005D   21000    21500
        QTYOO       P(7,0)  A000006D   21100    21400
        READRQ      TAG       11100D   12100    14100    17000    18400
                              20700    22200    25600    26100    26600
                              27900    30900    32400    34900    36300
                              36600    43900
* 7031  RECCUS      A(1)    B000001D
        RECER       EXCPT     37700    47200
        RECERR      TAG       18800    32800    37600D
        RECID       A(8)       6000D   11400    11500    11600    11700
                              11800    11900    25900    26400
        RECID2      A(8)       7100D   18800    32800    47900
* 7031  RECITM      A(1)    A000001D
        RETURN      A(6)      45700D   45900M   46000
        SLSM        P(9,2)    27300M 1000001D
        SLSTM       P(9,2)  A000012D   26800    27000    27100    27300
        SLSTY       P(11,2) A000013D   27400
        SLSY        P(11,2)   27400M 1000002D
        TIMRSP      A(1)    O000004D   42900    43000    43100
        TRY88       TAG       31500D   42900
        TRY89       TAG       17600D   43000
        UFR         P(5,2)    22000M Z000009D
        UFRT        P(5,2)  A000011D   22000
        UNITQ       A(2)    A000008D   21700
        UNT         A(2)      21700M Z000006D
```

Figure D-16 (Part 11 of 13). Source Program Example — RSDINT

```
* 7031  XITMIN      TAG       17300D
        *BLANK      LITERAL   40500    40600
        '        '  LITERAL   17900    31800
        '      '    LITERAL   45700
        '*CANCL'    LITERAL   45900
        'CIMENU  '  LITERAL   11400
        'DTLMNU  '  LITERAL   11800
        'DTLRSP'    LITERAL   32800
        'DTLSCR  '  LITERAL   11900
        'ICF00   '  LITERAL    9100     9300    31200    38200    38700
        'ICF01   '  LITERAL    9200     9500    17200    38400    38800
        'INTLIB  '  LITERAL   40800
        'ITMMNU  '  LITERAL   11500
        'ITMRSP'    LITERAL   18800
        'ITMSC2  '  LITERAL   11600    25900
 5738RG1 V2R1M0  910524             IBM AS/400 RPG/400                   INTLIB/RSDINT        10/05/90  16:12:28      Page      17
        'ITMSC3  '  LITERAL   11700    26400
        'RTDINTCL'  LITERAL   40700
        '0300'      LITERAL   18600    32600
        '0310'      LITERAL   42500
        '04'        LITERAL   17500    18700    31400    32700    41000
        '1'         LITERAL   13500    13600    16700    16800    18200
                              25300    25400    25800    30600    30700
                              32100    36000    36100    42900    43000
                              43700
        '2'         LITERAL   43100
        0           LITERAL   20900    27000
        000000      LITERAL   20500
        01285       LITERAL   45800
        100         LITERAL   26900
 Indicator References:
        INDICATOR  REFERENCES (M=MODIFIED D=DEFINED)
        *IN        I000001  I000002  I000003  J000001  J000002  J000003
                   K000001  K000002  K000003  L000001  L000002  L000003
                   M000001  M000002  M000003  N000001  N000002  N000003
                   O000001  O000002  O000003    13500    16700    16800
                     18200    25300    25400    25800    30600    30700
                     32100    36000    36100    43700
        LR           39000M
        OA            4600D   48101
* 7031  10           18000M   31900M
        25           18200    32100    43700
        46           27000M   27100
* 7031  66           32200M
* 7031  86           38700M   38800M   42800M
* 7031  87           11300M
        88           11200M   31600M   31900M   32000    42900
        89           11200M   17700M   18000M   18100    43000
* 7031  90
        97         I000001  J000001  K000001  L000001  M000001  N000001
                   O000001    25800
        98         I000002  J000002  K000002  L000002  M000002  N000002
                   O000002    16800    25400    30700    36100
        99         I000003  J000003  K000003  L000003  M000003  N000003
                   O000003    13500    16700    25300    30600    36000
     * * * * *  E N D  O F  C R O S S  R E F E R E N C E  * * * * *
 5738RG1 V2R1M0  910524             IBM AS/400 RPG/400                   INTLIB/RSDINT        10/05/90  16:12:28      Page      18
                       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:   23
         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 D-16 (Part 12 of 13). Source Program Example — RSDINT

```
5738RG1 V2R1M0  910524            IBM AS/400 RPG/400                INTLIB/RSDINT       10/05/90  16:12:28      Page      19
                        F i n a l   S u m m a r y
Message Count:  (by Severity Number)
           TOTAL    00     10     20     30     40     50
            25      25      0      0      0      0      0
Program Source Totals:
   Records . . . . . . . . . . :   481
   Specifications  . . . . . . :   234
   Table Records . . . . . . . :   0
   Comments  . . . . . . . . . :   247
PRM has been called.
Program RSDINT is placed in library INTLIB. 00 highest Error-Severity-Code.
        * * * * *   E N D   O F   C O M P I L A T I O N   * * * * *
```

Figure D-16 (Part 13 of 13). Source Program Example — RSDINT

RPG/400 Target Program for a Two-Session Inquiry

The following describes an RPG/400 target program for a two-session inquiry.

Program Files: The RPG/400 two-session 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 An AS/400 printer file used to print records, both sent and received, as well as major and minor ICF return codes.

DDS Source: The DDS source for the ICF file (CFILE) is illustrated in Figure D-17.

```
5714PW1 R01M00  880301                SEU SOURCE LISTING                        10/14/87 17:20:35               PAGE 1
SOURCE FILE . . . . . . .  QINTSRC/INTLIB
MEMBER  . . . . . . . . .  CFILE
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
    A*************************************************************
    A*                                                           *
    A*                          ICF FILE                         *
    A*            USED IN TARGET TWO 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
    A            PARTDS        80
    A            FILL4         64
```

Figure D-17. DDS Source for an ICF File Used by a Target Program

The DDS for the database file (PFILE) is illustrated in Figure D-18.

```
 5714PW1 R01M00  880301                 SEU SOURCE LISTING                          10/16/87 07:43:14              PAGE 1
 SOURCE FILE . . . . . . .  QINTSRC/INTLIB
 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 D-18. DDS Source for a Database File Used by a Target Program

ICF File Creation and Program Device Entry Definition: The command needed to create the ICF file is:

```
CRTICFF  FILE(INTLIB/CFILE)
  SRCFILE(INTLIB/QINTSRC)
  SRCMBR(CFILE)
  ACQPGMDEV(RQSDEV)
  TEXT("TARGET ICF FILE FOR TWO 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 example illustrated in Figure D-19 on page D-58. The ICF file used in the example is defined by the user, and uses externally described data formats. The reference numbers in the explanation below correspond to the numbers in the program example.

All output operations to the ICF file in the example are done using the write operation.

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: The continuation lines on the file specification for CFILE define the data structure name; for example, 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 the error section (section 5).

If a detach indication is received, control goes to section 6 of the program. Otherwise, the program goes to section 3. When a detach is received, indicator 44 is set on, as defined by the RCVDETACH DDS keyword in the ICF file.

3 If an error occurs (a major return code greater than 03 is returned from the read operation), the program goes to section 5. Otherwise, the program goes to section 4.

The program also tests to see whether the receive detach indicator (indicator 44) is set. If it is, the program goes to section 6.

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, a fail indication is sent to the remote program using a write operation in combination with a fail.

If an error occurs on the write operation (a major return code greater than 03), control passes to section 5.

If no error occurs on the write operation, the program returns to section 2.

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 section 6.

6 Control passes to this section whenever the program has detected a communication error or has received a detach indication from the source program. The last record indicator is set on, which ends the program. CFILE is implicitly closed.

7 The subroutine *PSSR is called for I/O operation errors that are not handled by the subroutine in section 6. This subroutine checks to see 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.

```
 5738RG1 V2R1M0  910524            IBM AS/400 RPG/400                  INTLIB/RTDINT        10/05/90  16:13:08      Page       1
 Compiler . . . . . . . . . . . . . :   IBM AS/400 RPG/400
 Command Options:
   Program  . . . . . . . . . . . . :   INTLIB/RTDINT
   Source file  . . . . . . . . . . :   INTLIB/QINTSRC
   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 . . . . . . . . . . . . . :   RTDINT
   File . . . . . . . . . . . . . . :   QINTSRC
   Library  . . . . . . . . . . . . :   INTLIB
   Last Change  . . . . . . . . . . :   10/05/90  15:27:19
   Description  . . . . . . . . . . :   RPG Target Intra Program Example
 5738RG1 V2R1M0  910524            IBM AS/400 RPG/400                  INTLIB/RTDINT        10/05/90  16:13:08      Page       2
 SEQUENCE                                                                      IND    DO    LAST       PAGE   PROGRAM
 NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE     LINE   ID
                     S o u r c e   L i s t i n g
      100  H*****************************************************************                  10/13/87
      200  H*  THIS PROGRAM WILL HANDLE THE REQUEST FOR EITHER A BUYER     *                  03/17/89
      300  H*  NUMBER OR AN ITEM NUMBER.  THIS IS ACCOMPLISHED BY MAKING   *                  03/17/89
      400  H*  THE DATA BASE FILE STRUCTURE (KEY LENGTH, KEY POSITION,     *                  03/17/89
      500  H*  RECORD LENGTH, RECORD SIZE, ETC.) THE SAME FOR BOTH FILES   *                  03/17/89
      600  H*  WITH ONLY THE RECORD CONTENTS DIFFERENT.                    *                  03/17/89
      700  H*                                                             *                  03/17/89
      800  H*  THIS PROGRAM ENDS WHEN A DETACH REQUEST IS RECEIVED FROM    *                  03/17/89
      900  H*  THE SOURCE PROGRAM.                                        *                  03/17/89
     1000  H*****************************************************************                  10/13/87
     1100   * 1                                                                               03/17/89
            H                                                                                          *****
     1200  FCFILE   CF  E                    WORKSTN                                         10/13/87
     1300  F                                            KINFDS FEEDBK                       10/13/87
     1400  F                                            KINFSR *PSSR                        10/14/87
            RECORD FORMAT(S):  LIBRARY INTLIB FILE CFILE.
                     EXTERNAL FORMAT SNDPART RPG NAME SNDPART
                     EXTERNAL FORMAT RCVPART RPG NAME RCVPART
     1500  FPFILE   IF  E           K        DISK                                            10/13/87
            RECORD FORMAT(S):  LIBRARY INTLIB FILE PFILE.
                     EXTERNAL FORMAT DBREC RPG NAME DBREC
     1600  FQPRINT  O   F     132            PRINTER                                         03/17/89
  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 DBREC FILE PFILE FORMAT DBREC.
  C000001                                      1   1 RECCUS
  C000002                                      2   7 DBSEQ
  C000003                                      8 137 DBDATA
  C000004                                    138 150 DBFILL
```

Figure D-19 (Part 1 of 5). Target Program Example —RTDINT

```
    1700  IFEEDBK      DS                                                        10/13/87
    1800  I                                    *ROUTINE LOC                       10/14/87
    1900  I                                    *STATUS  ERR                       10/14/87
    2000  I                                      38  47 FMTNM                     10/05/90
    2100  I                                     273 282 PGMDEV                    10/13/87
    2200  I                                     401 404 MAJMIN                    10/13/87
    2300  I                                     401 402 MAJCOD                    10/13/87
    2400  I                                     403 404 MINCOD                    10/13/87
    2500  C*****************************************************************     03/17/89
5738RG1 V2R1M0  910524            IBM AS/400 RPG/400                 INTLIB/RTDINT       10/05/90  16:13:08      Page       3
SEQUENCE                                                                          IND    DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE     LINE   ID
    2600  C*                                                              *       03/17/89
    2700  C*  READ THE REQUEST FROM THE SOURCE PROGRAM.  INDICATOR 40     *       03/17/89
    2800  C*  INDICATES RCVTRNRND OCCURRED.  INDICATOR 44 INDICATES THAT  *       10/05/90
    2900  C*  DETACH HAS BEEN RECEIVED.                                   *       03/17/89
    3000  C*                                                              *       03/17/89
    3100  C*  INDICATOR 99 WILL BE TURNED ON FOR "I/O ERRORS" THEREBY     *       03/17/89
    3200  C*  PREVENTING THE RPG DEFAULT ERROR HANDLER FROM BEING CALLED. *       03/17/89
    3300  C*  THIS IS NECESSARY TO ALLOW THE PROGRAM TO PROCESS THE       *       03/17/89
    3400  C*  ICF MAJOR/MINOR RETURN CODE.  THIS PROGRAM CHECKS           *       10/05/90
    3500  C*  FOR ERRORS ON EVERY ICF FILE OPERATION.  A MAJOR            *       10/05/90
    3600  C*  CODE GREATER THAN 03 INDICATES AN ERROR.                    *       03/17/89
    3700  C*                                                              *       03/17/89
    3800  C*****************************************************************     03/17/89
    3900   * 2                                                                    03/17/89
    4000  C           READ      TAG                                               10/13/87
    4100  C                     READ RCVPART               9950                2 3 10/13/87
    4200  C           MAJCOD    CABGT'03'      ERROR                              10/13/87
    4300  C           *IN44     CABEQ'1'       END              DET RECV?         10/13/87
    4400  C                     MOVE RECID2    DBSEQ                              10/13/87
    4500  C           MAJMIN    CABEQ'0000'    XMIT             RCVTRNRND?        10/13/87
    4600  C           *IN40     CABEQ'1'       XMIT             RCVTRNRND?        10/13/87
    4700  C                     GOTO READ                       NO,TRY AGAIN      03/17/89
    4800  C*****************************************************************     03/17/89
    4900  C*                                                              *       03/17/89
    5000  C*  A REQUEST FROM THE SOURCE PROGRAM RESULTS IN READING A      *       03/17/89
    5100  C*  SINGLE RECORD CONTAINING THE REQUESTED BUYER OR ORDER       *       03/17/89
    5200  C*  NUMBER.  THE RESPONSE WILL BE RETURNED IN A SINGLE RECORD   *       03/17/89
    5300  C*  CONTAINING EITHER THE ITEM OR BUYER INFORMATION, DEPENDING  *       03/17/89
    5400  C*  ON THE DATA BASE CONTENT.                                   *       03/17/89
    5500  C*                                                              *       03/17/89
    5600  C*  THE RESPONSE IS SENT TO THE SOURCE PROGRAM BY WRITING TO    *       03/17/89
    5700  C*  THE ICF FILE USING FORMAT SNDPART.                         *       10/05/90
    5800  C*                                                              *       03/17/89
    5900  C*****************************************************************     03/17/89
    6000   * 3                                                                    03/17/89
    6100  C           XMIT      TAG                                               10/13/87
    6200  C           DBSEQ     CHAINPFILE            9897   98 IF NOT FD   1 2  11/21/88
    6300  C                     MOVE DBSEQ     ITEMNO                             10/13/87
    6400  C                     MOVE RECCUS    RECTYP            RECD FMT ID      10/13/87
    6500  C*****************************************************************     03/17/89
    6600  C*                                                              *       03/17/89
    6700  C*  WHEN THE REQUESTED BUYER OR ITEM NUMBER IS NOT FOUND,       *       03/17/89
    6800  C*  000000 IS PROPAGATED TO THE KEY FIELD BEFORE THE RESPONSE   *       03/17/89
    6900  C*  IS SENT BACK TO THE SOURCE PROGRAM.                         *       03/17/89
    7000  C*  WHEN A DISK I/O OPERATION COMPLETES UNSUCCESSFULLY, A FAIL  *       03/17/89
    7100  C*  INDICATION IS SENT.                                         *       03/17/89
    7200  C*                                                              *       03/17/89
    7300  C*****************************************************************     03/17/89
    7400   * 4                                                                    03/17/89
    7500  C   98                MOVE '000000'  ITEMNO                             10/13/87
    7600  C   98                MOVE '1'       *IN20            SEND FAIL         11/21/88
    7700  C                     MOVELDBDATA    EDATA            MOVE  DATA        10/13/87
```

Figure D-19 (Part 2 of 5). Target Program Example —RTDINT

```
5738RG1 V2R1M0  910524            IBM AS/400 RPG/400                 INTLIB/RTDINT        10/05/90  16:13:08      Page       4
SEQUENCE                                                                                    IND    DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE     LINE   ID
    7800  C   97              MOVE '1'       *IN20           SEND FAIL                    11/21/88
    7900  C                   WRITESNDPART                   DATA W/DET                   10/13/87
    8000  C           MAJCOD  CABGT'03'      ERROR                                        10/13/87
    8100  C                   MOVE '0'       *IN20           RESET IND                    11/21/88
    8200  C                   GOTO READ                                                   10/13/87
    8300  C*                                                                              10/13/87
    8400  C*************************************************************                 03/17/89
    8500  C*                                                           *                 03/17/89
    8600  C* IF ANY ICF FILE ERROR OCCURS, PRINT THE ERROR MESSAGE, AND   *                 10/05/90
    8700  C* THEN END THE JOB.                                          *                 10/05/90
    8800  C*                                                           *                 03/17/89
    8900  C*************************************************************                 03/17/89
    9000   * 5                                                                           03/17/89
    9100  C           ERROR   TAG                                                         10/13/87
    9200  C                   EXCPTMMERR                                                  03/17/89
    9300  C           END     TAG                                                         10/13/87
    9400   * 6                                                                           03/17/89
    9500  C                   SETON                    LR                  1          10/13/87
    9600  C                   RETRN                                                       10/13/87
    9700  C*************************************************************                 10/14/87
    9800  C*                                                           *                 03/17/89
    9900  C*    THIS IS THE PROGRAM EXCEPTION/ERROR SUBROUTINE THAT     *                 03/17/89
   10000  C*    RECEIVES CONTROL WHEN AN EXCEPTION OR ERROR OCCURS      *                 03/17/89
   10100  C*    AFTER AN I/O IS ISSUED TO AN ICF PROGRAM DEVICE AND     *                 03/17/89
   10200  C*    THERE IS A NON-ZERO VALUE UPDATED IN THE RPG STATUS     *                 03/17/89
   10300  C*    FIELD (ERR).  THIS ROUTINE CHECKS FOR STATUS VALUES THAT *                 03/17/89
   10400  C*    RELATE TO ICF OPERATION.                                *                 03/17/89
   10500  C*    IF THE PROGRAM DEVICE IS ALREADY ACQUIRED, THE EXCEPTION *                 03/17/89
   10600  C*    IS IGNORED, OTHERWISE THE PROGRAM IS TERMINATED.        *                 03/17/89
   10700  C*                                                           *                 03/17/89
   10800  C*************************************************************                 10/14/87
   10900   * 7                                                                           03/17/89
   11000  C           *PSSR   BEGSR                                                       10/14/87
   11100  C                   MOVE '      '  RETURN  6        DEFAULT                   10/14/87
   11200  C           ERR     CABEQ01285     ENDPSR           ALREADY ACQ?              10/14/87
   11300  C                   MOVE '*CANCL'  RETURN           JOB ENDS                  10/14/87
   11400  C           ENDPSR  ENDSRRETURN                    BACK TO MAIN              10/14/87
   11500  C*************************************************************                 10/13/87
   11600  OQPRINT  E 1            MMERR                                                  10/13/87
   11700  O                                 21 'ERROR ON ICF FILE'                       10/05/90
   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
* 6103     12501   OVERFLOW INDICATOR OA ASSIGNED TO FILE QPRINT.
  D000000   OUTPUT FIELDS FOR RECORD SNDPART FILE CFILE FORMAT SNDPART.
  D000001                         RECTYP     1  CHAR    1
  D000002                         ITEMNO     7  CHAR    6
5738RG1 V2R1M0  910524            IBM AS/400 RPG/400                 INTLIB/RTDINT        10/05/90  16:13:08      Page       5
SEQUENCE                                                                                    IND    DO    LAST       PAGE   PROGRAM
NUMBER    *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  USE    NUM   UPDATE     LINE   ID
  D000003                         EDATA    137  CHAR  130
  D000004                         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      1200   RPG PROVIDES SEPARATE INDICATOR AREA FOR FILE CFILE.
5738RG1 V2R1M0  910524            IBM AS/400 RPG/400                 INTLIB/RTDINT        10/05/90  16:13:08      Page       6
               K e y  F i e l d  I n f o r m a t i o n
                    PHYSICAL    LOGICAL
       FILE/RCD     FIELD       FIELD       ATTRIBUTES
   02  PFILE
           DBREC
                    DBSEQ                   CHAR    6
```

Figure D-19 (Part 3 of 5). Target Program Example —RTDINT

```
5738RG1 V2R1M0  910524            IBM AS/400 RPG/400                 INTLIB/RTDINT       10/05/90  16:13:08      Page       7
                      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        1200D
         RCVPART                  1200D B000000     4100
         SNDPART                  1200D A000000     7900  D000000
   02  PFILE       DISK           1500D    6200
         DBREC                    1500D C000000
   03  QPRINT      PRINTER        1600D   11600    12501
Field References:
       FIELD       ATTR    REFERENCES (M=MODIFIED D=DEFINED)
       *IN20       A(1)       7600M    7800M    8100M
       *IN40       A(1)       4600
       *IN44       A(1)       4300
       *PSSR       BEGSR      1200    11000D
       DBDATA      A(130)  C000003D    7700
* 7031  DBFILL      A(13)   C000004D
       DBSEQ       A(6)    C000002D    4400M    6200     6300
       EDATA       A(130)  A000003D    7700M D000003D
       END         TAG        4300     9300D
       ENDPSR      ENDSR     11200    11400D
       ERR         Z(5,0)     1900D   11200
       ERROR       TAG        4200     8000     9100D
       FEEDBK      DS(404)    1200     1700D
       FILL1       A(13)   A000004D D000004D
* 7031  FILL4       A(64)   B000003D
       FMTNM       A(10)      2000D   12300
       ITEMNO      A(6)    A000002D    6300M    7500M D000002D
* 7031  LOC         A(8)       1800D
       MAJCOD      A(2)       2300D    4200     8000    11900
       MAJMIN      A(4)       2200D    4500
       MINCOD      A(2)       2400D   12100
       MMERR       EXCPT      9200    11600
* 7031  PARTDS      A(80)   B000002D
       PGMDEV      A(10)      2100D   12500
       READ        TAG        4000D    4700     8200
       RECCUS      A(1)    C000001D    6400
       RECID2      A(6)    B000001D    4400
       RECTYP      A(1)    A000001D    6400M D000001D
       RETURN      A(6)      11100D   11300M   11400
       XMIT        TAG        4500     4600     6100D
       '      '    LITERAL   11100
       '*CANCL'    LITERAL   11300
       '0'         LITERAL    8100
       '0000'      LITERAL    4500
5738RG1 V2R1M0  910524            IBM AS/400 RPG/400                 INTLIB/RTDINT       10/05/90  16:13:08      Page       8
       '000000'    LITERAL    7500
       '03'        LITERAL    4200     8000
       '1'         LITERAL    4300     4600     7600     7800
       01285       LITERAL   11200
Indicator References:
       INDICATOR  REFERENCES (M=MODIFIED D=DEFINED)
       *IN           4300     4600     7600M    7800M    8100M
       LR            9500M
       OA            1600D   12501
* 7031  05
* 7031  10
* 7031  15
       20            7600M    7800M    8100M
* 7031  25
* 7031  30
       40            4600
       44            4300
* 7031  50            4100M
       97            6200M    7800
       98            6200M    7500     7600
* 7031  99            4100M
     * * * * *  E N D  O F  C R O S S  R E F E R E N C E  * * * * *
```

Figure D-19 (Part 4 of 5). Target Program Example —RTDINT

```
5738RG1 V2R1M0  910524            IBM AS/400 RPG/400                 INTLIB/RTDINT        10/05/90  16:13:08      Page       9
                       M e s s a g e   S u m m a r y
* QRG6103 Severity:  00   Number:    1
          Message . . . . :   No Overflow Indicator is specified but an
            indicator is assigned to a file and automatic skip to 6 is
            generated.
* QRG7031 Severity:  00   Number:   11
          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   * * * * *
5738RG1 V2R1M0  910524            IBM AS/400 RPG/400                 INTLIB/RTDINT        10/05/90  16:13:08      Page       10
                         F i n a l   S u m m a r y
 Message Count:  (by Severity Number)
            TOTAL    00     10     20     30     40     50
              13     13      0      0      0      0      0
 Program Source Totals:
    Records . . . . . . . . . . :   125
    Specifications  . . . . . . :   53
    Table Records . . . . . . . :   0
    Comments  . . . . . . . . . :   72
 PRM has been called.
 Program RTDINT is placed in library INTLIB. 00 highest Error-Severity-Code.
          * * * * *   E N D   O F   C O M P I L A T I O N   * * * * *
```

Figure D-19 (Part 5 of 5). Target Program Example —RTDINT

Bibliography

The publications listed here provide additional information about topics described or referred to in this book. These books are listed with their full title and order number.

AS/400 Books

The following AS/400 books contain additional information you may need when developing application programs that use intrasystem communications support.

- *ICF Programming*, SC41-5442, provides the application programmer with information needed to write programs that use AS/400 communications and the OS/400 intersystem communications function (OS/400-ICF).
- *Communications Configuration*, SC41-5401, contains general configuration information, including detailed descriptions of network interface, line, controller, device, mode, and class-of-service descriptions, configuration lists and connection lists.
- *DDS Reference*, SC41-5712, contains information about coding data description specifications for files.
- *C/400* User's Guide*, SC09-1347, provides the information needed to write, test, and maintain C/400 programs for 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.
- *RPG/400 User's Guide*, SC09-1816, provides the information needed to write, test, and maintain RPG/400 programs for the AS/400 system.
- *CL Reference*, SC41-5722, contains the commands, command parameters, and syntax for the commands used in this book.
- *Work Management*, SC41-5306, contains information about how to create an initial work management environment and how to change it.
- *System Operation*, SC41-4203, provides information on how to use the system unit operator display.

System/36 Communications Books

The following book provides a description of the Intra Subsystem on the System/36, and information about setting up and configuring the Intra Subsystem, communications operations, and return codes:

- *Interactive Communications Feature: Programming for Subsystems and Intra Subsystem Reference*, SC21-9533.

The following two books provide information and examples about the interactive communications feature, a feature of the System Support Program Product on the System/36 that allows a program to communicate interactively with another program or system:

- *Interactive Communications Feature: Base Subsystems Reference*, SC21-9530.
- *Interactive Communications Feature: Guide and Examples*, SC21-7911.

The following book provides an overview for programming in the System/36 environment:

- *System/36 Environment Programming*, SC41-4730.

Index

D

E

F

G

I

J

K

L

M

N

O

P

R

S

T

V

W

Reader Comments—We'd Like to Hear from You!

AS/400
Intrasystem Communications
Programming
Version 4

Publication No. SC41-5447-00

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

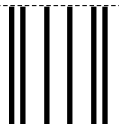
Company or Organization

Phone No.

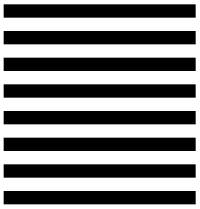
Reader Comments—We'd Like to Hear from You!
SC41-5447-00

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

SC41-5447-00

Cut or Fold
Along Line

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC41-5447-00

Spine information:

AS/400

Intrasystem Communications Programming

Version 4