

Application System/400™

SC21-9593-1

**Communications:
BSC Equivalence Link Programmer's Guide**





Application System/400™

SC21-9593-1

**Communications:
BSC Equivalence Link Programmer's Guide**

Second Edition (September 1989)

This major revision makes obsolete SC21-9593-0.

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

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

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

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

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM licensed program in this publication is not intended to state or imply that only IBM's licensed program may be used. Any functionally equivalent program may be used instead.

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

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

This publication could contain technical inaccuracies or typographical errors. To report an error, use the Reader's Comment Form at the back of the publication. If the form has been removed, address your comments to: IBM Corporation, Information Development, Department 245, Rochester, Minnesota, U.S.A. 55901. IBM reserves the right to use or distribute the information you supply in any way it believes appropriate without incurring any obligation to you.

Application System/400, AS/400, Operating System/400, OS/400, C/400, COBOL/400, and RPG/400 are trademarks of the International Business Machines Corporation.

400 is a registered trademark of the International Business Machines Corporation.

© Copyright International Business Machines Corporation 1988, 1989. All rights reserved.

About This Guide

This guide supplies the programming information you need to use the Binary Synchronous Communications Equivalence Link (BSCCL) with the IBM AS/400 system. This guide and the *Communications Programmer's Guide* are intended to be used together. You should be familiar with the concepts explained in the *Communications Programmer's Guide* and then apply those concepts to the detailed information presented here for using BSCCL.

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

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

Who Should Use This Guide

This guide is intended for system and application programmers who write communications programs that use BSCCL. It has information for IBM AS/400 programmers and for programmers on other systems and devices that communicate with BSCCL on the AS/400 system.

What You Should Know

You should be able to program in the language you use and be familiar with the following information:

- General data communications information, described in *Data Communications Concepts*, GC21-5169, Chapters 1 through 4
- AS/400 system programming terminology, mainly work station terminology
- Terminology of the remote system
- Binary synchronous communications as described in the *General Information—Binary Synchronous Communications*, GA27-3004
- Communications configuration information described in the *Communications: User's Guide*, SC21-9601
- Intersystem communications function (ICF) support described in the *Communications: Programmer's Guide*, SC21-9590
- Data management concepts described in the *Programming: Data Management Guide*, SC21-9658

How This Guide Is Organized

The following list provides a brief description of the information in each chapter:

Chapter 1 describes the remote systems, communications lines, and data formats you can use with BSCCL on the AS/400 system.

Chapter 2 describes the commands used for configuring BSCCL.

Chapter 3 describes how to activate and deactivate the line, controller, and device descriptions used by BSCCL.

Chapter 4 describes the BSCCL parameters used to define an ICF file, and the language operations and communications functions that you use when you write a communications application that uses BSCCL.

Chapter 5 provides programming considerations for the remote system and AS/400 programmer.

Appendix A contains tables of language operations, DDS keywords, and system-supplied formats supported by BSCCL.

Appendix B contains descriptions of the communications return codes that BSCCL returns to a program after each language operation.

Appendix C contains a list of BSC control codes and protocols.

Appendix D contains application flow examples for communications between the AS/400 system and another system using BSCCL.

Appendix E provides program explanations and examples for COBOL/400 and RPG/400.

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

How This Guide Has Changed

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

- A rearrangement of the manual, as described in the section "How This Guide Is Organized"
- Selective prompting for the CMNTYPE parameter on the program device entry commands, described in Chapter 4, "Writing Application Programs"
- A new section in Chapter 4, "Writing Application Programs," explaining return code processing for your program
- Information about prestart jobs, noted in Chapter 5, "BSCCL Considerations"
- A new section in Chapter 5, "BSCCL Considerations," explaining programming considerations for the reverse interrupt indicator

Related Online Information

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

Help for Displays

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

- Field
- Extended

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

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

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

Index Search

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

Help for Control Language Commands

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

Online Education

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

Question-and-Answer Function

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

Related Printed Information

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

The following AS/400 manuals contain additional information you may need when you use BSCEL.

- *Communications: Programmer's Guide*, SC21-9590, supplies the application programmer with information needed to write communications programs on the AS/400 system.
- *Communications: User's Guide*, SC21-9601, supplies communications information that is common in the communications support for the AS/400 system. This includes setting and changing communications values and starting and stopping communications.
- *General Information—Binary Synchronous Communications*, GA27-3004, describes binary synchronous communications.
- *Information Directory*, GC21-9678, lists the manuals in the AS/400 library, lists the tasks that are described in the manuals, and supplies a master glossary of AS/400 terms.
- *Languages: C/400 User's Guide*, SC09-1303, provides information regarding the C/400 programming language.
- *Languages: COBOL/400 User's Guide*, SC09-1158, provides information regarding the COBOL/400 programming language.
- *Languages: RPG/400 User's Guide*, SC09-1161, provides information regarding the RPG/400 programming language.
- *Programming: Control Language Reference*, SBOF-0481, contains the commands, command parameters and syntax for the commands used in this manual.
- *Programming: Data Description Specifications Reference*, SC21-9620, contains information about coding data description specifications for files.
- *Programming: Data Management Guide*, SC21-9658, contains information about the concepts and structure of data management on the system.

Other Related Manuals

If you are using System/36-compatible RPGII BSCA telecommunications programming, refer to the *Languages: System/36 Compatible RPG/II User's Guide and Reference*, SC09-1162, for information about specific commands and parameters needed for this application.

Contents

Chapter 1. Introduction to the Binary Synchronous Communications	
Equivalence Link	1-1
Remote Systems Supported	1-1
Communications Lines Supported	1-2
Data Formats Supported	1-3
Chapter 2. Configuring BSCEL	2-1
Defining the BSCEL Configuration	2-1
Chapter 3. Running BSCEL	3-1
Vary On and Vary Off Support	3-1
Chapter 4. Writing Application Programs	4-1
Using an Intersystem Communications Function File	4-1
Specifying the Program Device Entry Commands	4-2
Comparing the Program Device Entry Command and Configuration Parameters	4-8
Determining Record Lengths	4-9
Starting a Session	4-10
Open/Acquire Operation	4-10
Starting a Transaction	4-11
Evoke Function	4-11
RMTBSCEL Considerations	4-12
Security Function	4-13
Sending Data	4-13
Write Operation	4-13
Subdevice-Selection Function	4-14
End-of-Group Function	4-15
Variable-Length-Data Function	4-15
Receiving Data	4-15
Read Operation	4-15
Invite Function	4-16
Read-from-Invited-Program-Devices Operation	4-16
Record-Identification Function	4-16
Variable-Length-Data Function	4-16
Notifying the Remote Program of Problems	4-16
Fail Function	4-17
Using Additional Functions and Operations	4-17
Request-to-Write Function	4-17
Allow-Write Function	4-18
Cancel-Invite Function	4-18
Timer Function	4-18
Get-Attributes Operation	4-19
Ending a Transaction	4-19
Detach Function	4-19
Ending a Session	4-20
Release Operation	4-20
End-of-Session Function	4-20
Using Response Indicators	4-21
Receive-End-of-Group	4-21
Receive-Turnaround	4-21
Receive-Detach	4-22
Using the Input/Output Feedback Area	4-22

Using Return Codes	4-22
Chapter 5. BSCCL Considerations	5-1
Program Start Request Considerations	5-1
Syntax for Program Start Requests	5-1
Formats for Program Start Requests	5-4
Continuation Format for Program Start Requests	5-5
Examples of Program Start Requests	5-5
Program Start Requests Received from Remote Systems	5-6
Program Start Requests Sent by the AS/400 System	5-7
Failed Program Start Requests	5-8
Prestarting Jobs for Program Start Requests	5-10
Remote System Considerations	5-11
BSCCL Commands	5-11
Receiving Null Records	5-12
Using 3740 Data Entry Systems	5-12
Using Office Systems	5-13
Programming Considerations	5-15
BSCCL Online Messages	5-15
The Reverse-Interrupt Indicator	5-16
Appendix A. Language Operations, DDS Keywords, and System-Supplied	
Formats	A-1
Language Operations	A-1
Supported DDS Keywords	A-3
System-Supplied Formats	A-4
Appendix B. Using BSCCL Return Codes	B-1
Major Code 00	B-1
Major Code 02	B-4
Major Code 03	B-6
Major Code 04	B-8
Major Codes 08-11	B-9
Major Code 34	B-10
Major Code 80	B-11
Major Code 81	B-14
Major Code 82	B-20
Major Code 83	B-29
Appendix C. BSC Control Codes and Protocols	C-1
Appendix D. BSCCL Application Flow Examples	D-1
Appendix E. Example Programs	E-1
COBOL/400 Source Program for Local System	E-1
Configuration	E-1
Program Files	E-2
Program Explanation	E-6
COBOL/400 Source Program for Remote System	E-15
Configuration	E-15
Program Files	E-15
Program Explanation	E-19
RPG/400 Source Program for Local System	E-26
Configuration	E-26
Program Files	E-26
Program Explanation	E-30

RPG/400 Source Program for Remote System	E-37
Configuration	E-37
Program Files	E-37
Program Explanation	E-41
Glossary	G-1
Index	X-1

Figures

1-1.	BSCCEL Example Communications Network Configuration	1-3
5-1.	Syntax for Program Start Requests	5-2
5-2.	*EXEC Format for Program Start Request	5-6
5-3.	Continuation Format for Program Start Request	5-6
D-1.	Communications between Two AS/400 Application Programs	D-2
D-2.	Communications between an AS/400 System and a Remote Device Not Using BSCCEL	D-3
D-3.	Communications between an AS/400 System and a Remote System Using Multiple Files	D-4
D-4.	Starting an AS/400 System from a Remote System	D-5
D-5.	An AS/400 System Receiving Data from an Office System	D-7
D-6.	An AS/400 System Sending Data to an Office System	D-8
E-1.	Communication between a Local AS/400 Program and a Remote AS/400 Program	E-1
E-2.	DDS for the BSCCELICF File Used by the Local System (COBOL/400)	E-3
E-3.	DDS for the Database File Used in BSCCEL Data Transfer on the Local System (COBOL/400)	E-4
E-4.	DDS for the Printer File on the Local System (COBOL/400)	E-5
E-5.	COBOL/400 Source Program for the Local System	E-8
E-6.	DDS for the BSCCELICF Program File Used by the Remote System (COBOL/400)	E-16
E-7.	DDS for the Database File Used in BSCCEL Data Transfer Program on the Remote System (COBOL/400)	E-17
E-8.	DDS for the Printer File on the Remote System (COBOL/400)	E-18
E-9.	COBOL/400 Source Program for the Remote System	E-20
E-10.	DDS for the BSCCELICF File Used by the Local System (RPG/400)	E-27
E-11.	DDS for the Database File Used in BSCCEL Data Transfer on the Local System (RPG/400)	E-28
E-12.	DDS for the Printer File on the Local System (RPG/400)	E-29
E-13.	RPG/400 Source Program for the Local System	E-32
E-14.	DDS for the BSCCELICF Program File Used by the Remote System (RPG/400)	E-38
E-15.	DDS for Database File Used in BSCCEL Data Transfer Programs on the Remote System (RPG/400)	E-39
E-16.	DDS for the Printer File on the Remote System (RPG/400)	E-40
E-17.	RPG/400 Source Program for the Remote System	E-42

Tables

4-1.	BSC/CL Control Characters That Cannot Be Used as Record Separator Characters	4-5
4-2.	Comparing Configuration Command Parameters and Program Device Entry Command Parameters	4-8
5-1.	Coding Positions for a Program Start Request	5-3
5-2.	Continuation Format for Program Start Requests	5-5
5-3.	Session Activity	5-7
5-4.	Reason Codes for Rejected Program Start Requests	5-8
A-1.	Language Operations	A-1
A-2.	Language Operations	A-2
A-3.	Valid DDS Keywords for BSC/CL	A-3
A-4.	Valid System-Supplied Formats for BSC/CL	A-4
B-1.	Actions for Return Code 0000	B-2
C-1.	BSC Line Protocols and Control Characters	C-1

Chapter 1. Introduction to the Binary Synchronous Communications Equivalence Link

This chapter is an introduction to the Binary Synchronous Communications Equivalence Link (BSC¹EL). It lists the remote systems, communications lines, and data formats supported by BSC¹EL and presents an example showing an AS/400¹ configuration of a BSC¹EL communications network.

Remote Systems Supported

BSC¹EL supplies distributed data processing support to AS/400 system users who want to communicate with a remote system or device using binary synchronous communications (BSC). It also supplies online and batch communications between application programs on different BSC systems.

BSC¹EL communicates with another AS/400 system that has BSC or BSC¹EL support. Use an AS/400 system with BSC¹EL support to communicate with any of the following IBM systems:

- AS/400 system with BSC¹EL support
- System/38 with BSC support
- System/36 with SSP-ICF BSC¹EL support or with SSP BSC telecommunications support
- System/34 with SSP-ICF BSC¹EL support or with SSP BSC telecommunications support
- Systems using BSC to communicate as if they were System/3s:
 - Series/1
 - System/7 with Modular System Program/7 (MSP/7)
- Systems using BSC to communicate as if the AS/400 system were a System/3 terminal:
 - OS/VS, or DOS/VSE with Basic Telecommunications Access Method (BTAM)
 - IBM 3705, 3720, 3725 using Network Control Programs (NCP), including emulation programs and partitioned emulation programs
- IBM 3741 Model 2 Data Station or Model 4 Programmable Work Station
- Systems or devices using BSC to communicate as if they were 3741 devices:
 - IBM 5110 or 5120 Computer
 - IBM 5231 Data Collection Controller Model 2 (in transmit mode only)
 - IBM 5260 Point-of-Sale Terminal
 - IBM 5280 Distributed Data System
- Systems or devices using BSC to communicate as office systems:
 - IBM 6640 Document Printer
 - IBM 6670 Information Distributor
 - 430, 440, 442, 450, and 452 Information Processors
 - IBM 5520 Administrative System

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

- IBM 6580 Displaywriter System
- IBM 3747 Data Converter
- IBM 3750 Switching System (except in the United States and Canada)

BSCCEL allows you to communicate with any of these remote systems or devices by handling the data passed between the AS/400 system and the remote system. BSCCEL does not distinguish between any of these remote systems or devices in the way it handles the data passed between the AS/400 system and the remote system. Using BSCCEL, you can start a program on the remote system or the remote system can start a program on the local AS/400 system.

The AS/400 system supplies a configuration option for BSCCEL to handle 3740 multiple files or office systems documents. BSCCEL handles any communications line protocol needed to connect your AS/400 system to the remote system.

The user program must handle any unique considerations for remote system data streams.

Communications Lines Supported

The AS/400 system can have more than one BSC communications line active at the same time. Each communications line is identified by one of the following types (all lines do not have to be the same type):

- Point-to-point switched
 - Manual answer
 - Automatic answer
 - Manual dial
 - Automatic dial
- Point-to-point nonswitched
- Multipoint tributary

By using point-to-point communications lines:

- BSCCEL supports only one communications session at a time on each line.
- BSCCEL cannot share a communications line with another communications program.

By using multipoint tributary communications lines:

- BSCCEL supports more than one communications session on each line. (Each communications session is associated with a different BSC device description.)
- BSCCEL can share a communications line with another communications program.

A communications program can conduct more than one session at the same time on a multipoint tributary line if each session uses a different BSC device description. A communications program can conduct more than one session at the same time, each on a different point-to-point line, if each session uses a different BSCCEL configuration.

A communications program can also conduct sessions consecutively on one line. After one session ends, another session on the same line can be started imme-

diately. However, in any session, only one transaction is active at a time. One transaction must end before another one begins. (Transactions are discussed in "Starting a Transaction" on page 4-11.)

Figure 1-1 is an example of a BSC/CEL network that connects the AS/400 system and remote systems with various line types. (For information on creating a BSC/CEL configuration, see Chapter 2, "Configuring BSC/CEL" on page 2-1, and the *Communications User's Guide*.)

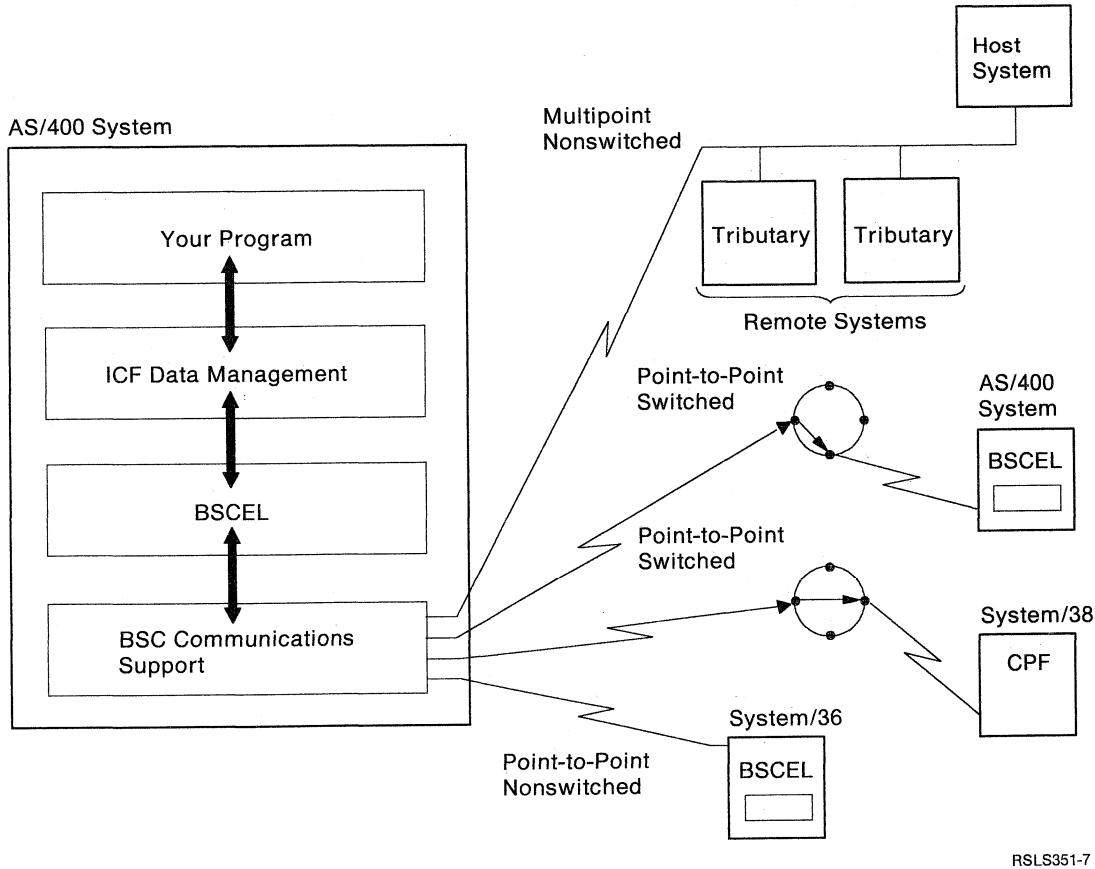


Figure 1-1. BSC/CEL Example Communications Network Configuration

Data Formats Supported

BSC/CEL supports the following data formats. (For information about data formats and how they are specified, see "Comparing the Program Device Entry Command and Configuration Parameters" on page 4-8 and "Specifying the Program Device Entry Commands" on page 4-2.)

- Unblocked data records in either transparent or nontransparent text mode
- Blocked data records in either transparent or nontransparent text mode
- Blocked data records with record separators in nontransparent text mode
- Blocked data records with intermediate text block (ITB) characters in nontransparent text mode
- Blocked data records with blank compression in nontransparent text mode

- Blocked data records with blank truncation in either transparent or nontransparent text mode
- 3740 multiple files (with null records separating the files)
- Office systems, with end-of-text (ETX) control characters separating the documents

Use the same data format for both the sending and receiving systems. (Data formats are described in detail in “Syntax for Program Start Requests” on page 5-1.)

BSCCL also supplies EBCDIC and ASCII translation. (You must specify EBCDIC or ASCII as the value of the CODE parameter on the Create Line Description for BSC (CRTLINBSC) command.) The AS/400 system processes all data in EBCDIC characters.

If the remote system uses ASCII characters, BSCCL translates the following characters:

- Output data to ASCII, before sending it to the remote system
- Input data from the remote system to EBCDIC, before passing it to the AS/400 communications program

Chapter 2. Configuring BSCCL

This chapter describes the commands used for configuring BSCCL.

When using BSCCL 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*.

The following is a brief introduction of the commands you use to configure BSCCL. For a complete description of these and related commands, see the *Communications User's Guide* and the online help.

Defining the BSCCL Configuration

The AS/400 system with BSCCL support permits creating and storing many BSCCL configuration descriptions on the system. Each configuration description name must be unique. On a point-to-point line, only one BSCCL configuration can be active at a time. On a multipoint line, multiple BSCCL configurations can be active.

A BSCCL configuration consists of a BSC line, controller, and device description. You create these descriptions using the following commands in the order listed:

- Create Line Description for BSC (CRTLINBSC)
- Create Controller Description for BSC (CRTCTLBSC)
- Create Device Description for BSC (CRTDEVBSC)

To change one or more of the attributes of a BSCCL configuration, use the following commands:

- Change Line Description BSC (CHGLINBSC)
- Change Controller Description BSC (CHGCTLBSC)
- Change Device Description BSC (CHGDEVBSC)

Certain parameters for the configuration commands can only be changed when the configuration description is varied off. Refer to the *Communications User's Guide* to determine if you must vary off the configuration description to make changes. If the configuration description is varied off to make changes, you must vary on the configuration description after the changes are made. This permits using the new attributes for your session.

To display the status of the line, controller, and device descriptions for BSC, use the Work Configuration Status (WRKCFGSTS) command. This command allows you to determine if the BSC line is available for use. The Retrieve Configuration Status (RTVCFGSTS) command can also be used.

To delete a BSC configuration description, use the following commands:

- Delete Line Description (DLTLIND)
- Delete Controller Description (DLTCTLD)
- Delete Device Description (DLTDEVD)

Chapter 3. Running BSCEL

This chapter contains the information you need to run BSCEL.

Vary On and Vary Off Support

Once BSCEL is configured, you can use the Vary Configuration (VRYCFG) command to activate and deactivate the line, controller, and device descriptions used by BSCEL.

The VRYCFG command prepares the local AS/400 system to communicate with the remote system. The remote system must also be prepared to communicate with the AS/400 system.

Use the VRYCFG command and specify STATUS(*ON) to vary on the configuration descriptions. On a multipoint line, the AS/400 system automatically responds when it is selected or polled after the line description, controller description, and device description are varied on.

Use the VRYCFG command and specify STATUS(*OFF) to vary off the configured descriptions. When you vary off a configuration, the association between BSCEL and the communications line is broken. There are no more user program communications on the line.

For additional information concerning the Vary Configuration command, refer to the *Communications User's Guide*.

The VRYCFG command has the following parameters:

CFGOBJ

Specifies the name of the description for the line, controller, or device to be varied on or off or a list of names of configuration elements of the same description type, such as line, controller, or device type.

CFGTYPE

Specifies the type of configuration description to be varied on or off.

***LIN:** The line is varied.

***CTL:** The controller is varied.

***DEV:** The device is varied.

STATUS

Specifies the status to which the configuration object is to be varied.

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

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

RANGE

Specifies what configuration elements should be varied, such as only the configuration element specified (*OBJ) or the configuration element specified and its attached configuration elements (*NET). For lines, the attached configuration elements are controllers and devices. For controllers, the configuration elements are devices. Devices are considered not to have attached configuration

elements. For devices there is no difference in specifying RANGE(*OBJ) or RANGE(*NET).

***NET:** All downline attached configuration elements are varied.

***OBJ:** Only the specified objects are varied.

RESET

Specifies if a reset is to be done for the input/output processor (IOP) associated with the object.

***NO:** The associated IOP is not reset.

***YES:** The associated IOP is reset.

Chapter 4. Writing Application Programs

This chapter describes the BSCCEL parameters used to define an ICF file, as well as how to use communications operations and functions to:

- Start and end a communications session
- Start and end a communications transaction
- Send and receive data
- Notify your program of errors
- Use response indicators

This chapter also discusses the I/O feedback area (which contains the results of read and write operations) and return code processing.

Using an Intersystem Communications Function File

An intersystem communications function (ICF) file must be created before your application can use BSCCEL. 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. Appendix A provides a list of the DDS keywords and system-supplied formats supported by BSCCEL.

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 valid for BSCCEL, and are described in detail in the *Communications Programmer's Guide*.

CRTICFF	The Create ICF File command allows you to create an ICF file.
CHGICFF	The Change ICF File command allows you to change the file attributes of the ICF file.
OVRICFF	The Override ICF File command allows you to temporarily change the file attributes of the ICF file at run time. These changes are only in effect for the duration of the job and do not affect other users of the file.
DLTF	The Delete File command allows you to delete a file from the system.
DSPFD	The Display File Description command displays the file description of any file on the system. This information may be printed or displayed.
DSPFFD	The Display Field Description command displays the description of the fields in any file on the system. This information may be printed or displayed.

ADDICFDEVE	The Add ICF Device Entry command allows you to permanently add a program device entry that contains a program device name, remote location information, and session level attributes.
CHGICFDEVE	The Change ICF Device Entry command allows you to permanently change the program device attributes previously added with the ADDICFDEVE command.
OVRICFDEVE	The Override ICF Device Entry command allows you to: <ul style="list-style-type: none"> • 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	The Remove ICF Device Entry command allows you to permanently remove the program device entry previously added 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 BSCEL. The default values are underlined.

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:** BSCEL uses the library list to locate the ICF file.

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

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

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

PGMDEV

Specifies the program device name being defined in the ICF file. The total number of program devices that can be added (and active) to an ICF file is determined by the MAXPGMDEV parameter on the CRTICFF or CHGICFF command. Specify a program device as follows:

program-device-name: A 1- to 10-character value that specifies 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. This program device name must be unique throughout the entries for the ICF file. BSCEL allows only one program device name for each remote location to be active in the file at one time.

RMTLOCNAME

Specifies the remote location name with which your program communicates.

***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 remote 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. You can specify *REQUESTER only once in the file. *REQUESTER is valid only for a target communications job. If you specify *REQUESTER in any other type of job, an escape message is sent when the program device is acquired. There is no default for this parameter.

remote-location-name: A 1- to 8-character value that specifies the name of the remote location with which your source program is communicating. This name is used by the AS/400 system and matches the RMTLOCNAME parameter in the Create Device Description for BSC (CRTDEVBSC) command. The remote location does not need to exist at the time these commands run, but must exist (the configuration description that contains this RMTLOCNAME must be varied on) at the time the program acquires the program device. You may add a given remote location name many times using different program device names. When a program is running, only one program device name associated with each BSCSEL remote location may be acquired to the file at any one time.

FMTSLT

Specifies the type of record 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 there are any record identification (RECID) keywords 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 there are no RECID keywords in the file, an error message is sent and the acquire operation for the program device fails.

***RMTFMT**: BSCSEL does not support this value. If you specify this value, the acquire operation for the program device fails.

CMNTYPE

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

***BSCSEL**: The prompt for all BSCSEL-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).

BLOCK

Specifies whether the system or the user blocks and deblocks transmitted records. With this parameter, you can specify one of the following conditions for record formatting:

- No blocking/deblocking: The record format described in DDS is the format for both the record and the block.
- User blocking/deblocking: You must provide the BSC controls needed to describe the record format to the system.

- System blocking with record separator characters: You specify the record separator character used by the system to determine record boundaries within the block.
- System blocking of fixed-length records: The system uses fixed-length records, and blocks and deblocks records accordingly.

Specify these conditions as follows:

***DEV D:** BSCEL uses the block option specified in the device description. The block information in the device description is specified in the BLOCK parameter and the SEPCHAR parameter of the CRTDEV BSC command and the CHGDEV BSC command.

***NONE:** Blocking and deblocking are not done by the system.

***ITB:** Records are blocked or deblocked based on the location of an intermediate-text-block (ITB) control character. For input files, a record is delimited by locating the next ITB character. An end-of-text (ETX) or end-of-transmission-block (ETB) character is used also as an ITB character to delimit a block. For output files, an ITB character is added after the record. If it is the last character of the block, the ITB is replaced by an ETX or ETB character.

***IRS:** Records are blocked or deblocked based on the location of an interrecord-separator (IRS) character, hex 1E. For input files, a record is delimited by locating the next IRS character. For output files, an IRS character is added after the record.

***NOSEP:** A record separator character is not contained in the block that is sent to or received from the device. The system blocks and deblocks the records using a fixed-length record, as specified in the DDS format specifications.

***USER:** Your program provides all control characters (including record separator characters, BSC framing characters, and transparency characters) necessary to send records.

When sending records, BSCEL scans the buffer for the last nonblank byte to determine the length of the data to be sent. For this reason, you must make sure that the unused portion of the buffer contains blanks, or an error occurs.

When receiving records, you must be aware that the received text ends with an ETB or an ETX control character. BSCEL pads the remaining buffer space with blanks. This method of blocking allows you to send and receive variable-length data blocks by using a single record format capable of accommodating the maximum block length. Except for padding and truncating with blanks, BSCEL passes the data to and from the system when user blocking is specified. Before selecting this option, you should understand BSC methodology. You cannot specify BLOCK(*USER) when you specify RMTBSCEL(*YES).

***SEP:** Records are blocked or deblocked based on the location of a user-specified record separator character. For input files, a record is delimited by locating the next record separator character. For output files, a record separator character is added after the record.

record-separator-character: A value that specifies a unique, 1-byte record separator character. This value corresponds to the SEPCHAR parameter on the CRTDEV BSC command and the CHGDEV BSC command. The record separator character can be specified as 2 hexadecimal characters, as in BLOCK(*SEP FD), or the character can be specified as a single character, as in BLOCK(*SEP @). If a record separator character is not specified, BSCEL uses the record separator character hex 1E.

Do not use the following BSC control characters as record separator characters:

Table 4-1. BSC Control Characters That Cannot Be Used as Record Separator Characters

EBCDIC (Hex)	ASCII (Hex)	BSC Control Characters
01	01	SOH (start-of-header)
02	02	STX (start-of-text)
03	03	ETX (end-of-text)
10	10	DLE (data-link escape)
1D	1D	IGS (interchange group separator)
1F	1F	ITB (intermediate group separator)
26	17	ETB (end-of-transmission block)
2D	05	ENQ (enquiry)
32	16	SYN (synchronization)
37	04	EOT (end-of-transmission)
3D	15	NAK (negative acknowledgement)

Note: Do not use hex 00 as a record separator character for a session in which a source program starts a target program. (An evoke function or a program start request can be used to start a target program.) If you use hex 00 as a record separator character, BSC uses *NOSEP as the value of the BLOCK parameter. The value *NOSEP indicates that fixed-length record blocking is used.

If you specify any BLOCK parameter value other than *NONE or *USER, records are blocked as required by the system for output and are deblocked on input.

Blocking can be done with or without record separator characters. If you specify TRANSPY(*YES), the records can be blocked without record separator characters by specifying BLOCK(*NOSEP), or the records can be transmitted one record at a time by specifying BLOCK(*NONE). By specifying BLOCK(*USER), you can block records to include the BSC transparency controls.

If you specify TRANSPY(*NO), all blocking options are valid. The record length, when used, is obtained from the ICF file. A maximum of 512 records are blocked for transmitting. When the system blocks and deblocks the records, record separator characters and control characters are not passed to your program as data.

RCDLEN

Specifies the maximum record length (in bytes) for data sent and received. If a record is longer than the specified record length, an error occurs when you send or receive the record.

***DEV D:** BSC uses the record length from the device description.

record-length: A value that specifies the length of the longest record you expect to send or receive in this session. Valid values are 1 to 8192. For additional information about determining the record length of data, refer to "Determining Record Lengths" on page 4-9.

BLKLEN

Specifies the maximum block length (in bytes) for data sent and received. The block length must be at least as long as the record length (RCDLEN) and cannot be greater than the maximum buffer size (MAXBUFFER) specified with the line description (CRTLINBSC command). BSC checks to ensure that the maximum block length meets these requirements. If it does not, BSC fails the

acquire operation for the current session if RMTBSCEL(*YES) is specified. BSCEL fails the first input or output operation if RMTBSCEL(*NO) is specified.

***DEV D:** BSCEL uses the block length from the device description (CRTDEV BSC command).

block-length: A value that specifies the length of the largest block of data records you expect to send and receive in this session. Valid values are 1 to 8192. This value includes record separator characters when blocking is used.

TRNSPY

Specifies if data is sent in transparent text mode. Transparent text mode allows you to send all 256 extended binary-coded decimal interchange code (EBCDIC) character codes. Use this function when sending packed or binary data fields or data fields that contain characters that duplicate BSC control characters.

***DEV D:** BSCEL uses the text transparency option specified in the device description (CRTDEV BSC command).

***NO:** Text transparency is not used.

***YES:** Text transparency is used.

You cannot specify TRNSPY(*YES) when you specify CODE(*ASCII) on the Create Line Description for BSC (CRTLINBSC) command. If you do, BSCEL issues an error message when you attempt the acquire operation.

You can only specify TRNSPY(*YES) when you specify BLOCK(*NONE), BLOCK(*NOSEP), or BLOCK(*USER). This parameter is not relevant for received data since the data stream determines the transparency of received data.

If you specify TRNSPY(*YES) with BLOCK(*USER), BSCEL ignores the transparency indicator during write operations. You must give the correct controls in the data to send transparent data. For example, you must first specify the data-link-escape (DLE) and start-of-text (STX) control characters. The system provides the remaining control characters for transparent data transmission.

DTACPR

Specifies if blanks in the data are compressed for output and decompressed for input.

***DEV D:** BSCEL uses the data compression option specified in the device description (CRTDEV BSC command).

***NO:** Blanks in the data are not compressed or decompressed.

***YES:** Blanks in the data are compressed and decompressed. Data must not contain the intergroup separator (IGS) character (value hex 1D).

If you specify DTACPR(*YES) when you specify BLOCK(*USER), BSCEL ignores the data compression indicator. You cannot specify DTACPR(*YES) when you specify TRNSPY(*YES) or BLOCK(*ITB). If you do, BSCEL issues an error message when you attempt the acquire operation.

TRUNC

Specifies if trailing blanks are truncated from output records.

***DEV D:** BSCEL uses the truncation option specified in the device description (CRTDEV BSC command).

***NO:** Trailing blanks are not truncated from output records.

***YES:** Trailing blanks are truncated from output records.

If you specify TRUNC(*YES) when you specify BLOCK(*USER) or DTACPR(*YES), BSCCEL ignores the truncation indicator. You cannot specify TRUNC(*YES) with BLOCK(*NOSEP) or BLOCK(*ITB). If you do, BSCCEL issues an error message when you attempt an acquire operation.

GRPSEP

Specifies a separator for groups of data (the separator follows the last data record in the group).

***DEV**: BSCCEL uses the group separator option specified in the device description (CRTDEV BSC command).

***EOT**: An end-of-transmission (EOT) control character follows the last data record.

***DEV3740**: A null record (STXETX) follows the last data record.

***OFCSYS**: An end-of-text (ETX) control character follows the last data record.

RMTBSCCEL

Specifies the type of BSCCEL session with the remote system.

***DEV**: BSCCEL uses the remote session value specified in the device description (CRTDEV BSC command).

***NO**: The remote system cannot recognize BSCCEL commands or messages. For example, the remote system is a 3741 Data Entry Station, an Office System, a 5230 Data Collection System, or a System/38.

***YES**: The remote system can recognize the BSCCEL transaction starting commands, transaction ending commands, and online messages. For example, the remote system is another AS/400 system, a System/36, or a System/34 with BSCCEL support.

INLCNN

Specifies how the connection is made on the communications line for the session being started. This parameter applies only to switched communications lines.

***CTLD**: BSCCEL uses the switch type specified in the controller description (CRTCTL BSC command).

***DIAL**: The local system starts the call (the remote system answers).

***ANS**: The local system answers the call (the remote system calls).

SECURE

Specifies if this program device is protected from the effects of override commands at lower call levels. (This parameter is specified only on the OVRICFDEVE command.)

***NO**: This program device override is not protected from other program device overrides. Its values can be overridden by any program device override commands at lower call levels.

***YES**: This program device override is protected from other program device overrides. Its values cannot be overridden by any program device override commands at lower call levels.

Comparing the Program Device Entry Command and Configuration Parameters

The parameter values from the configuration commands are used for any BSCCL session in any ICF file, unless those values are changed by the program device entry commands.

Table 4-2 shows the relationship between the BSCCL parameters for the program device entry commands (ADDICFDEVE, CHGICFDEVE, and OVRICFDEVE) and the configuration commands. If there is no configuration parameter that corresponds to a program device entry parameter, it is marked with a dash (–). Except where noted, you specify all configuration parameters when you create the device description (CRTDEVBSCL command).

Table 4-2. Comparing Configuration Command Parameters and Program Device Entry Command Parameters

Description	Configuration Command Parameter	Program Device Entry Command Parameter
File and library name	–	FILE ⁴
Program device name	–	PGMDEV
Remote location name	RMTLOCNAME	RMTLOCNAME ¹
Format selection processing for input	–	FMTSLT
Communications type	–	CMNTYPE
Block information	BLOCK	BLOCK
Record separator character	SEPCHAR	BLOCK
Maximum user record length	RCDLEN	RCDLEN
Maximum block length	BLKLEN	BLKLEN
Transparency	TRNSPY	TRNSPY
Data compression indicator	DTACPR	DTACPR
Trailing blanks truncated	TRUNC	TRUNC
Separator for groups of data	GRPSEP ⁵	GRPSEP ⁵
Remote session indicator	RMTBSCEL	RMTBSCEL
Switch type for session start	INLCNN ²	INLCNN
Secure from override indicator	–	SECURE ³

¹ If you specify RMTLOCNAME(*REQUESTER) on the command, BSCCL ignores any of the data format parameters you specify, as well as the INLCNN parameter. (Data format parameters are BLOCK, SEPCHAR, RCDLEN, BLKLEN, TRNSPY, DTACPR, TRUNC, GRPSEP, and RMTBSCEL.) Instead, the remote program start request determines the data format parameters for this session.

If you intend to use the default device description value (*DEVDD) for other parameters, you must specify the same remote location name for the configuration parameter and the program device entry parameter.

² You specify this parameter when you create the controller description (CRTCTLBSCL command).

³ This parameter is only valid on the OVRICFDEVE command.

⁴ This parameter is only valid on the ADDICFDEVE and CHGICFDEVE commands.

⁵ The value of the GRPSEP parameter cannot be specified on a remote program start request. The local and remote systems are each responsible for setting this value correctly before the program start request is sent by the source program.

The ADDICFDEVE and CHGICFDEVE program device entry commands cause permanent changes for any BSCCL session that uses the specified program device. The OVRICFDEVE program device entry command causes job-level changes (as long as the OVRICFDEVE command remains in effect) for any BSCCL session that uses the specified program device.

For additional information about the relationship of these commands, refer to the *Communications Programmer's Guide*.

Determining Record Lengths

The record length for an output operation is determined by the record format specified.

- If you are using user-defined formats, the record length is determined by the record definition in DDS. You can use the VARLEN keyword to change the length of the data being sent. If you use VARLEN, the value specified for the variable length cannot be greater than the length of the record definition.
- If you are using system-supplied formats, you specify the length in the first four bytes of the data.

The record length for an input operation is determined by the record format specified.

- If you are using user-defined formats, either you specify the record format in your program, or it is selected by the system, based on the input data. You must ensure that your ICF file has a default format with a length equal to the longest record you expect your program to receive.
- If you are using system-supplied formats and the system-supplied QICDMF file, the input length is always 4096, unless you override this value using the OVRICFF command.

After the record length is determined, it is checked for errors as follows:

- The record length must not be greater than the maximum record length (MAXRCDLEN) value specified on the CRTICFF or OVRICFF command.
- The record length must not be greater than the application program's maximum record length.
- The record length must not be greater than the maximum user record length (RCDLEN) for the communications session. You specify the value for the RCDLEN parameter on the CRTDEVBS, ADDICFDEVE, CHGDEVBS, CHGICFDEVE, and OVRICFDEVE commands.
- The record length must not be greater than the maximum block length (BLKLEN), if record blocking is used for the communications session. You specify the value for the BLKLEN parameter on the CRTDEVBS, CHGDEVBS, ADDICFDEVE, CHGICFDEVE, and OVRICFDEVE commands.
- The record length must not be greater than the maximum buffer size (MAXBUFFER) for the communications session. You specify the value for the MAXBUFFER parameter on the CRTLINBS and CHGLINBS commands.

Following are specific return codes your program can receive for record length errors:

- Return code 3401 on an input operation, when the length of data received is greater than the record length for the input operation.
- Return code 8187 on an input or output operation, when the maximum user record length (RCDLEN), or the maximum block length (BLKLEN) is greater than the maximum buffer size (MAXBUFFER) and RMTBSCEL(*NO) is specified. See "RMTBSCEL Considerations" on page 4-12 for information on how to specify RMTBSCEL(*NO).
- Return code 8287 on an acquire operation when the maximum user record length (RCDLEN), or the maximum block length (BLKLEN) is greater than the

maximum buffer size (MAXBUFFER), and RMTBSCSEL(*YES) is specified. See “RMTBSCSEL Considerations” on page 4-12 for information on how to specify RMTBSCSEL(*YES).

- Return code 828B on an acquire operation when the maximum user record length (RCDLEN) is greater than the maximum block length (BLKLEN).
- Return code 831F on an output operation, when the record length is greater than the maximum record length (MAXRCDLEN), or the record length is greater than the application program’s maximum record length, or the record length is greater than the maximum user record length (RCDLEN).

If your program cannot specify a record length that is large enough to receive the data, you must end your program and change either the record length associated with the input operation, or the value of the RCDLEN parameter.

Starting a Session

A communications session is a logical connection between two systems through which a local program can communicate with a program at a remote location. A communications session is established with an acquire operation and is ended with a release operation or end-of-session function.

Open/Acquire Operation

Your program must open an ICF file and acquire a program device before it can direct any input or output operations to the program device. Only program devices defined to the file by the ADDICFDEVE or OVRICFDEVE command can be acquired.

You can acquire a program device implicitly using the open operation or explicitly using the acquire operation. The acquire operation is performed automatically as part of the open operation if you specify the ACQPGMDEV parameter on the ICF file. For a description of the open operation, see the *Communications Programmer’s Guide*.

Source Program: In a source program, after opening the ICF file, use an acquire operation to start the session. The program device name on the acquire operation identifies the session and must match the program device name specified in an associated ADDICFDEVE or OVRICFDEVE command. Any values specified for the data format parameters on the ADDICFDEVE or OVRICFDEVE command are used for the session.

For a point-to-point line, the acquire operation reserves the local AS/400 system line. Once your program starts the session, no other local AS/400 program can start a session on that line until your session ends. If you specify RMTBSCSEL(*YES) for your session and the remote system is also using BSCSEL, the communications line is reserved at the remote system. While this line is reserved, no other remote program can start a session on this line until your session ends.

If you specify RMTBSCSEL(*YES) for your session and you are using a switched line, the switched connection is made when the acquire operation is processed. If you specify RMTBSCSEL(*NO) for your session using a switched line, the connection is made when the first input or output operation is processed.

For a multipoint line, your program or other local AS/400 programs can start more than one session on the local AS/400 line. However, each session must be directed to a different device description.

Target Program: A target program (and the session) on the AS/400 system is started when the AS/400 system receives a program start request from the remote system. Before your target program can send or receive data, it must first make a logical connection to the source program. This logical connection is made when your target program uses an acquire operation.

The program device name on the acquire operation identifies the session. This name must match the program device name specified in an associated ADDICFDEVE or OVRICFDEVE command. You must specify a requesting device for the remote location (RMTLOCNAME(*REQUESTER)) on the ADDICFDEVE or OVRICFDEVE command for the target program that is started by a program start request.

Any values specified for the ADDICFDEVE or OVRICFDEVE command parameters that are not required parameters are ignored, except for the format select (FMTSLT) parameter. If you specify a remote format with FMTSLT(*RMTFMT), the acquire operation fails. The local AS/400 system BSCCEL configuration parameters are used for the session unless parameters are sent on the program start request to override them.

Program start requests are discussed further in “Program Start Request Considerations” on page 5-1.

Starting a Transaction

A transaction is a logical connection between two programs. Use the evoke function to start a transaction between your program and a program on a remote system. If you start a program on a remote system, you must specify whether the remote system recognizes BSCCEL commands. If security is required on the remote system, use the security function to pass security information with the evoke function. The following sections discuss these considerations.

Evoke Function

Use the evoke function to start a transaction after you start a session. If you specify RMTBSCCEL(*YES) for the session started by your source program, the evoke function attempts to start a program on the remote system.

Your program can use an evoke function only after it starts a session. You can use more than one evoke function in a BSCCEL session. However, only one transaction at a time is active. The previous transaction must end before you use the next evoke function.

If a remote program start request started your program, do not use an evoke function after acquiring a program device that has a remote location specified as a requesting device (RMTLOCNAME(*REQUESTER)). The session and transaction have already been started with the program on the remote system.

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

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

BSCCEL does not support the synchronization level keyword with the evoke function.

For information on how to code the evoke function, refer to the *Communications Programmer's Guide* and the *DDS Reference*.

RMTBSCSEL Considerations

The type of transaction started by an evoke function is determined by the type of remote session you specify on the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE program device entry commands or on the CRTDEVBSC or CHGDEVBSC configuration commands. Specify RMTBSCSEL(*NO) when your program communicates with other systems or devices (such as a System/38 or a 3741 Data Entry Station) that cannot recognize BSCSEL commands and online messages. Specify RMTBSCSEL(*YES) when your program communicates with other systems that are also using BSCSEL.

Specifying RMTBSCSEL(*NO)

When you specify RMTBSCSEL(*NO) and your program has started a session, you can issue the evoke function to start a transaction, but it does not start a program on the remote system. BSCSEL assumes the program on the remote system is already active. Because the evoke function does not cause line transmission, the parameters specified with the evoke function are not sent and are ignored by BSCSEL. If an evoke function is not used, the first input or output operation starts the transaction.

The evoke function is optional when you specify RMTBSCSEL(*NO). If the evoke function is not used, the first input or output operation from your program starts the transaction and data transmission with the program at the remote system.

If you want to convert your program to use other communications types, use the evoke function to start a transaction when you specify RMTBSCSEL(*NO).

Specifying RMTBSCSEL(*YES)

When you specify RMTBSCSEL(*YES) and your program has started a session, the evoke function starts a transaction and also starts a program on the remote system with a program start request. The RMTBSCSEL(*YES) parameter causes BSCSEL to build and send the program start request. You can specify the following parameters with the evoke function:

- The name of the target program to be evoked. This is a required parameter with a maximum length of 8 bytes.
- The name of the library containing the target program. This is an optional parameter with a maximum length of 8 bytes.
- User-defined parameters for the target program. These are optional parameters with a maximum combined length of 118 bytes. The target program defines the number and format of the parameters.

Note: If you specify program initialization parameters (user-defined parameters) with the evoke function, each parameter that is sent should be equal in length to the corresponding parameter specified in the target program. If it is longer than the parameter length in the target program, truncation occurs. If it is shorter than the parameter length in the target program, results that are not predictable may occur.

- Security parameters. These are optional parameters.

See "Syntax for Program Start Requests" on page 5-1 for more information on the parameters you specify with the evoke function.

Security Function

When you specify RMTBSCEL(*YES) and use an evoke function, you may need to provide security information for the program start request that BSCCEL sends to the remote system. You must provide security information if there is security on the remote system.

Use the security function with the evoke function to pass the following security parameters:

- The password. This is an optional parameter with a maximum length of 4 bytes.
- The user identifier. This is an optional parameter with a maximum length of 8 bytes.

BSCCEL does not support the profile ID and ignores it if you specify it with the security information.

Sending Data

You can send data during a transaction using the write operation. With the write operation, you can specify the end of a group of records or subdevice selection. You can also change the record length using the variable-length-data function. (See “Determining Record Lengths” on page 4-9.)

Write Operation

The write operation issues a data record from your program to the remote system in this session. BSCCEL sends the record in the data format specified by the configuration parameters or by the ADDICFDEVE or OVRICFDEVE command. When you use a write operation, it issues one data record to BSCCEL and returns control to your program without waiting for the operation to be completed.

Your program can only use a write operation during a transaction. To use a write operation without sending any data, specify an output record length of zero.

If the last function your program used before a write operation was an invite function, and data is not available, the write operation causes an implicit cancel-invite. If data is available, your program receives return code 0412 (output exception occurred). Your program must issue an input operation to receive the data before it can issue an output operation. (See page B-8 for more information on return code 0412.)

Combining Write with Other Operations and Functions

You can combine the write operation with several communications operations and functions. For example, if you combine a write operation with a read operation or invite function, BSCCEL sends an end-of-transmission indication to the remote system. If you specify GRPSEP(*DEV3740) for this session, BSCCEL sends a null record after the last data record, and then sends the end-of-transmission indication. If you specify GRPSEP(*OFCSYS) for this session, BSCCEL sends an end-of-text (ETX) control character after the last data record, and then sends the end-of-transmission (EOT) indication. BSCCEL then requests input data from the remote system. With an invite function, control returns to your program without waiting for the remote system to send the data. After an invite, your program must use a read or read-from-invited-program-devices operation to obtain the data. (Use the timer function or the WAITRCD parameter on the CRTICFF command to limit the waiting time for the read-from-invited-program-devices operation.)

If you specify an output record length of zero when the communications line is in a contention state, no line transmission will occur if you combine the write operation with any of the following functions or operations:

- Allow-write function
- Invite function
- Read operation
- Detach function with RMTBSCEL(*NO)

Specifying Record Blocking

If you do not specify record blocking when you use a write operation, BSCEL sends one data record to the remote system for each write operation. If you do specify record blocking, BSCEL blocks data records before sending them to the remote system.

When you specify record blocking and issue consecutive write operations (without a read operation or invite function), your program is normally two or more write operations ahead of the data on the communications line. If a communications line fails while you are sending data, your program is notified by a return code on its current write operation. Since the current operation is two or more write operations ahead of the failing operation, your program cannot determine if all data was sent before the line failed.

Subdevice-Selection Function

Use the subdevice selection function to specify the remote system device (such as a printer, punch, or diskette) to which you are sending data. The receiving controller then directs output from your program to the appropriate device. The subdevice selection is designed primarily to support specific hardware devices, such as 3776, 3777, and 3780. You should only use the subdevice selection if your program communicates with one of these devices, and you specified RMTBSCEL(*NO) for the communications session.

BSCEL processes the subdevice selection only when the output operation is:

- The first I/O operation in a session that was started by a source program's acquire operation. (The source program receives an error message if the subdevice is specified with an evoke function.)
- The first output operation used after an input operation, for which an end-of-transmission indication was received.
- The first output operation used after an output operation that specified either the allow-write, detach, or end-of-group function.

BSCEL ignores the subdevice selection if it has been selected at any other time.

BSCCEL sends a device selection character in a separate record as follows:

Parameter Value	Character Sent
*DC1	Hex 11
*DC2	Hex 12
*DC3	Hex 13
*DC4	Hex 5D

End-of-Group Function

When you use the end-of-group function, BSCCEL indicates to the remote system that this is the last record of a user-defined group of records. This function also indicates to BSCCEL that your program is not requesting any input. BSCCEL returns control to your program after the remote system acknowledges that it has received the last data record.

When you specify the end-of-group function, you must also specify one of the following values for the group separator (GRPSEP) parameter during configuration or on the ADDICFDEVE or OVRICFDEVE command:

- *EOT. An end-of-transmission (EOT) control character follows the last data record.
- *DEV3740. A null record (STXETX) follows the last data record.
- *OFCSYS. An end-of-text (ETX) control character follows the last data record.

Variable-Length-Data Function

For information about determining the record length for an output operation, refer to “Determining Record Lengths” on page 4-9.

Receiving Data

You can use two operations to receive data: read and read-from-invited-program-devices. Use the read operation to receive data from a specific program device. Use the read-from-invited-program-devices operation to receive data from any previously invited program device. A record length is specified for each input operation. See “Determining Record Lengths” on page 4-9.

You can use several functions with these operations, including invite, timer, and record identification.

Read Operation

Your program uses the read operation to obtain a data record from a specific program device acquired to the ICF file. The read operation also causes the program to wait for the data if it is not immediately available. Your program receives control when the data is available.

Whether or not a record format is specified on the read operation, the value for the FMTSLT parameter on the program device entry command determines which record format is used to process the received data. If a record format is not specified on the read operation, the system may use the default record format in the file. This default record format should be at least as large as the maximum user record length

(RCDLEN) configured on the device description. See the *Communications Programmer's Guide* for more information on what the default record formats are for each value of FMTSLT.

Invite Function

Your program uses the invite function to request input data from a specific program device. Your program receives control after the invite request without waiting for the input data. To get the data, your program must use either a read or read-from-invited-program-devices operation later in this transaction.

The invite function and read-from-invited-program-devices operations are used together. After using an invite function, use the read-from-invited-program-devices operation to receive the data from the remote system.

When data is received from an invited program device, that device is no longer invited. You must use another invite function to make it eligible to respond to a read-from-invited-program-devices operation.

You do not need to use an invite function before a read operation to receive data. However, if you use a read operation, and the program device has an outstanding invite, the read completes the invite and receives the data when it becomes available.

Read-from-Invited-Program-Devices Operation

Your program uses the read-from-invited-program-devices operation to get data from any program device that has responded to an invite function previously used by your program. If data becomes available to your program from more than one program device before you use the read-from-invited-program-devices operation, your program receives the data that was first made available.

Record-Identification Function

The record-identification function identifies and selects the record format to use with a read operation, depending on what data is received from the remote system.

This function is only applicable if you specify FMTSLT(*RECID) on the ADDICFDEVE or OVRICFDEVE command.

Variable-Length-Data Function

For information about determining the record length for an input operation, refer to "Determining Record Lengths" on page 4-9.

Notifying the Remote Program of Problems

You can inform the application program that an error occurred in the data being sent or received using the fail function. Specify this function on the write operation.

Fail Function

Use the fail function to indicate an error when sending or receiving data. The function indicates that the AS/400 system is ending the current communications session with the remote system.

When sending data, the fail function causes BSCCEL to discard all records not sent in the current transaction. If some records have been sent, BSCCEL abnormally ends the session by sending the remote system a temporary text delay (TTD) followed by an EOT indicator.

When receiving data, the fail function causes BSCCEL to discard all incoming records. BSCCEL abnormally ends the session by sending an EOT control character to the remote system.

Your program should either end or use the end-of-session function after it uses the fail function. Either method will end the communications session. After the session has ended, you can attempt to start another communications session.

Using Additional Functions and Operations

You can use the following additional functions and operations with BSCCEL:

- Request-to-write function (to request to send data while your program is receiving data)
- Allow-write function (to complete sending data and to allow the remote system to send data)
- Cancel-invite function (to cancel an invite that your program issued)
- Get-attributes operation (to get the status of the current session)

Request-to-Write Function

Your program uses the request-to-write function to ask the remote system to stop sending so your program can send. Use the request-to-write function while your program is receiving data to tell the remote system you want to change the direction of data transmission. After issuing the request-to-write, your program must continue to receive data until the remote system sends an end-of-transmission indication. If the remote system allows the change, your program can send data, a message, or both to the remote system.

When you use the request-to-write function, BSCCEL sends a reverse-interrupt (RVI) indicator to the remote system as the response to the next data record received. BSCCEL sends only one RVI indicator to the remote system. If you issue additional request-to-write functions before the change of direction, BSCCEL ignores them.

When your program uses the request-to-write function and the remote system is also using BSCCEL, the program on the remote system receives return code 0010 (operation completed successfully – request to write) as the result of its next output operation, after remote BSCCEL receives the RVI indication. If the remote system is another AS/400 system, the program on that remote system can also use the receive-turnaround response indicator to determine if the end-of-transmission indication was received. For each output operation issued before and after that output operation, the remote program receives return code 0000 (operation completed successfully – continue) until it stops sending.

If the remote system sends an RVI indicator, your program receives return code 0010 at the end of a write operation. Your program should stop sending data and use a read operation as soon as possible. (For information on return codes 0000 and 0010, see page B-1 and page B-2.)

Allow-Write Function

Your program uses the allow-write function to explicitly inform the remote system or device that your system is done sending. Use the function while you are sending data to clear the buffers, forcing any data to be sent. (You can achieve the same effect by using the write operation followed by a read operation or invite function.)

The allow-write function forces BSCSEL to send an EOT control character to the remote system. After using an allow-write function, your application program can use either a read operation or an invite function followed by a read-from-invited-program-devices operation to receive data from the remote system. It can also use a write operation to begin sending data again.

Once the allow-write function forces the end-of-transmission, the line goes to contention state. In this state, either program can attempt to send or receive data. If it is a switched line, it may be disconnected if the time specified on the INACTTMR (Inactivity Timer) parameter on the Create Line Description for BSC (CRTLINBSC) command ends.

Cancel-Invite Function

Your program uses the cancel-invite function to cancel a valid invite function for which no data has yet been received from an invited program device. A successful cancel-invite function allows your program to begin sending data again.

If data or a message is received, or if BSCSEL must pass a return code to your program, the cancel-invite function is rejected and your program receives return code 0412 (output exception occurred). Your program must continue to receive data until it receives a return code that indicates a detach or an EOT was received. (See page B-8 for more information on return code 0412.)

Using Switched Lines: If you are using a switched line and specify RMTBSCSEL(*NO), and your transaction begins with an invite function, you cannot cancel that particular invite function. The cancel-invite function is rejected and return code 0412 is sent to your program. Your program must continue to receive data until it receives a detach or an EOT indication.

Timer Function

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

Use the timer function to set the timer interval. The timer function is issued on an output operation.

If data is available, your program receives the data and the successful return code. If an error occurs, your program receives a return code that describes the error.

Another way to specify the time interval is with the WAITRCD parameter on the CRTICFF, CHGICFF, and OVRICFF commands. The WAITRCD parameter estab-

lishes the maximum time interval used for all read-from-invited-program-devices operations issued for the ICF file.

When the timer function is in effect, the value specified for the WAITRCD parameter is ignored.

Get-Attributes Operation

Your program uses the get-attributes operation to determine the status of the current session. You can issue it at any time during the session. The operation gets the current status information about the session in which your program is communicating.

Ending a Transaction

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

Your program can end communications using the detach function. The detach function ends the transaction between the two systems, but the session between the AS/400 system and the remote system is still active. If your program started the session, it can start another transaction by using another evoke function. If the remote system started the session, it can start another transaction by sending another program start request to the AS/400 system.

Detach Function

Use the detach function to end the transaction. The detach function explicitly informs BSCCEL that your program is done sending or receiving and is ending the transaction.

If you specify RMTBSCCEL(*YES), or if your program was started by an *EXEC program start request, BSCCEL sends a detach indication to the remote system. (See "Formats for Program Start Requests" on page 5-4 for information on *EXEC.) Control returns to your program after the remote system acknowledges that it has received the detach indication. (See "Receive-Detach" on page 4-22 for information on how to handle detach indicators.)

If you specify RMTBSCCEL(*NO), or if your program was started by an *EXNC program start request, BSCCEL sends an EOT indication to the remote system if necessary. A detach indication is not sent.

The detach function is optional when you specify RMTBSCCEL(*NO). The following considerations apply to this case:

- If GRPSEP(*DEV3740) or GRPSEP(*OFCSYS) is used with the end-of-group function, and you are using an output only file, you must use the detach function to force the end of transmission.
- If the detach function is not used when you specify RMTBSCCEL(*NO), BSCCEL will send the EOT indication if either the allow-write function is used, or GRPSEP(*EOT) is used with the end-of-group function.
- If you want to convert your program to use another communications type and that type supports the detach function, use the detach function to end the transaction when you specify RMTBSCCEL(*NO).

Ending a Session

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

If your program started the session (source program), your program must end the session using either the release operation or the end-of-session function. You should primarily use the release operation. Use end-of-session only when you want to force the session to end. The release operation ends the session only if all processing is complete. The end-of-session operation *always* ends the session.

Release Operation

The release operation ends the session if all processing is complete. The processing done by the release operation is as follows:

- Source program
 - If the program device is invited, the release operation fails.
 - If a transaction is still active on the session, the release operation fails.
 - If a transaction is not active on the session, the session ends. If you specify RMTBSCEL(*YES) for this session, BSCCL sends a *REL command to the remote system. This informs BSCCL on the remote system that the session has ended so a program on the remote system can start a session. If the communications line is a switched line, BSCCL disconnects the line.

If the release operation fails, your program can use an end-of-session function to force the session to end.

- Target program
 - The release operation severs the logical connection between the application and the requesting program device. The session is not ended.
 - The program (or another program in the same job structure) can establish the connection again to the same session by acquiring the requesting program device. The communications session, including the state of the session, remains intact.

End-of-Session Function

The end-of-session function forces the session to end. The only possible return codes from end-of-session are 0000 or 830B (program device not acquired).

To prevent your program from ending abnormally because of a communications error, you may want to use the end-of-session function in your program as a general recovery action for all unexpected errors that are not permanent and that you have not handled individually in your program. For example, use the end-of-session function rather than trying the failing operation again in that session or specifying some special recovery action for each error. Use the end-of-session function in source and target programs as follows:

- Source program. Use the end-of-session function if you want to force the session to end. Generally, you should use the release operation to end the session.
- Target program. A target program must issue an end-of-session function or go to end of job in order to end the session.

Using Response Indicators

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

- Is the last record in a user-defined group
- Ends the transmission
- Ends the transaction

Response indicators are only effective for input operations, and you can use more than one response indicator on a single read operation. However, these indicators are optional, and major and minor return codes can also be used to indicate the status of input operations. Refer to Appendix B for descriptions of the return codes referred to in this section.

Receive-End-of-Group

You can use the receive-end-of-group response indicator to determine if the last record received in the input buffer was the end of a user-defined group of records. Data is not returned with this response indicator.

If you specify GRPSEP(*DEV3740) for this session, it indicates that a null record (STXETX or DLESTXETX) has been received. For example, this could indicate the end of a diskette file from a 5280 or 3741 Data Entry Station. Your program should continue to issue read operations until your program detects that an end-of-transmission (EOT) was received.

If you specify GRPSEP(*OFCSYS) for this session, it indicates that the previous record was the last record in a transmission block ending with end-of-text (ETX). For example, this could indicate the end of a document from a word processing device. Your program should continue to issue read operations until your program detects that an end-of-transmission (EOT) was received.

If you specify GRPSEP(*EOT) for this session, it indicates that an EOT or DLEEOT was received following a data block that ended with an ETX.

The presence of the end-of-group function is also indicated by the minor return code 08 with the major return codes 00 or 02, the minor return code 01 with the major return code 03 if GRPSEP(*DEV3740) or GRPSEP(*OFCSYS) is specified, or the minor return code 00 with the major return code 03 if GRPSEP(*EOT) is specified.

Receive-Turnaround

You can use the receive-turnaround response indicator to determine if the last record received in the input buffer ends a transmission. Data is not returned with this response indicator.

If the remote system sends an EOT or DLEEOT indicator following a data block that ends with an ETX, the receive-turnaround response indicator informs your program that the remote system is finished sending data. Your program can begin sending data, or can continue to receive data, depending on your application.

Normally, the receive-turnaround indicator is not set when an EOT or DLEEOT is received following a data block that ends with an end-of-text block (ETB). In this case, an exception response is given and data transmission ends abnormally.

However, if the EOT is sent as a result of an RVI, some devices do not end the last data block with an ETX because more data is still available to be sent. In this case, no exception response is given.

The presence of the turnaround indication is also indicated by the minor return codes 20 and 30 with major return codes 00 or 02, or the minor return code 00 with major return code 03.

Receive-Detach

You can use the receive-detach response indicator to determine if the last record received in the input buffer ends a transaction. Data is not returned with this response indicator.

The remote system informs your program that it is ending this communications transaction with your program when it sends a detach indication. This occurs only when you specify RMTBSCSEL(*YES) for the session, or when an *EXEC program start request started your program. Your program can no longer communicate with the program on the remote system, but the session with the remote system is still active. If your program started the transaction, it can use an evoke function to start another program, or it can end the session. If a remote program start request started the transaction, your program should end the session. Your program can then do noncommunicating functions before your program ends.

Note: This indicator has no meaning if you specify RMTBSCSEL(*NO) for this session, or if an *EXNC program start request started your program (see "Formats for Program Start Requests" on page 5-4 for information on *EXNC). The remote system does not send a detach indication.

The presence of the detach function is also indicated by the minor return codes 08, 28, and 38 with major return codes 00 or 02, or the minor return code 08 with the major return code 03.

Using the Input/Output Feedback Area

In addition to ICF messages, major/minor return codes, and high-level language status values, the I/O feedback area contains the results of read and write operations for your application program. For general information about the I/O feedback areas, see the *Communications Programmer's Guide*.

The *safe indicator* field in the file-dependent I/O feedback area applies only to BSCSEL operations. This field shows that an ETX control character has been received in the buffer. The safe indicator is not set if BLOCK(*USER) was specified for the session.

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 done
- The operation that should be done next

For example, a major return code of 00 indicates that data was received. Along with this major code you can receive from BSCSEL, for example, one of the following minor codes:

- 01: Indicates that your program should continue receiving data.
- 08: Indicates that the remote 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 a target program, end the session and continue local processing or go to end of job.

Another example would be a major code of 83. In this case either the local system, remote system, or remote program has detected an error that may be recoverable. Different minor codes can be returned just as with the 00 major code. For example, if your program receives an E8 return code, your program has used a cancel-invite function in a session that was not invited. The cancel-invite function is only valid when it is used after a valid invite function. 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 correcting the error in your program before trying to communicate with another program.

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 was taken.

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

Chapter 5. BSCCEL Considerations

This chapter describes program start requests and discusses considerations for using them with remote and local systems. It also discusses prestart jobs, BSCCEL commands for remote systems, and online messages for the local AS/400 system.

Program Start Request Considerations

A program start request is a request made by a source program to start a target program. After the target program is started, a communications transaction is started allowing data to be exchanged between the two programs.

If the remote system uses BSCCEL, the AS/400 system can send a program start request to the remote system. On the AS/400 system, the source program can send a program start request in one of the following ways:

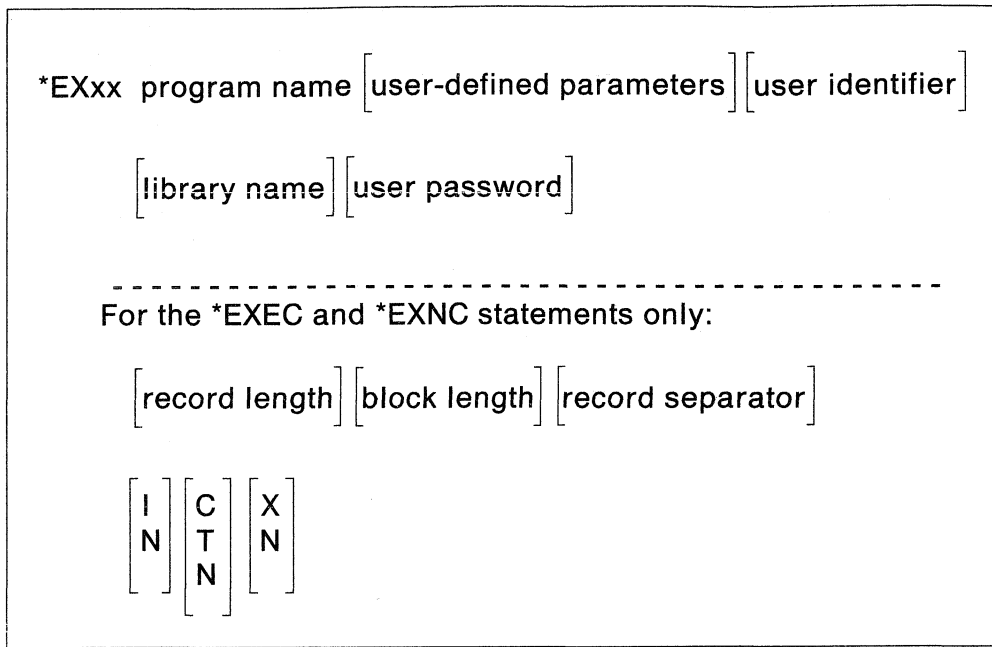
- If you specify RMTBSCCEL(*YES) for the communications session, BSCCEL automatically formats and sends the program start request when the source program uses an evoke function.
- If you specify RMTBSCCEL(*NO) for the communications session, you can send data in the proper format for a program start request with your program's first output operation.

A remote system can send a program start request to the AS/400 system.

The following sections describe the program start request format and syntax. Considerations for using program start requests from remote and local AS/400 systems are also discussed.

Syntax for Program Start Requests

Figure 5-1 on page 5-2 shows the syntax for BSCCEL program start requests.



RSL354-3

Figure 5-1. Syntax for Program Start Requests

Any user-defined parameters entered follow the program name (up to position 127) and are used by the target program. As many as 118 bytes of user-defined parameters can be passed to the target program. When an AS/400 system target program begins, it can access the user-defined parameters as if they were parameters passed on a Call Program (CALL) command.

The sender of the program start request uses the positional parameters (specified in positions 128 through 160) to pass any data-related specifications and security information, if necessary, to the system that receives the program start request.

A program start request statement cannot be greater than 160 characters. If the communications session uses record blocking, the program start request statement must be sent as the only record in the first block of data. BSCCEL processes the entire block as a program start request statement. At least one blank must separate the program name that begins in position 7 from the parameters. Position 127 must also contain a blank to separate the user-defined parameters (up to position 127) from the positional parameters (positions 128 through 160).

The record is only as long as the last field used in the program start request. Any positions not used should contain blanks.

Table 5-1 on page 5-3 lists the coding positions and describes the fields for a program start request.

Table 5-1. Coding Positions for a Program Start Request

Coding Positions	Field	Description
1 to 6	*EXxxb	Type of program start request used to start the program (xx = EC, EX, NC, or NX, and b = blank). Position 6 must be blank except when using the continuation format.
7 to xx	Program name	The name of the target program. The name must be 1 to 8 characters long and be followed by at least one blank.
xx to 127	User-defined parameters	Parameters you specify for the target program. This field begins with the first nonblank character that follows the program name. Position 127 must be a blank, if sent.
128 to 135	User identifier ¹	The user identifier of the user whose target program is being started. If security is active on the target program's system, this identifier must be defined on that system.
136 to 143	Library name ¹	The name of the library on the system that contains the target program to be started. If the target program is in the AS/400 system and no library is specified, *LIBL is assumed.
144 to 147	User password ¹	The 4-character password of the user whose target program is being started. If security is active on the target program's system, this password must be defined on that system and must be the correct password for the user identifier specified.
148 to 151	Record length ²	The maximum user record length passed between the communicating programs (4 decimal digits, right-adjusted). Do not specify a value greater than the maximum buffer size configured on the AS/400 system.
152 to 155	Block length ²	The length of the block of data records sent or received (4 decimal digits, right-adjusted). Specify 0000 for no record blocking. Do not specify a value greater than the maximum buffer size configured on the AS/400 system.
156 to 157	Record separator ²	The hexadecimal value of the character used as the separator between records. If you specify 00, a record separator is not used.
158	ITB choice ²	Indicates if ITB characters are used to separate records in a block. Specify I to use ITB characters, N to not use them.
159	Blank control ² (C, T, or N)	Indicates if blank compression, blank truncation, or neither is used for the records being sent or received. Specify C for blank compression, T for blank truncation, or N for neither.
160	Transparency choice ² (X or N)	Indicates if data transparency is used by the AS/400 system to send data. (Use transparency to send packed decimal or binary data. Transparency is automatically set for received data.) Specify X to use transparency, N to not use it.

¹ The user identifier, library name, and user password fields are positional and must be padded on the right with blanks if another field follows. If security is not used on the target program's system, the user identifier and password are not required; however, the coding positions must contain blanks.

² The data format parameters (positions 148 through 160) are used only if you specify *EXEC or *EXNC in positions 1 through 5. In this case, these values override the values specified in the target program's configuration. To use the configuration values, specify blanks in these fields.

Formats for Program Start Requests

The source program can use four types of program start requests. The different formats indicate whether the session is a communicating or noncommunicating session. They also indicate if BSCCEL commands and online messages are passed from BSCCEL on the local system to the remote system.

Note: All formats may not be applicable for all systems or devices. Use the information in this section and evaluate your device or system configuration to determine which formats can be used.

The formats consist of five characters. The first three characters (*EX) are the same for all formats. The fourth character indicates if BSCCEL commands and messages are passed (E) or not passed (N). The fifth character indicates whether the type is communicating (C) or noncommunicating (X).

The formats are:

- *EXEC** Commands and messages passed, communicating session. In this session, either program sends and receives data. The remote system uses BSCCEL to recognize BSCCEL commands and online messages.
- *EXEX** Commands and messages passed, noncommunicating session. In this session, the request statement is the only source of parameters for the target program. No further communications occur in this session between the source program and the target program on the remote system. The remote system uses BSCCEL to recognize BSCCEL commands and online messages.
- *EXNC** No commands or messages passed, communicating session. In this session, either program sends and receives data, but the remote system cannot recognize BSCCEL commands or online messages.
- *EXNX** No commands or messages passed, noncommunicating session. In this session, the request statement is the only source of parameters for the target program. No further communications occur in this session between the source program and the target program on the remote system. The remote system cannot recognize BSCCEL commands or online messages.

The format of the program start request received by the AS/400 system indicates whether RMTBSCCEL(*YES) or RMTBSCCEL(*NO) is used for the session. This format overrides the value specified for RMTBSCCEL in the BSCCEL configuration description or on the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command for this session. The formats *EXEC and *EXEX indicate that RMTBSCCEL(*YES) is used. The formats *EXNC and *EXNX indicate that RMTBSCCEL(*NO) is used.

If the source program uses BSCCEL to communicate and uses an evoke function to start a target program, BSCCEL automatically formats and sends one of the following program start requests:

- If the evoke function is sent with a detach function, BSCCEL sends *EXEX.
- If the evoke function is sent without a detach function, BSCCEL sends *EXEC.

The *EXNC and *EXNX statements are normally used by systems or devices (such as a System/38 or a 3741) that cannot process BSCCEL commands and messages, but can start a target program on a remote system. If either of these statements starts a program on the AS/400 system, BSCCEL does not send BSCCEL commands or online messages to the remote system.

Continuation Format for Program Start Requests

Some systems or devices (for example, the 3741) cannot send records longer than 128 bytes. BSCCEL allows you to break a program start request into two records and send it in the **continuation format**. Each of the two records should be only long enough to send the information required to start an AS/400 target program.

Table 5-2 shows the continuation format for a program start request.

Table 5-2. Continuation Format for Program Start Requests

Position	Field
First Record	
1 through 5	*EXNC or *EXNX
6	C
7 through xx	Program name
xx through 126	User-defined parameters
Second Record	
1 through 5	*EXNC or *EXNX
6	C
7 through 14	User identifier
15 through 22	Library name
23 through 26	User password
27 through 30	Record length
31 through 34	Block length
35 and 36	Record separator
37	ITB choice
38	Blank control
39	Transparency choice

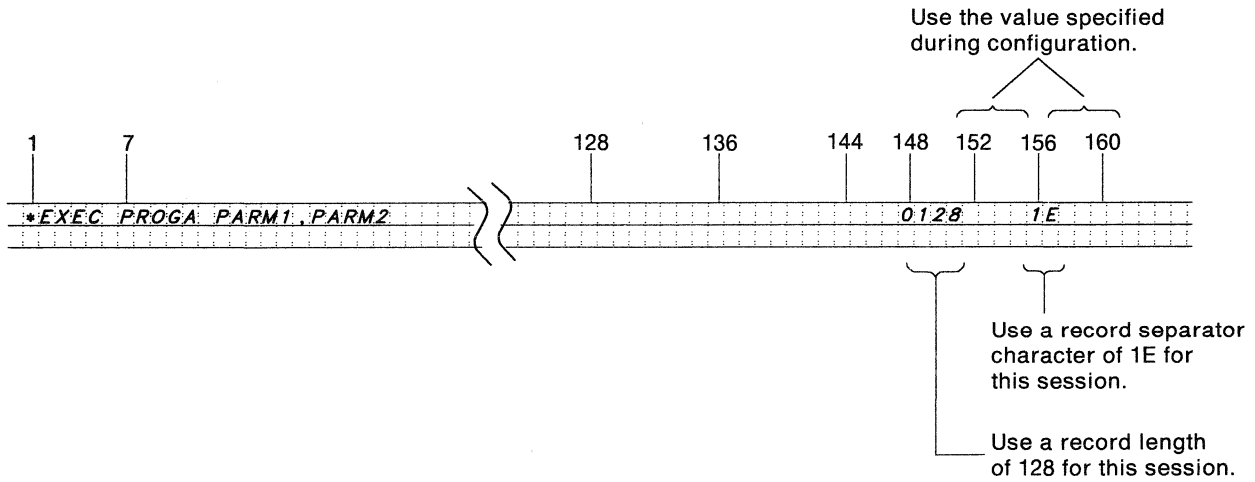
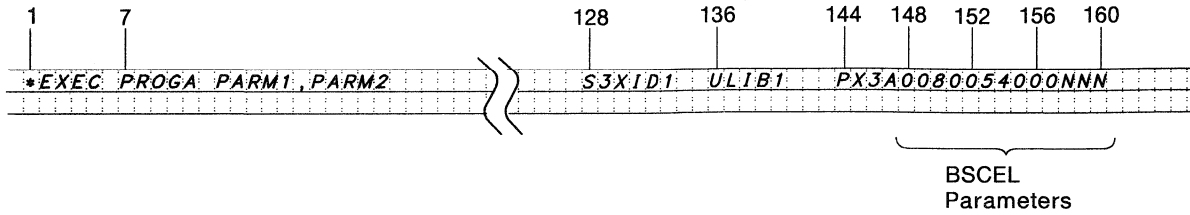
One or more blanks must follow the program name in the first record to separate it from the user-defined parameters. Both records must use the same format (positions 1 through 5) and must specify continuation (C in position 6). BSCCEL ignores all positions after position 126 of the first record and after position 39 of the second record.

Examples of Program Start Requests

Following are examples of the record format used to send program start requests.

Note: These examples do not show all programming considerations or techniques.

Figure 5-2 on page 5-6 shows the record format for *EXEC in the standard format of 160 bytes.



RSL3352-1

Figure 5-2. *EXEC Format for Program Start Request

Figure 5-3 shows the same information as in Figure 5-2, but uses the continuation format (with two records) with the *EXNC statement.

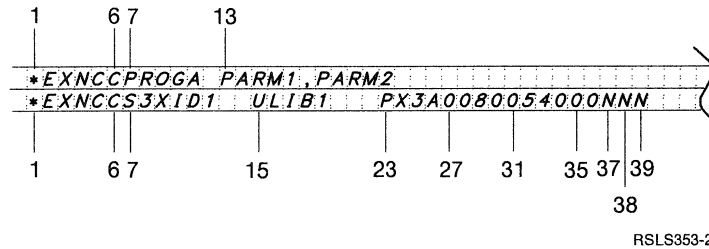


Figure 5-3. Continuation Format for Program Start Request

Program Start Requests Received from Remote Systems

A remote system must send a program start request to the AS/400 system to start a target program on the AS/400 system. When BSCCEL receives a program start request from a remote system, it attempts to start the specified AS/400 program. BSCCEL passes any user-defined parameters included with the request as if they were passed with a Call Program (CALL) command.

Table 5-3 on page 5-7 describes the session activity for an AS/400 target program started by a program start request.

Table 5-3. Session Activity

Activity	Result
The remote system sends a program start request when a transaction has already been started with your target program.	The session ends abnormally and BSCEL passes return code 819D to your target program. Your target program cannot continue to communicate in this session.
Your target program finished sending and receiving data, and the session ended normally.	BSCEL ends the session normally.
Your target program tries to send or receive data after the transaction ends.	BSCEL rejects the operation and passes return code 8327 to your target program. Your target program cannot continue to communicate in this session.

Program Start Requests Sent by the AS/400 System

Your source program on the AS/400 system can send a program start request to a remote system that uses BSCEL. To send a program start request, specify RMTBSCEL(*YES) on the BSCEL configuration or on the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command. When you use an evoke function, BSCEL builds and sends a program start request to the remote system.

The program start request is 160 bytes long and uses the same format as those created and sent by the remote system when it starts a program on your AS/400 system. It contains the following parameters you specify with the evoke function:

- The name of the target program to be evoked. This is a required parameter with a maximum length of 8 bytes.
- The name of the library containing the program. This is an optional parameter with a maximum length of 8 bytes.
- User-defined parameters. These are optional parameters with a maximum combined length of 118 bytes.
- Security parameters. These are optional parameters and are discussed in "Security Function" on page 4-13.

The total length of the target program name and user-defined parameters cannot be greater than 119 bytes. The target program defines the number and format of the user-defined parameters. If the remote system is another AS/400 system, these parameters are passed to the target program as if they were passed with a Call Program (CALL) command.

Note: If you specify program initialization parameters (user-defined parameters) with the evoke function, each parameter that is sent should be equal in length to the corresponding parameter specified in the target program. If it is longer than the parameter length in the target program, truncation occurs. If it is shorter than the parameter length in the target program, results that are not predictable may occur.

The program start request also contains parameters that define the data format used for this session, such as the maximum user record length and the block length. The data format parameters (positions 148 through 160) override the corresponding configuration parameters at the remote system. The target program uses the same data format as the source program.

The following actions occur when the remote system receives the program start request:

- The evoke parameters start the specified target program.
If the target program starts successfully, the transaction begins. If not, the source program receives a return code indicating that the evoke function failed.
- Any user-defined parameters sent with the evoke function are passed to the target program.
- The target program can start other programs in the remote system, but the target program cannot start another BSCEL transaction.

Failed Program Start Requests

Message CPF1269 is sent to the system operator message queue (QSYSOPR) when the AS/400 system rejects an incoming program start request. You can use the reason code in the message to determine why the program start request was rejected.

The CPF1269 message contains two reason codes. If one of the reason codes is zero, it can be ignored. If only one nonzero reason code is received, that reason code represents why the program start request was rejected.

If the System/36 environment is installed on your AS/400 system, there can be two nonzero reason codes. These two reason codes occur when the operating system cannot determine whether the program start request was to start a job in the System/36 environment or in the AS/400 environment. One reason code explains why the program start request was rejected in the System/36 environment and the other explains why the program start request was rejected in the AS/400 environment. Whenever you receive two reason codes, you should determine which environment the job was to run in and correct the problem for that environment.

Table 5-4 describes the reason codes for failed program start requests.

Table 5-4 (Page 1 of 3). 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 Entry (HLDCMNDEVE) 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.
501	Job description was not found.

Table 5-4 (Page 2 of 3). Reason Codes for Rejected Program Start Requests

Reason Code	Reason Description
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 was provided, but a password was sent.
722	A user ID was provided, but no password was sent.
723	No password was associated with the user ID.
725	User ID is not a valid name.
801	Program initialization parameters are present but not allowed.
802	Program initialization parameters exceed 2000 bytes for a prestart job.
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.
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.

Table 5-4 (Page 3 of 3). Reason Codes for Rejected Program Start Requests

Reason Code	Reason Description
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 to 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).
1901	The record or block length exceeds maximum buffer size.
1902	ASCII and transparency are mutually exclusive.
1903	Transparency and blank compression conflict.
1904	Block length is required with data format.
1905	Blank truncation and ITB conflict.
1906	Blank compression and ITB conflict.
1907	3740 multiple files and ITB conflict.
1908	Record separator and transparency conflict. ¹
1909	Record separator and ITB conflict.
1910	The record length exceeds the block length.
1911	Record separator character not valid.
1912	BLOCK(*USER) and RMTBSCSEL(*YES) conflict.
1913	BLOCK(*NOSEP) and blank truncation conflict. ²
1914	Program name not valid.
1915	Program start request record was too long.

¹ You can receive this reason code if transparency is specified as a program start request parameter value and BLOCK(*SEP) and SEPCHAR('00'X) are specified on the AS/400 system.

² You can receive this reason code if a record separator character '00'X is specified as a program start request parameter value and TRUNC(*YES) is specified on the AS/400 system.

Prestarting Jobs for Program Start Requests

To minimize the time required to carry out a program start request, you can use the prestart job entry to start a job on the AS/400 system before the remote program sends a program start request. 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 details about how to use prestart jobs, refer to the *Communications Programmer's Guide*.

For BSCSEL, you should note that the WAIT parameter on the Add Prestart Job Entry (ADDPJE) and the Change Prestart Job Entry (CHGPJE) commands specifies if a program start request waits for a prestart job to become available or is rejected if a

prestart job is not immediately available when the program start request is received. You should specify *NO as the value for this parameter to avoid errors that can occur because of BSC timeout considerations.

Remote System Considerations

Remote systems either use BSC to communicate with BSC on the AS/400 system, or they do not use BSC.

If the remote system uses BSC, consider the following:

- If the local AS/400 system starts the session with an acquire operation, specify RMTBSC(*YES) in the BSC configuration description, or on the ADDICFDEVE or OVRICFDEVE command.
- If the remote system starts the session with a program start request, the program start request is either an *EXEC or *EXEX request. The evoke function used by the program on the remote system automatically builds and sends the request statement.

If the remote system does not use BSC, consider the following:

- If the local AS/400 system starts the session with an acquire operation, specify RMTBSC(*NO) in the BSC configuration description or on the ADDICFDEVE or OVRICFDEVE command. If the AS/400 system is expected to receive first, the application program on the AS/400 must issue a read operation before the remote system begins sending data.
- If the remote system starts the session with a program start request, the program start request should be either an *EXNC or *EXNX request statement, unless additional programming is done in the remote system to allow it to function like BSC.

BSC Commands

BSC sends some commands to the remote system for certain operations or conditions. The commands are sent only if you specify RMTBSC(*YES) or if the session was started by an *EXEC program start request.

The program on the AS/400 system that uses BSC does not receive these commands when receiving input. BSC checks all incoming data for these commands. Communications programs that use BSC should not send data that has the same format as one of these commands.

Acquire (*ACQ) Command

BSC sends the Acquire (*ACQ) command to the remote system when an acquire operation is used in an AS/400 program. When BSC on the remote system receives the *ACQ command, the issuing system acquires the session. Neither local nor remote BSC accept any other acquire operations (except acquire operations for requesting devices by remote BSC) until the program that used the acquire operation uses a release operation.

The syntax of this command is the 4 characters *ACQ.

Release (*REL) Command

The Release (*REL) command is sent as the result of a release operation, except for a release of a requesting device by BSCCL at a remote system. A *REL command received by BSCCL indicates that the remote system ended the session.

The syntax of this command is the 4 characters *REL.

End-of-Transaction (*EOX) Command

The End-of-Transaction (*EOX) command is sent as the result of a write operation when the detach function is specified. Receipt of the *EOX command by BSCCL indicates that the remote system ended the transaction.

The syntax of the command varies as follows:

- If you do not include data with the write operation, the format is the 4 characters *EOX.
- If you do include data and you are using blocking, the following occurs:
 - The data record is placed in the block.
 - The block is sent.
 - The *EOX command is sent.
- If you do include data but you are not using blocking, the following occurs:
 - The data record is sent.
 - The *EOX command is sent.

Receiving Null Records

A null record contains only the BSC control characters STXETX. BSCCL sends null records when:

- Your program uses a write operation (with no other functions specified) and a record length of zero bytes.
- Your program uses a write operation with the end-of-group function and you specify GRPSEP(*EOT) either in the BSCCL device description or on the ADDICFDEVE or OVRICFDEVE command, and you specify a record length of zero bytes.
- Your program uses a write operation with the end-of-group function while it is sending multiple files, and you specify GRPSEP(*DEV3740) either in the BSCCL device description or on the ADDICFDEVE or OVRICFDEVE command.

When the AS/400 system receives a null record from the remote system, it passes the return code 0301 (no data – continue to receive) to your program. (Return code 0301 is discussed on page B-6.) Your program must use a read operation for each null record received.

Using 3740 Data Entry Systems

This section describes the considerations for using program start requests and multiple file formats with the 3740 Data Entry Systems.

Sending Program Start Requests

Use the continuation format to send a program start request from a 3741 work station. To start a program on the AS/400 system from a 3741 using the expanded communications buffer feature, the diskette used in the 3741 should contain the program start request files and data files in the following order:

- File 1** Contains the first record of the program start request being sent in the continuation format.
- File 2** Contains the second record of the program start request being sent in the continuation format.
- Files 3 to xx** If the 3741 is only sending data in this session, these files contain the data records to be sent to an AS/400 system.

If the 3741 is sending and receiving data in this session, the first group of files (beginning with File 3) should contain the data records to be sent to an AS/400 system. The last group of files should receive any data from an AS/400 system.

Sending and Receiving Multiple File Formats

To configure BSCEL to send and receive multiple files in the 3740 format, specify GRPSEP(*DEV3740) in the BSC device description or on the ADDICFDEVE or OVRICFDEVE command.

Each file sent in 3740 multiple file format ends with a null record (STXETX). The null record indicates the end of one file and the beginning of the next file. Consider the following:

- To indicate the end of a file, your program can use the write operation with the end-of-group function to cause BSCEL to generate the null record to be sent after the last data record in the file. Another file can then be sent.
- To indicate the end of a file and to begin receiving files from the remote system, your program can use the write operation (with or without the end-of-group function) followed by an input operation. (Do not specify the detach or evoke function with this write operation.)

BSCEL sends the last data record and a null record followed by an EOT control character. BSCEL then waits for input from the remote system.
- To indicate the end of the last file and to end the session, your program can use a write operation with the detach function.

When multiple 3740 files are being received, return code 0301 (no data – continue to receive) indicates the end of each 3740 data file. Return code 0301 is returned to your program each time a null record is received. Return code 0300 (no data – EOT received) indicates that all files have been received. (For more information on these return codes, see page B-6.)

Using Office Systems

To configure BSCEL to send and receive office documents from systems and devices, such as the 6580 Displaywriter system, specify GRPSEP(*OFCSYS) for the BSC device description, on the ADDICFDEVE command, or on the OVRICFDEVE command.

Records sent in office-systems format end with an end-of-text-block (ETB) control character. To indicate the end of a document, the last record sent for each docu-

ment ends with an end-of-text (ETX) control character instead of an ETB character. Consider the following:

- To indicate the end of a document, your program can use the write operation with the end-of-group function to cause BSCCEL to send the ETX character with the last record for the document. Another document can then be sent.
- To indicate the end of a document and to begin receiving documents from the remote system, your program can use a write operation (with or without the end-of-group function) followed by an input operation. Do not specify the detach or evoke function with this write operation. BSCCEL sends the last record in the document with an ETX control character followed by an end-of-transmission (EOT) control character. BSCCEL then waits for input from the remote system.
- To indicate the end of the last document and to end the session, your program can use a write operation with the detach function.

When you specify GRPSEP(*OFCSYS), BSCCEL rejects any write operation that generates a null record and passes return code 832B (session error occurred) to your program. This rejection includes any write operation that specifies zero bytes as the record length. (For more information on return code 832B, see page B-32.)

When office systems documents are received, return code 0301 (no data – continue to receive) indicates the end of a document. Your program receives return code 0301 each time an ETX character is received. Return code 0300 (no data – EOT received) indicates that all the documents have been received. (For more information on return codes 0300 and 0301, see page B-6.)

Using a 6580 Displaywriter

The remote 6580 Displaywriter System must be configured to communicate in CPU mode using the 2770 protocol. When the 6580 sends a program start request to the AS/400 system, it must use the send format of *page image – text only*.

If you select a block size of 128 bytes for the 6580, the program start request must be sent as the first document in the send queue. It must consist of two 128-character lines in the continuation format.

If you select a block size of either 256 or 512 bytes for the 6580, the program start request must be sent as the first document in the send queue. It must consist of one 160-character line in the standard format.

Programming Considerations

The following programming considerations should be noted when using BSCCEL support for communications.

BSCCEL Online Messages

BSCCEL receives and sends online messages that inform it and your program of key events. It sends the messages for the following reasons:

- BSCCEL sends an online message after receiving an *EXEC or *EXEX program start request. The message informs the remote system of the success or failure of the program start request.
- If you specified RMTBSCCEL(*YES) or if the session was started by an *EXEC program start request, and your program abnormally ends, BSCCEL sends an online message to the remote system. The message informs the remote system that the session ended abnormally.

An online message generated by BSCCEL on the AS/400 system is 90 bytes long and is in one of the following formats:

ICFx BSCLn timer message-text

Where:

x = M for an informational message, E for an error message.

nnnn = the reason code associated with the failure of the program start request. Refer to Table 5-4 on page 5-8 for the reason codes of the failed program start requests.

ICFE CPI timer message-text

Where:

nnnn = the message sequence number.

BSCCEL support on an AS/400 system responds to an online message it receives as follows:

- BSCCEL checks all incoming data records for ICFM or ICFE in the first 4 bytes of the record. The first 14 characters of each of these messages is sent to the history log (QHST) and to the system operator message queue (QSYSOPR) as CPI6103.
- An ICFM message received as the result of an evoke function (in response to a program start request) is not sent to QHST or QSYSOPR and cannot be received by the application program. This message is sent by BSCCEL on the remote system to indicate that the program start request was successful.
- An ICFE message received because of an evoke function (in response to a program start request) can optionally be received by the application program. The application program receives a return code indicating that the evoke function failed, and that a message is waiting. The application program can use an input operation to receive the message, use another evoke function, or can end the session.
- If an ICFE message is received while a transaction is active, the application program receives a return code indicating that a message and a detach indication were received. If your program started the session, it must use another

voke function to begin a transaction, or use an end-of-session function or release operation to end the session. If the session was started by a program start request, your program must use an end-of-session function as the next communications operation, perform other (local) processing, or end.

- BSCCEL also recognizes a status message from the remote system without an active session. This message is normally received from a device (such as a 3741) after a session has abnormally ended. The status message is not analyzed by BSCCEL, but is sent to QHST and QSYSOPR as CPI6103. The message text is in the following format:

%x y

Where:

% = the first character of a 3741 device status message

x = a single character that identifies the type of remote system

y = one or more characters that identify the status of the remote system

Refer to your remote systems manual or call the remote system location for an explanation of the message.

Additional Online Messages BSCCEL Can Receive

BSCCEL support on an AS/400 system can receive (but not send) online messages in the following format:

ICFx SYS-nnnn message-text

Where

x = M for an informational message, E for an error message

nnnn = the message identification code

These messages are sent by a System/36 or a System/34. If your program receives one of these messages, contact the operator at the remote system for an explanation of the message.

The Reverse-Interrupt Indicator

When your program sends data, BSCCEL provides BSC control characters (used to frame a block of data) before the data is sent on the data link. To provide the correct ending control character for a block of data, BSCCEL must hold one block of data in storage until your program issues its next operation.

When sending data, your program can usually detect that the other program wants to send data with the major/minor return code 0010 (a reverse-interrupt, or RVI, indicator is received from the remote system on a successful output operation). However, because BSCCEL holds a block of data, the notification to the application program is delayed or can be potentially lost.

The following example explains this situation. In this example, Program A starts sending data to Program B on a BSC line, using unblocked data records (BLOCK(*NONE) is specified on the device description). The first record from Program A is accepted by the communications support, but is held in internal storage until Program A issues its next operation. The major/minor return code 0000 is returned to Program A, and Program A writes a second record. At this point, the first record can be sent on the data link. The second record is then held in internal storage and the return code 0000 is returned to Program A. If an RVI indi-

cator is received in response to the sending of the first record on the data link, the condition is reported to Program A at the next opportunity, when Program A writes its *third* record. At this point, the second record is sent on the data link and a return code of 0010 will be returned to Program A.

If only one or two records are sent by Program A followed by a read operation, an EOT is sent on the data link and the RVI notification is lost. When Program B receives the EOT indication, it may send data to Program A.

If Program A sends data to Program B using blocked data records, the notification of the RVI indicator will be given to Program A on the first record of the third block. If there are four records to a block, notification of an RVI indication received on the first block sent on the data link would not be given until Program A writes the ninth record (the first record of the third block).

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

This appendix contains charts that show the following for BSCEL:

- Valid language operations supported by the intersystem communications function (ICF)
- Valid operations for each programming language that supports ICF
- Data description specifications (DDS) processing keywords
- System-supplied formats

Use the high-level language operations and ICF communications functions to communicate with a program or device (such as a 3741 Data Entry Station) at a remote location.

Language Operations

The following table describes the language operations supported by ICF.

Table A-1. Language Operations

ICF Operations	Description
Open	Opens the ICF file.
Acquire	Establishes a session between the application and the remote location.
Get attributes	Used to determine the status of the session.
Read	Obtains data from a specific session.
Read-from-invited program devices	Obtains data from any session that has responded to an invite function.
Write	Passes data records from the issuing program to the other program in the transaction.
Write/Read	Allows a write operation followed by a read operation. Valid for RPG/400 ¹ only.
Release	Attempts to end a session.
Close	Closes the ICF file.

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

Table A-2 shows all the valid operations for each programming language that supports ICF (RPG/400, COBOL/400¹, C/400¹).

Note: C/400 statements are case sensitive.

ICF Operation	RPG/400 Operation Code	COBOL/400 Procedure Statement	C/400 Function
Open	OPEN	OPEN	fopen
Acquire	ACQ	ACQUIRE	QXXACQUIRE
Get Attributes	POST	ACCEPT	QXXDEVAT
Read	READ	READ	fread
Read-from-Invited-Program-Devices	READ ¹	READ ¹	QXXREADINVDEV, followed by an fread ¹
Write	WRITE	WRITE	fwrite
Write/Read	EXFMT	Not supported	Not supported
Release	REL	DROP	QXXRELEASE
Close	CLOSE	CLOSE	fclose
<p>¹ A read operation can be directed either to a specific program device or to any invited program device. The support provided by the compiler you are using determines whether to issue an ICF read or read-from-invited-program-devices operation, based on the format of the read operation. For example, if a read is issued with a specific format or terminal specified, the read operation is interpreted as an ICF read operation. Refer to the appropriate language reference manual for more information.</p>			

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

Supported DDS Keywords

Table A-3 lists the data description specifications (DDS) keywords that you use to specify the communications functions for BSCEL.

Table A-3. Valid DDS Keywords for BSCEL

DDS Keyword	Function
ALWWRT	Allow-write
CNLINVITE	Cancel-invite
DETACH	Detach
ENDGRP	End-of-group
EOS	End-of-session
EVOKE	Evoke
FAIL	Fail
INVITE	Invite
RCVDETACH	Receive detach
RCVENDGRP	Receive-end-of-group
RCVTRNRND	Receive turnaround
RECID	Record-identification
RQSWRT	Request-to-write
SECURITY	Security
SUBDEV	Subdevice selection
TIMER	Timer
VARLEN	Variable-length-data

System-Supplied Formats

Table A-4 presents the system-supplied formats that you use to specify the communications functions for BSCCL.

Table A-4. Valid System-Supplied Formats for BSCCL

System-Supplied Format	Function
\$\$CNLINV	Cancel-invite
\$\$EOS	End-of-session
\$\$EVOK	Evoke with invite
\$\$EVOKET	Evoke with detach
\$\$EVOKNI	Evoke (no invite)
\$\$FAIL	Fail
\$\$RCD	Request-write with invite
\$\$SEND	Invite or send with invite
\$\$SENDE	Send with end-of-group
\$\$SENDET	Detach or send with detach
\$\$SENDNI	Send (no invite)
\$\$TIMER	Timer

For more information on how to use the DDS keywords or the system-supplied formats, refer to the *Communications Programmer's Guide*.

Appendix B. Using BSCEL Return Codes

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

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

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

Notes:

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

Major Code 00

Major Code 00 – Operation completed successfully.

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

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

Code Indication/Action

0000 **Description:** The last operation performed by your program completed successfully. Your program can continue.

Action: Refer to Table B-1 on page B-2 for the actions that can be taken in this session.

<i>Table B-1. Actions for Return Code 0000</i>		
Type of Session	Last Operation Performed	Action Your Program Can Take
Started by a source program	Acquire or open	Use an evoke function ¹ .
	Write operation with detach function	Use an evoke function, use a release operation, continue local processing, or end your program.
	Any other output operation	Use another output operation (without an evoke function), or use an input operation.
Started by a remote program start request ²	Acquire or open	Use an input or output operation.
	Write operation with detach function	Your session has ended. Continue local processing, or end your program.
	Any other output operation	Use another output operation (without an evoke function), or use an input operation.
<p>¹ A target program is started on the remote system only if you specify RMTBSCEL(*YES) in the configuration or on the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command. If you specify RMTBSCEL(*NO), the evoke function is optional and a transaction is started without starting a target program.</p> <p>² A target program on an AS/400 system (a session started by a program start request) cannot use an evoke function. A program on the AS/400 system can use an evoke only in a session it has started. A target program can use a release operation to pass the requesting device to another program, which can then acquire the requesting device. If the target program directs another communications operation to the source program after it has used the release operation, that communications operation is unsuccessful. However, subsequent communications operations in the next program are processed normally. A target program should use an end-of-session function if the requesting device is not to be passed to another program.</p>		

0001 Description: Your program has received data on a successful input operation. It can continue to receive input until BSCEL returns an end-of-transmission indication (which allows your program to send data) or a detach indicator.

Action: Use another input operation. If your program detects an end-of-file condition or its equivalent, indicating that the last of the data was just received, it can use an output operation.

0008 Description: BSCEL received a detach indication with the last of the data on a successful input operation. Communications have ended with the program on the remote system, but the session is still active.

Action: If your program started the transaction, use another evoke function (to start another program), use a release operation (to perform local processing or to start another session), or end your program. If a remote program start request started the transaction, use an end-of-session function or end your program.

- 0010** **Description:** On a successful output operation, BSCEL received a reverse-interrupt (RVI) indication from the remote system. The program on the remote system wants to send data as soon as possible.
- Action:** Use an input operation as soon as possible.
- 0020** **Description:** On a successful input operation, BSCEL received a message and an end-of-transmission indication from the remote system. The message is in your program input buffer. Your program received the message as a result of its previous unsuccessful evoke function. (Your program received return code 831A as a result of the unsuccessful evoke function.)
- Action:** Respond to the message in the input buffer (for example, display it). Your program now has control of the session. Use another evoke function (to start another program), use a release operation (to perform local processing or to start another session), or end your program.
- 0021** **Description:** BSCEL received a message from the remote system on a successful input operation. The message is in your program input buffer. Your program can continue to receive input.
- Action:** Respond to the message in the input buffer (for example, display it), and use another input operation. If your program can detect an end-of-file condition or its equivalent, indicating that the last of the data was just received, it can use an output operation.
- 0028** **Description:** On a successful input operation, BSCEL received a detach indication with a message from the remote system. The message is in your program input buffer and describes the status of the transaction that has ended. Communications have ended with the program on the remote system, but the session is still active.
- Action:** Respond to the message in the input buffer (for example, display it). If your program started the transaction, use another evoke function (to start another program), use a release operation (to perform local processing or start another session), or end your program. If a remote program start request started the transaction, use an end-of-session function, or end your program.
- 0030** **Description:** On a successful input operation, BSCEL received a truncated message from the remote system and an end-of-transmission indicator. The message is in your program input buffer and was truncated because it was too long for the buffer. Your program received the message because its previous evoke function was not successful. (Your program received return code 831A as a result of the unsuccessful evoke function.)
- Action:** Respond to the message in the input buffer (for example, display it). Use another evoke function (to start another program), use a release operation (to perform local processing or to start another session), or end your program.
- 0031** **Description:** On a successful input operation, BSCEL received a truncated message from the remote system. The message is in your program input buffer and was truncated because it was too long for the buffer. Your program can continue to receive input.
- Action:** Respond to the message in the input buffer (for example, display it), and use another input operation. If your program can detect an end-of-file condition or its equivalent, indicating that the last of the data was just received, it can use an output operation.

0038 **Description:** On a successful input operation, BSCEL received a detach indication with a truncated message from the remote system. The message is in your program input buffer and was truncated because it was too long for the buffer. It describes the status of the transaction that has ended. Communications have ended with the program on the remote system, but the session is still active.

Action: Respond to the message in the input buffer (for example, display it). If your program started the transaction, use another evoke function (to start another program), use a release operation (to perform local processing or to start another session), or end your program. If a remote program start request started the transaction, use an end-of-session 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 remote system. However, your job is being ended (controlled).

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

Code Indication/Action

0201 **Description:** Your program has received data on a successful input operation. A job ended (controlled) request is pending.

Action: Use another input operation. Continue to receive input until BSCEL returns an end-of-transmission or detach indication. If your program can detect an end-of-file condition or its equivalent, indicating that the last of the data was just received, it can use an output operation. The recommended action is to complete the active transaction and end your program because the system eventually ends your job and forces all processing for your job to stop.

0208 **Description:** On a successful input operation, BSCEL received a detach indicator with the last of the data. Communications have ended with the program on the remote system, but the session is still active. A job ended (controlled) request is pending.

Action: If your program started the transaction, use another evoke function (to start another program), use a release operation (to perform local processing), or end your program. If a remote program start request started the transaction, use an end-of-session function or end your program. The recommended action is to complete the active transaction and end your program because the system eventually ends your job and forces all processing for your job to stop.

- 0220** **Description:** On a successful input operation, BSCCEL received a message from the remote system and an end-of-transmission indicator. The message is in your program input buffer. Your program received the message as a result of its previous unsuccessful evoke function. (Your program received return code 831A as a result of the unsuccessful evoke function.) A job ended (controlled) request is pending.
- Action:** Respond to the message in the input buffer (for example, display it). Use another evoke function (to start another program), use a release operation (to perform local processing), or end your program. The recommended action is to end your program because the system eventually cancels your job and forces all processing for your job to stop.
- 0221** **Description:** On a successful input operation, BSCCEL received a message from the remote system. The message is in your program input buffer. Your program can continue to receive input. A job ended (controlled) request is pending.
- Action:** Respond to the message in the input buffer (for example, display it), and use another input operation. Your program can continue to receive input until BSCCEL returns an end-of-transmission or detach indication. If your program can detect an end-of-file condition or its equivalent, indicating that the last of the data was just received, it can use an output operation. The recommended action is to complete the active transaction and end your program because the system eventually ends your job and forces all processing for your job to stop.
- 0228** **Description:** On a successful input operation, BSCCEL received a detach indicator with a message from the remote system. The message describes the status of the transaction that has ended. Communications have ended with the program on the remote system, but the session is still active. A job ended (controlled) request is pending.
- Action:** Respond to the message in the input buffer (for example, display it). If your program started the transaction, use another evoke function (to start another program), use a release operation (to perform local processing), or end your program. If a remote program start request started the transaction, use an end-of-session function or end your program. The recommended action is to end your program because the system eventually ends your job and forces all processing for your job to stop.
- 0230** **Description:** On a successful input operation, BSCCEL received a truncated message and an end-of-transmission indicator from the remote system. The message is in your program input buffer and was truncated because it was too long for the buffer. Your program received the message as the result of its previous unsuccessful evoke function. A job ended (controlled) request is pending. (Your program received return code 831A as a result of the unsuccessful evoke function.)
- Action:** Respond to the message in the input buffer (for example, display it). Use another evoke function (to start another program), use a release operation (to perform local processing or to start another session), or end your program. The recommended action is to end your program because the system eventually ends your job and forces all processing for your job to stop.

- 0231** **Description:** On a successful input operation, BSCEL received a truncated message from the remote system. The message is in your program input buffer and was truncated because it was too long for the buffer. Your program can continue to receive input. A job ended (controlled) request is pending.
- Action:** Respond to the message in the input buffer (for example, display it), and use another input operation. Your program can continue to receive input until BSCEL returns an end-of-transmission or detach indication. If your program can detect an end-of-file condition or its equivalent, indicating that the last of the data was just received, it can use an output operation. The recommended action is to complete the active transaction and end your program because the system eventually ends your job and forces all processing for your job to stop.
- 0238** **Description:** On a successful input operation, BSCEL received a detach indicator with a truncated message from the remote system. The message is in your program input buffer and was truncated because it was too long for the buffer. It describes the status of the transaction that has ended. Communications have ended with the program on the remote system, but the session is still active. A job ended (controlled) request is pending.
- Action:** Respond to the message in the input buffer (for example, display it). If your program started the transaction, use another evoke function (to start another program), use a release operation (to perform local processing), or end your program. If a remote program start request started the transaction, use an end-of-session function or end your program. The recommended action is to end your program because the system eventually ends your job and forces all processing for your job to stop.

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 Indication/Action

- 0300** **Description:** On a successful input operation, BSCEL received an end-of-transmission indication with no data. If you specified GRPSEP(*DEV3740) for this session, this return code indicates that the last file has been received. If you specified GRPSEP(*OFCSYS) for this session, this return code indicates that the last document has been received. If you specified neither, this return code indicates that the last record in the file has been received. The session is still active between the local and remote systems. If you specified the DDS keyword RCVTRNRND, the receive-turnaround response indicator is also set on.
- Action:** Use an input operation, use an output operation, or end the transaction using a write operation with a detach function.

- 0301** **Description:** On a successful input operation, BSCEL received no data. If you specified GRPSEP(*DEV3740) for this session, this return code indicates that the last record in a file has been received (a null record). If you specified GRPSEP(*OFCSYS) for this session, this return code indicates that the last record in a document has been received. If you specified the DDS keyword RCVENDGRP, the receive-end-of-group response indicator is also set on.
- Action:** Use another input operation. Your program can continue to receive input until BSCEL returns an end-of-transmission indication or a detach indication.
- 0308** **Description:** On a successful input operation, BSCEL received a detach indicator without data. Communications have ended with the program on the remote system, but the session is still active. If you specified the DDS keyword RCVDETACH, the receive-detach response indicator is also set on.
- Action:** If your program started the transaction, use another evoke function (to start another program), or use a release operation (to perform local processing or to start another session). If a remote program start request started the transaction, use an end-of-session function or end your program.
- 0309** **Description:** Your program is being canceled (controlled). No data was received. This return code is only applicable to the read-from-invited-program-devices operation.
- Action:** Your program can continue processing. The recommended action is to complete any active transactions, release program devices, close the file, and end your program. The system eventually changes a job ended (controlled) to a job ended (immediate) and forces all processing to stop for your job.
- Message:**
- CPF4741 (Notify)
- 0310** **Normal Indication:** The time interval specified by a timer function or by the WAITRCD value specified for the ICF file has ended.
- This return code only applies to the read-from-invited-program-devices operation.
- Because no specific program device name is associated with the completion of this operation, the program device name in the common I/O feedback area contains an *N.
- Normal Action:** Issue the operation to perform the intended function after the specified time interval has ended. If you are 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 data.
- Messages:**
- CPF4742 (Status)
CPF4743 (Status)

Major Code 04

Major Code 04 – Output exception occurred.

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

Action: Issue an input operation to receive the data, message, or return code.

Note: If your program issues another output operation before an input operation, your program receives a return code of 831C.

Code Indication/Action

0411 Description: BSCEL on the remote system has sent a message for your program. An output exception occurred because your program tried an output operation to send data when it should have used an input operation to receive the message. Your program must receive the message before it can perform an output operation.

Action: Use an input operation to receive the message.

Message:

CPF4705 (Notify)

0412 Description: One of the following occurred:

- The program on the remote system sent data for your program. An output exception occurred because your program tried an output operation to send data when it should have used an input operation to receive the data already sent from the remote system.
- An output exception occurred when your program used a cancel-invite operation because data, a message, or a return code from BSCEL was available.

Action: Use an input operation to receive the data, message, or return code.

Messages:

CPF4702 (Notify)

CPF4705 (Notify)

CPF4750 (Notify)

CPF4799 (Notify)

Major Codes 08-11

Major Codes 08-11 – Miscellaneous program errors occurred.

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

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

Code **Indication/Action**

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

Action: Your program can begin communicating with the session that is already available. If a different session is desired, use another acquire operation and specify a different program device name. (The program device name must have been specified in the PGMDEV parameter of the ADDICFDEVE or OVRICFDEVE command that preceded the program.)

Messages:

CPD4077 (Diagnostic)
CPF50A0 (Status)

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

Action: Use an invite function (a write operation with invite specified) followed by a read-from-invited-program-devices operation.

Message:

CPF4740 (Notify)

Major Code 34

Major Code 34 – Input exception occurred.

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

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

Code Indication/Action

3401 Description: The length of the data record sent by the program on the remote system is greater than the length of your program input buffer. The length of the data record received from the remote system, if available, is in the I/O feedback area's actual-record-length field.

Action: Use another input operation if your program can specify a record size that is large enough to receive the data. Otherwise, end the session, close the file, correct the file record size, and open the file again.

Messages:

CPF4768 (Notify)

CPF5319 (Notify)

3441 Description: A valid record format name was specified on the input operation, and the format selection type is *RECID. Although the data received matches one of the record formats in the file, it does not match the format specified on the input operation.

Action: Correct the program to use an input operation that does not specify record format name, or specify the correct record format name to process the data.

Message:

CPF5058 (Notify)

3451 Description: The file record size specified when the file was opened was not large enough for the data or indicators (for files defined with a nonseparate indicator area) received.

For a file using a nonseparate indicator area, the actual record length field in the device-dependent input/output feedback area contains the number of indicators specified by the format.

Action: End the session, close the file, correct the file record size, and open the file again.

Message:

CPF4768 (Notify)

Major Code 80

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

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

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

8081	Description: The operation was not successful because a system error condition was detected.
-------------	---

	Action: Communications support may have ended. Your communications configuration may need to be varied off and then on again. Do one of the following:
--	---

- | | |
|--|---|
| | <ul style="list-style-type: none">• Continue local processing• Close the file, open the file again, and establish the session again• End your program |
|--|---|

	Messages:
--	------------------

	CPF4182 (Escape)
	CPF4562 (Escape)
	CPF4582 (Escape)
	CPF4601 (Escape)
	CPF4602 (Escape)
	CPF4603 (Escape)
	CPF4701 (Diagnostic)
	CPF4707 (Diagnostic)
	CPF4708 (Diagnostic)
	CPF4709 (Diagnostic)
	CPF4710 (Diagnostic)
	CPF4711 (Diagnostic)
	CPF4712 (Diagnostic)
	CPF4713 (Diagnostic)
	CPF4714 (Diagnostic)
	CPF4715 (Diagnostic)
	CPF4716 (Diagnostic)
	CPF4717 (Diagnostic)
	CPF4722 (Diagnostic)

CPF4731 (Diagnostic)
CPF4732 (Diagnostic)
CPF5105 (Escape)
CPF5257 (Escape)
CPF5306 (Escape)
CPF5351 (Escape)
CPF5352 (Escape)
CPF5353 (Escape)
CPF5355 (Escape)
CPF5410 (Escape)
CPF5411 (Escape)
CPF5447 (Escape)
CPF5451 (Escape)

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

Action: Communications with the program on the remote system cannot be resumed 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 continue local processing or it can end.

Messages:

CPF4566 (Escape)
CPF5269 (Escape)
CPF5274 (Escape)
CPF5358 (Escape)

80B3 Description: The open operation was not successful because the ICF file you are opening is in use in another process. The session was not started.

Action: Wait for the ICF file to become available, then use another open operation. Otherwise, you may continue other processing or end your program.

Consider increasing the WAITFILE parameter with the CHGICFF or OVRICFF command to allow more time for the file to become available.

Message:

CPF4128 (Escape)

80EB Description: The open operation was not successful due to one of the following:

- An option of update or delete was used to open the file, but the option is not supported by the device.
- There was a mismatch in the specification of a separate indicator area between your program and the ICF file.

Action: If the open option specified is incorrect, close the file, choose the correct option, and open the file again.

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

change the ICF file, you must delete it and create it again with the new DDS specifications. To change the program, check the user's guide for the appropriate high-level language.

Messages:

CPF4104 (Escape)
CPF4133 (Escape)
CPF4156 (Escape)
CPF4238 (Escape)
CPF4250 (Escape)
CPF4345 (Escape)
CPF5522 (Escape)
CPF5549 (Escape)

80ED Description: The open operation was not successful because there was a file-level check between your program and the ICF file.

Action: Close the file. Recompile the program to match the file level of the ICF file, or change or override the file to LVLCHK(*NO). Open the file again.

Message:

CPF4131 (Escape)

80EF Description: The open operation was not successful because your program does not have authority to the ICF file.

Action: Close the file. Obtain authority to the ICF file from the security officer. Try the open operation again.

Message:

CPF4104 (Escape)

80F8 Description: An operation was tried to a file that has been marked in error.

Action: Close the file. Refer to messages in the job log to determine what errors occurred. Take the appropriate recovery action for those errors.

Messages:

CPF4132 (Escape)
CPF5129 (Escape)

Major Code 81

Major Code 81 – Permanent session error (nonrecoverable).

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

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

If your program started the session, you can:

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

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

- Continue processing without the session.
- End.

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

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

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

Several other minor codes indicate a line or remote system error and may require an operator to correct the error.

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

Code Indication/Action

810A Description: On an unsuccessful operation, BSCEL detected an invalid combination of values. ASCII code was specified in the line description, but transparency was specified in the device description or on the ADDICFDEVE or OVRICFDEVE command. The two values are not valid together. The session has ended.

Action: If you want to use transparency, change the value of the CODE parameter from ASCII to EBCDIC in the line description (CRTLINBSC command). If you do not want to use transparency, change the value of the TRNSPY parameter from *YES to *NO in the device description (CRTDEVBSC command), ADDICFDEVE command, or OVRICFDEVE command.

Message:

CPF5511 (Escape)

8140 Description: A cancel reply was received from the operator, program, or system default value for a previous inquiry or notify message. The session has ended.

Action: If your program started the session, use another acquire operation to start the session again. If a program start request started your program, use an end-of-session function or end your program.

Message:

CPF5104 (Escape)

8187 Description: On an unsuccessful input or output operation, BSCEL detected an invalid combination of values. Either the record length or the block length specified in the device description or ADDICFDEVE or OVRICFDEVE command is greater than the maximum buffer size specified in the line description. The maximum buffer size must be at least as large as the record length (if record blocking is not used) or the block length (if record blocking is used). The session has ended.

Action: If the record length and block length are correct for your program, change the MAXBUFFER parameter in the CHGLINBSC command. If the record length or block length are not correct, change the RCDLEN or BLKLEN parameter in the CHGDEVBSC command, ADDICFDEVE command, or OVRICFDEVE command.

If your program started the session, use another acquire operation to start the session again. If a program start request started your program, use an end-of-session function or end your program.

Message:

CPF5232 (Escape)

8191 Description: A permanent line error occurred on an output operation. The session has ended.

Action: If your program started the session, use another acquire operation to start the session again. If a program start request started your program, use an end-of-session function or end your program.

Messages:

CPF4707 (Diagnostic)

CPF4708 (Diagnostic)

CPF4709 (Diagnostic)
CPF4710 (Diagnostic)
CPF4711 (Diagnostic)
CPF4713 (Diagnostic)
CPF4714 (Diagnostic)
CPF4715 (Diagnostic)
CPF4716 (Diagnostic)
CPF4717 (Diagnostic)
CPF4722 (Diagnostic)
CPF5351 (Escape)
CPF5353 (Escape)

8192 **Description:** A permanent line error occurred on an input operation. The session has ended.

Action: If your program started the session, use another acquire operation to start the session again. If a program start request started your program, use an end-of-session function or end your program.

Messages:

CPF4709 (Diagnostic)
CPF4710 (Diagnostic)
CPF4711 (Diagnostic)
CPF4713 (Diagnostic)
CPF4714 (Diagnostic)
CPF4715 (Diagnostic)
CPF4716 (Diagnostic)
CPF4722 (Diagnostic)
CPF5228 (Escape)
CPF5351 (Escape)
CPF5353 (Escape)

8193 **Description:** A disconnect indication (for switched lines only) was received or the switched connection failed on an output operation. Either the switched connection was not established, a disconnect time-out in the remote system was exceeded, the line was unexpectedly disconnected, or your program sent invalid data. The session has ended.

Action: If the switched connection failed, check with the remote system location to determine why the remote system sent a disconnect indication. Otherwise, verify that your program did not cause a time-out and that it did not send data that was invalid. Also, verify that it did not try to send data after the transaction had ended. If your program started the session, use another acquire operation to start the session again. If a program start request started your program, use an end-of-session function or end your program.

Messages:

CPF4701 (Diagnostic)
CPF5260 (Escape)
CPF5351 (Escape)

8194 **Description:** A disconnect indication (for switched lines only) was received or the switched connection failed on an input operation. Either the switched connection was not established, a disconnect time-out in the remote system was exceeded, or the line was unexpectedly disconnected. The session has ended.

Action: If the switched connection failed, check with the remote system location to determine why the remote system sent a disconnect indication. Otherwise, verify that your program did not cause a time-out. Verify that it did not try to receive data after it had received an end-of-transaction indicator. If your program started the session, use another acquire operation to start the session again. If a program start request started your program, use an end-of-session function or end your program.

Messages:

CPF4701 (Diagnostic)
CPF5260 (Escape)
CPF5351 (Escape)

8197 Description: On an output operation, the remote system ended the transmission abnormally because it could not continue the session. The session has ended.

Action: If your program started the session, use another acquire operation to start the session again. If a program start request started your program, use an end-of-session function or end your program.

Messages:

CPF4712 (Diagnostic)
CPF5351 (Escape)

8198 Description: On an input operation, the remote system ended the transmission abnormally. The remote system ended the line transmission abnormally because it could not continue the session. The session has ended.

Action: If your program started the session, use another acquire operation to start the session again. If a program start request started your program, use an end-of-session function or end your program.

Messages:

CPF4712 (Diagnostic)
CPF5351 (Escape)

8199 Description: On an output operation, either the transmit retry parameter value or the receive retry parameter value in the line description was exceeded. This error occurred because too much time elapsed between successive data blocks being sent to, or received by, the remote system. The session has ended.

Action: Make sure the values for the transmit retry (TMTRTY) parameter and the receive retry (RCVRTY) parameter in the line description (CRTLINBSC command) are large enough for local and remote program delays. Examine your program for excessive delays between output operations. If your program started the session, use another acquire operation to start the session again. If a program start request started your program, use an end-of-session function or end your program.

Messages:

CPF4717 (Diagnostic)
CPF5351 (Escape)

819A Description: On an input operation, either the receive retry parameter value or the transmit retry parameter value in the line description was exceeded. This error occurred because too much time elapsed between successive data blocks being received from, or sent by, the remote system. The session has ended.

Action: Make sure the values for the receive retry (RCVRTY) parameter and the transmit retry (TMTRTY) parameter in the line description (CRTLINBSC command) are large enough for local and remote program delays. Examine your program for excessive delays between 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, use an end-of-session function or end your program.

Messages:

CPF4717 (Diagnostic)
CPF5351 (Escape)

819C Description: On an input operation, the length of the data block sent by the remote system was greater than the maximum buffer size specified in the line description. The session has ended.

Action: Make sure that the value for the maximum buffer size (MAXBUFFER) parameter in the line description (CRTLINBSC command) is large enough for your program. If this parameter value is correct, notify the remote system programmer and verify that the record length or block length from the program on the remote system is correct. If your program started the session, use another acquire operation to start the session again. If a program start request started your program, use an end-of-session function or end your program.

Message:

CPF5350 (Escape)

819D Description: On an input operation, one of the following occurred:

- BSCSEL received unexpected data from the remote system after your program received a detach indicator or before your program used an evoke function.
- BSCSEL received an unexpected program start request from the remote system while your session was still active.
- BSCSEL received an invalid command from the remote system while your session was still active.
- BSCSEL did not receive an EOT following a command or a message, as expected.

The session has ended.

Action: Make sure your program did not use a detach function before the transaction was expected to complete. Make sure the data sent by the program on the remote system did not contain a BSCSEL command. Also, check to see if the remote system sent a program start request while your session was still active. If your program started the session, use another acquire operation to start the session again. If a program start request started your program, use an end-of-session function or end your program.

Message:

CPF5306 (Escape)

81E9 Description: An input operation was sent, but the data received does not match any record formats in the file. The format selection option (FMTSLT parameter) for the file is *RECID. There is no format in the file defined without a RECID keyword. Therefore, there is no default record format to use. The session has ended.

Action: Verify that the data sent was correct. If the data was incorrect, change the program on the remote system so it sends the correct data. If the data was correct, add a RECID keyword definition to the file that matches the data sent, or define a record format in the file without a RECID keyword so 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, use an end-of-session function or end your program.

Message:

CPF5291 (Escape)

Major Code 82

Major Code 82 – Open or acquire operation failed.

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

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

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

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

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

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

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

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

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

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

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

Code Indication/Action

8209 Description: An open or acquire operation was not successful because a prestart job is being canceled. This may be caused by one of the following:

- An End Job (ENDJOB), End Prestart Job (ENDPJ), End Subsystem (ENDSBS), End System (ENDSYS), or Power Down System (PWRDWNSYS) command was being issued
- The maximum number of prestart jobs (MAXJOBS parameter) was reduced by the Change Prestart Job Entry (CHGPJE) command
- The maximum number of program start requests (MAXUSE parameter) was exceeded
- Too many unused prestart jobs may exist
- The prestart job had an initialization error

Action: Determine the cause of the problem, correct it, and start this job again.

Messages:

CPF4292 (Escape)

CPF5313 (Escape)

820A Description: On an unsuccessful open or acquire operation, BSCEL detected one of the following invalid combination of values:

1. The value ASCII was specified for the CODE parameter in the line description, and the value *YES was specified for the TRNSPY parameter either in the device description or on the ADDICFDEVE or OVRICFDEVE command. The two values are not valid together. The session was not started.
2. BLOCK(*USER) and RMTBSCEL(*YES) were both specified. The two parameters with the specified values are not valid together. The session was not started.

Action:

1. If you want to use transparency, change the value of the CODE parameter from ASCII to EBCDIC on the line description (CRTLINBSC command). If you do not want to use transparency, change the value of the TRNSPY parameter from *YES to *NO on the device description (CRTDEVBSC command), ADDICFDEVE command, or OVRICFDEVE command.
2. If you want to use RMTBSCEL(*YES), change the value of the BLOCK parameter to a value other than *USER. See page 4-3 for a list of BLOCK parameter values you can use. If you want to use BLOCK(*USER), change the value of the RMTBSCEL parameter from *YES to *NO.

Messages:

CPF4303 (Escape)

CPF5511 (Escape)

8233 Description: On an unsuccessful open or acquire operation, BSCEL detected a program device name that was not valid. Either an ADDICFDEVE or OVRICFDEVE command was not provided, or the program device name in your program does not match the program device name specified in the ADDICFDEVE or OVRICFDEVE command for the session being started. The session was not started.

Action: If the error is in your program, specify the correct program device name in your program. If an incorrect identifier was specified on the ADDICFDEVE or OVRICFDEVE command, specify the correct value in the PGMDEV parameter. Try the open or acquire operation again.

Messages:

CPF4103 (Escape)
CPF4116 (Escape)
CPF4288 (Escape)
CPF4747 (Escape)
CPF5068 (Escape)
CPF5070 (Escape)
CPF5355 (Escape)

8281 Description: On an unsuccessful open or acquire operation, BSCEL detected a system error condition. Either the file was previously in error, or the file could not be opened because of a system error.

Action: Communications support may have ended. Your communications configurations may need to be varied off and then varied on again. Continue local processing, close and then open the file to start the session again, or end your program.

Messages:

CPF4182 (Escape)
CPF4221 (Escape)
CPF4304 (Escape)
CPF4369 (Escape)
CPF4370 (Escape)
CPF5105 (Escape)
CPF5257 (Escape)
CPF5317 (Escape)
CPF5318 (Escape)
CPF5355 (Escape)
CPF5411 (Escape)

8282 Description: The open or acquire operation was not successful because the device for the remote location is not usable. For example, this may occur because communications have been stopped for the device by a Hold Communications Device (HLDCMNDEV) command, or a cancel reply has been taken to an error recovery message for the device. No operations should be issued to the device.

Action: Close the file. Communications with the program on the remote system cannot be resumed 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. Once the device is reset, communications can be started again by reopening the file.

Messages:

CPF4168 (Escape)
CPF4298 (Escape)
CPF4354 (Escape)
CPF5269 (Escape)
CPF5274 (Escape)
CPF5358 (Escape)

- 8287** **Description:** On an unsuccessful open or acquire operation, BSCEL detected an invalid combination of values. The value for either the record length or the block length parameter in the device description or ADDICFDEVE or OVRICFDEVE command is larger than the maximum buffer size in the line description. The maximum buffer size must be at least as large as the record length (if record blocking is not used) or the block length (if record blocking is used). The session was not started.
- Action:** If the record length and block length values are correct for your program, change the value of the MAXBUFFER parameter in the CHGLINBSC command. If the values are not correct, change the value of the RCDLEN or BLKLEN parameter in the CHGDEVBSC command, ADDICFDEVE command, or OVRICFDEVE command. Try the open or acquire operation again.
- Messages:**
- CPF4162 (Escape)
 - CPF5232 (Escape)
- 8289** **Description:** On an unsuccessful open or acquire operation, BSCEL detected an invalid combination of values. Both a record separator and transparency were specified in the device description or on the ADDICFDEVE or OVRICFDEVE command. The session was not started.
- Action:** If you want transparency, change the value of the BLOCK parameter in the device description (CHGDEVBSC command), ADDICFDEVE command, or OVRICFDEVE command. If you do not want transparency, change the value of the TRNSPY parameter from *YES to *NO. Try the open or acquire operation again.
- Messages:**
- CPF4303 (Escape)
 - CPF5511 (Escape)
- 828B** **Description:** On an unsuccessful open or acquire operation, BSCEL detected an invalid combination of values. The maximum user record length was specified to be greater than the block length in the device description or on the ADDICFDEVE or OVRICFDEVE command. The block length must be greater than or equal to the maximum user record length if blocking is used. The session was not started.
- Action:** Change the value of the RCDLEN or BLKLEN parameter on the device description (CHGDEVBSC command), ADDICFDEVE command, or OVRICFDEVE command. Try the open or acquire operation again.
- Messages:**
- CPF4117 (Escape)
 - CPF5112 (Escape)
- 828C** **Description:** On an unsuccessful open or acquire operation, BSCEL detected an invalid combination of values. Both 3740 multiple file format and intermediate text block (ITB) blocking were specified in the device description or on the ADDICFDEVE or OVRICFDEVE command. The session was not started.
- Action:** If you want to use multiple file formats, change the value of the BLOCK parameter in the device description (CHGDEVBSC command), ADDICFDEVE command, or OVRICFDEVE command. If you do not want to

use multiple file formats, change the value of the GRPSEP parameter. Try the open or acquire operation again.

Messages:

CPF4303 (Escape)
CPF5511 (Escape)

828D Description: On an unsuccessful open or acquire operation, BSCCEL detected an invalid combination of values. Both blank compression and intermediate text block (ITB) blocking were specified in the device description or on the ADDICFDEVE or OVRICFDEVE command. These values are not valid together. The session was not started.

Action: If you want blank compression, change the value of the BLOCK parameter on the device description (CHGDEVBSC command), ADDICFDEVE command, or OVRICFDEVE command. If you do not want blank compression, change the value of the DTACPR parameter. Try the open or acquire operation again.

Messages:

CPF4303 (Escape)
CPF5511 (Escape)

828E Description: On an unsuccessful open or acquire operation, BSCCEL detected one of the following invalid combinations of values:

1. Both TRUNC(*YES) and BLOCK(*ITB) were specified in the device description or on the ADDICFDEVE or OVRICFDEVE command. These values are not valid together. The session was not started.
2. Both TRUNC(*YES) and BLOCK(*NOSEP) were specified in the device description or on the ADDICFDEVE or OVRICFDEVE command. These values are not valid together. The session was not started.

Action: If you want blank truncation, change the value of the BLOCK parameter on the device description (CHGDEVBSC command), ADDICFDEVE command, or OVRICFDEVE command. If you do not want blank truncation, change the value of the TRUNC parameter from *YES to *NO. Try the open or acquire operation again.

Messages:

CPF4303 (Escape)
CPF5511 (Escape)

8290 Description: On an unsuccessful open or acquire operation, BSCCEL detected an invalid combination of values. Both blank compression and transparency were specified in the device description or on the ADDICFDEVE or OVRICFDEVE command. The session was not started.

Action: If you want blank compression, change the value of the TRNSPY parameter from *YES to *NO on the device description (CHGDEVBSC command), ADDICFDEVE command, or OVRICFDEVE command. If you do not want blank compression, change the value of the DTACPR parameter from *YES to *NO. Try the open or acquire operation again.

Messages:

CPF4303 (Escape)
CPF5511 (Escape)

- 8291** **Description:** A permanent line error occurred on an unsuccessful open or acquire operation. The session was not started.
- Action:** Try the open or acquire operation again, continue local processing, or end your program.
- Messages:**
- CPF4155 (Escape)
 - CPF4705 (Diagnostic)
 - CPF4707 (Diagnostic)
 - CPF4708 (Diagnostic)
 - CPF4709 (Diagnostic)
 - CPF4710 (Diagnostic)
 - CPF4711 (Diagnostic)
 - CPF4713 (Diagnostic)
 - CPF4714 (Diagnostic)
 - CPF4715 (Diagnostic)
 - CPF4716 (Diagnostic)
 - CPF4717 (Diagnostic)
 - CPF4722 (Diagnostic)
 - CPF5138 (Escape)
 - CPF5351 (Escape)
- 8293** **Description:** A disconnect indication (for switched lines only) was received on an unsuccessful open or acquire operation. The switched connection failed or the line was unexpectedly disconnected. The session was not started.
- Action:** Contact the remote system programmer and try to determine why the remote system sent a disconnect. Try the open or acquire operation again, continue local processing, or end your program.
- Messages:**
- CPF4701 (Diagnostic)
 - CPF5260 (Escape)
 - CPF5351 (Escape)
- 8297** **Description:** The remote system ended the transmission abnormally on an unsuccessful open or acquire operation. The remote system is ending the line transmission abnormally because it could not continue communications. The session was not started.
- Action:** Try the open or acquire operation again, continue local processing, or end your program.
- Messages:**
- CPF4712 (Diagnostic)
 - CPF5351 (Escape)
- 82A0** **Description:** On an unsuccessful open or acquire operation, BSCEL detected an invalid record separator character. The invalid record separator character was specified on the ADDICFDEVE or OVRICFDEVE command. The session was not started.
- Action:** Change the value of the record separator character in the BLOCK parameter on the ADDICFDEVE or OVRICFDEVE command. Try the open or acquire operation again.

Messages:

CPF4302 (Escape)
CPF5510 (Escape)

82A7 Description: The open or acquire operation was not successful because the specified program device was already in use. The session was not started.

Action: Wait for the program device to become available and try the open or acquire operation again, continue local processing, or end your program.

Messages:

CPF4106 (Escape)
CPF5507 (Escape)

82A8 Description: The open or acquire operation 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 trying the open or acquire operation again. If more program devices are needed, close your file and increase the MAXPGMDEV value in the ICF file.

Messages:

CPF4745 (Diagnostic)
CPF5041 (Status)

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

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

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

Verify that your program correctly handles the permanent error return codes (80xx, 81xx) it received on previously issued input and output operations. Because your program was started by a program start request, your program cannot attempt error recovery after receiving permanent error return codes. It is the responsibility of the remote program to attempt error recovery.

If the *REQUESTER device is not available and your program expects to communicate with the *REQUESTER device, the source program should send a program start request without a detach function.

If the *REQUESTER device is already acquired and your program expects to communicate with the *REQUESTER device, use the program device that acquired *REQUESTER. Your program is attempting to use two program devices that specify RMTLOCNAME(*REQUESTER) in the corresponding ICF device entry.

Messages:

CPF4366 (Escape)
CPF5381 (Escape)

82AA Description: The open or acquire operation was not successful because the remote location definition specified on the ADDICFDEVE or OVRICFDEVE command does not match any configured remote location on the system. The session was not started.

Action: Continue local processing, or close the file and end your program. Verify that the name of the remote location with which your program is trying to communicate was specified correctly with the RMTLOCNAME parameter on the ADDICFDEVE or OVRICFDEVE command.

Messages:

CPF4304 (Escape)
CPF4363 (Escape)
CPF4364 (Escape)
CPF5378 (Escape)
CPF5379 (Escape)

82AB Description: The open or acquire operation was not successful because the device description was not varied on. The session was not started.

Action: Vary on the device description and try the open or acquire operation again, use a different device description for communications, continue local processing, or end your program.

Messages:

CPF4128 (Escape)
CPF4285 (Escape)
CPF5333 (Escape)
CPF5355 (Escape)

82B3 Description: An open or acquire operation was not successful because your program is trying to use a device description that is already in use by another job.

Action: Use another device description or try an open or acquire operation using this device description when it becomes available.

Messages:

CPF4282 (Escape)
CPF5332 (Escape)

82EA Description: An open or acquire operation was not successful because RECID format selection processing was requested to a file that does not contain any record formats with a RECID keyword. The session was not started.

Action: Close the file. Change the record format selection (FMTSLT) parameter to select formats by 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 tried an open or acquire operation using a device description that is not supported for the ICF file. Your program is trying to use a device description that is not a valid communication 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: Continue local processing, or close the file and end your program. Verify that the name of the remote location with which your program is trying to communicate was specified correctly on the ADDICFDEVE or OVRICFDEVE command. If your program was trying to acquire the requesting program device, verify that your program is running in the correct environment.

Messages:

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

82EF Description: Your program tried an open or acquire operation using a device description for which the user is not authorized, or a device description that is in service mode, or with a file or library that is not found. The session was not started.

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

Messages:

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

82F5 Description: On an open or acquire operation, the remote format value for the format selection parameter is not valid. The session was not started.

Action: Change the value in the FMTSLT parameter from *RMTFMT to *PGM or *RECID on the ADDICFDEVE or OVRICFDEVE command. Try the open or acquire operation again.

Messages:

CPF4347 (Escape)
CPF5515 (Escape)

Major Code 83

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

Description: An error occurred during an I/O operation, but the session is still active. Recovery within your program might be possible.

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

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

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

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

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

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

Code Indication/Action

830B Description: Your program has tried a communications input or output operation either before the session was started or after it had ended.

The session may have ended because a release operation or end-of-session function was used, or because a permanent error occurred. Your program may have improperly handled a permanent-session error or a program-device-not-acquired error.

Action: Examine your program to ensure that an input or output operation is not tried without an active session and to ensure that the return code is handled properly. If you want your program to recover from an improperly handled error condition, use another acquire operation.

Messages:

CPF4079 (Diagnostic)
CPF4739 (Status)
CPF5067 (Escape)
CPF5068 (Escape)
CPF5070 (Escape)

831A Description: The target program ended abnormally or an evoke function failed to complete successfully. BSCEL received a message from the remote system describing the error condition. The session is still active.

Action: Use an input operation to receive the message, and print or display it. Use another evoke function to start a transaction, use an end-of-session function, or end your program.

Message:

CPF4796 (Notify)

831C Description: Your program's previous output operation received a return code indicating that the remote system sent a message or data. Your program did not properly handle the return code. This output operation is not successful because your program must first use an input operation to receive the previous message or data.

Action: Use an input operation to receive the message or data.

Message:

CPF4934 (Notify)

831E Description: Your program tried either an invalid operation or an invalid combination of operations. The session is still active. The error may have been caused by one of the following:

- Your program sent either an unrecognized operation, or an operation or function that is not supported by BSCEL.
- Your program requested an invalid combination of operations or keywords, such as a combined write-then-read operation with the invite function specified.
- Your program used an output operation with the invite or allow-write function for a file that was opened for output only.
- Your program used a close operation with a temporary close option.

Action: Try a different operation, use a release operation or end-of-session function, or end your program. Correct the error in your program before trying to communicate with the program on the remote system.

If the file was opened for output only, do not use an input operation (such as a read operation) or a write operation with a function that specifies an input operation (for example, write-with-invite or allow-write). (You can use a detach function to send an end-of-transmission indicator to the remote system.) If you need such an operation, end the session, close the file, and open the file again for input.

Messages:

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

831F Description: Your program specified invalid length or invalid data for the operation. 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 value specified for the maximum record length parameter in the device description or ADDICFDEVE or OVRICFDEVE command.
- If this was an evoke function, one of the following occurred:
 - The length of the specified target program name plus the length of the specified user-defined parameters was greater than 119 bytes.
 - The length of the specified library name or user ID was greater than 8 characters.
 - The length of the specified password was greater than 4 characters.
- 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 function was used, then the length of the user buffer was not valid.

The session is still active.

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, do one of the following:

- Change the record length in your program and recompile it.
- Change the record format length in the ICF file.
- For an output operation that used the variable-length-data-record function, verify that the length specified is less than the user record length specified for the file when it was opened.
- For an input operation, specify a data length equal to or less than the record format length, or do not specify a length.
- Change the value specified for the maximum user record length (RCDLEN) parameter in the device description (CRTDEVBS command), ADDICFDEVE command, or OVRICFDEVE command. The value for RCDLEN must be large enough for the longest record your program sends or receives.
- If the timer function was used, verify that the format of the timer interval is HHMMSS.

Messages:

CPF4762 (Notify)
CPF4765 (Notify)
CPF4767 (Notify)
CPF4797 (Notify)

8322 Description: Your program used a request-to-write function when it was not in receive state. The request-to-write function is only valid when your program is in receive state. The session is still active.

Action: Use an output operation to continue sending, use an input operation to begin receiving, use an end-of-session function, or end your program. Correct the error in your program before trying to communicate with another program on the remote system.

Message:

CPF4703 (Notify)

8327 Description: Your program used an invalid input or output operation when no transaction existed. Your program may have expected more data when there was none. Either your program or the program on the remote system has ended the transaction, or your program has not sent an evoke operation to start communicating with the program on the remote system. The session is still active.

Action: To recover, use an evoke function to start a transaction. Otherwise, use an end-of-session function or end your program. If a coding error in your program caused the error, correct your program.

Message:

CPF5098 (Notify)

8329 Description: BSCEL detected an invalid evoke function in this session. Your program was started by a program start request and cannot use evoke functions in this session.

Action: To recover, try a different operation or function. To use an evoke function in another session, use an acquire operation, then use the evoke function. Otherwise, use an end-of-session function or end your program. If a coding error in your program caused the error, correct your program.

Message:

CPF5099 (Notify)

832B Description: Your program used an invalid output operation with a record length of zero while office systems documents were being processed. When you specify GRPSEP(*OFCSYS) and a record length of zero, you cannot use the following:

- A write operation with the end-of-group function, as the first operation to a document
- A write operation with no additional functions specified

Action: If a coding error in your program caused the error, correct your program. If the data record is in error, correct it. Then try the write operation again.

Message:

CPF4798 (Notify)

- 832C** **Description:** BSCEL detected an invalid release operation following an invite function in your program. Because your program used the invite function, it cannot use a release operation to end the invited session.
- Action:** Use a read or read-from-invited-program-devices operation to complete the invite operation. Use an end-of-session function to end the session. If a coding error in your program caused the error, correct your program.
- Message:**
- CPF4769 (Notify)
- 832D** **Description:** BSCEL detected an invalid operation following an invite function in your program. Once you use an invite function, you cannot use another invite function for the same session until the first invite has been completed by a read or read-from-invited-program-devices operation.
- Action:** Use an input operation to receive the input that was invited before trying another invite function. Use an end-of-session function to end the session. If a coding error in your program caused the error, correct your program.
- Message:**
- CPF4924 (Notify)
- 832F** **Description:** Your program used an invalid evoke function or release operation before a transaction was completed. The operation or function was not performed. The session is still active.
- Action:** End the transaction by using a write operation with a detach function. Then use another evoke function to start another transaction, or use a release operation to end the session. If a coding error in your program caused the error, correct your program.
- Messages:**
- CPF4801 (Notify)
CPF5099 (Notify)
- 8334** **Description:** Your program used an evoke function without a target program name, or the target program name was longer than 8 characters.
- Action:** Correct the evoke function by supplying a target program name that is from one through eight characters long. Try the operation again.
- Message:**
- CPF4797 (Notify)
- 83E0** **Description:** Your program tried an operation using a record format that was not defined for the ICF file.
- Action:** Ensure that the name of the record format in your program is correct. Then make sure the record format is defined in the file definition. Correct the error before you try the operation again.
- Message:**
- CPF5054 (Notify)

83E8 **Description:** Your program used a cancel-invite function in a session that was not invited. The cancel-invite function is valid only when it is used after a valid invite function. The session and the transaction are still active.

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

Message:

CPF4763 (Notify)

83F6 **Description:** BSCEL detected invalid user-defined data on an unsuccessful output operation. The data was invalid for one of the following reasons:

- ASCII code was specified in the line description, and the data record contains a character that cannot be translated into ASCII.
- BLOCK(*USER) was specified in the device description or on the ADDICFDEVE or OVRICFDEVE command, and an invalid start or end character was found in a user-blocked data record.

The session is still active.

Action: Correct the data record. Try the output operation again.

Message:

CPF4706 (Notify)

83F7 **Description:** On an unsuccessful output operation, a user-blocked data record had a length of fewer than 2 characters. A length of at least 2 characters is necessary for the start and end character pair required for a BSC record.

Action: Correct the data record. Try the output operation again.

Message:

CPF4718 (Notify)

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

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

Message:

CPF5293 (Escape)

Appendix C. BSC Control Codes and Protocols

This appendix includes binary synchronous communications (BSC) line protocols and the BSC control characters. It assumes a knowledge of the BSC line protocol. This information is useful to an AS/400 programmer or remote system programmer who writes BSCEL communications programs.

The value specified for the BLKLEN or RCDLEN parameter must be greater than, or equal to, each system's transmission of data over the communication line, excluding BSC control characters. The transmitting station controls the size of the data blocks received by the system.

Table C-1 (Page 1 of 2). BSC Line Protocols and Control Characters

Name	Control Character	ASCII	EBCDIC
Start-of-heading	SOH	SOH '01'X	SOH '01'X
Start-of-text	STX	STX '02'X	STX '02'X
End-of-transmission block	ETB	ETB '17'X	ETB '26'X
End-of-text	ETX	ETX '03'X	ETX '03'X
End-of-transmission	EOT	EOT '04'X	EOT '37'X
Enquiry	ENQ	ENQ '05'X	ENQ '2D'X
Negative acknowledge	NAK	NAK '15'X	NAK '3D'X
Synchronous idle	SYN	SYN '16'X	SYN '32'X
Data link escape	DLE	DLE '10'X	DLE '10'X
Intermediate text block character	ITB	US '1F'X	IUS '1F'X
Even acknowledge	ACK0	DLE 0 '1030'X	DLE (70) '1070'X
Odd acknowledge	ACK1	DLE 1 '1031'X	DLE/ '1061'X
Wait-before-transmit— positive acknowledge	WACK	DLE; '103B'X	DLE, '106B'X
Mandatory disconnect	DISC	DLE EOT '1004'X	DLE EOT '1037'X
Intergroup separator	IGS	GS '1D'X	IGS '1D'X
Interrecord separator	IRS		IRS '1E'X
Reverse interrupt	RVI	DLE < '103C'X	DLE@ '107C'X
Temporary text delay	TTD	STX ENQ '0205'X	STX ENQ '022D'X
Transparent start-of-text	XSTX		DLE STX '1002'X
Transparent intermediate block	XITB		DLE IUS '101F'X
Transparent end-of-text	XETX		DLE ETX '1003'X

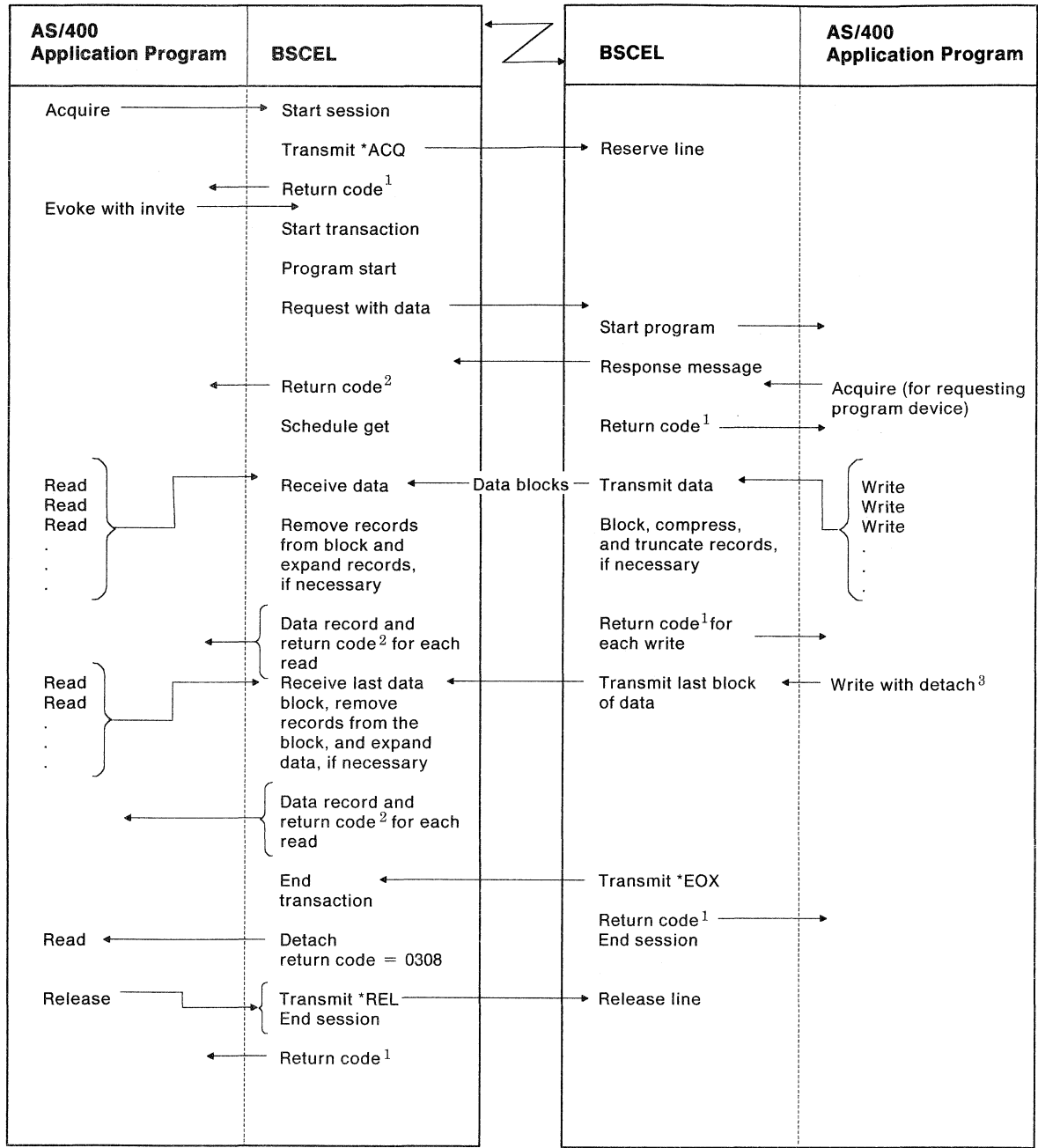
Table C-1 (Page 2 of 2). BSC Line Protocols and Control Characters

Name	Control Character	ASCII	EBCDIC
Transparent end-of-transmission block	XETB		DLE ETB '1026'X
Transparent synchronous idle	XSYN		DLE SYN '1032'X
Transparent block control	XENQ		DLE ENQ '102D'X
Transparent TTD	XTTD		DLE STX DLE ENQ '1002102D'X
Data DLE in transparent mode	XDLE		DLE DLE '1010'X

Appendix D. BSCCEL Application Flow Examples

This appendix contains application flow examples for communications between the AS/400 system and another system using BSCCEL.

Figure D-1 on page D-2 is an example of communications between two AS/400 application programs. Each program is using BSCCEL (RMTBSCCEL(*YES)) and data records are blocked.

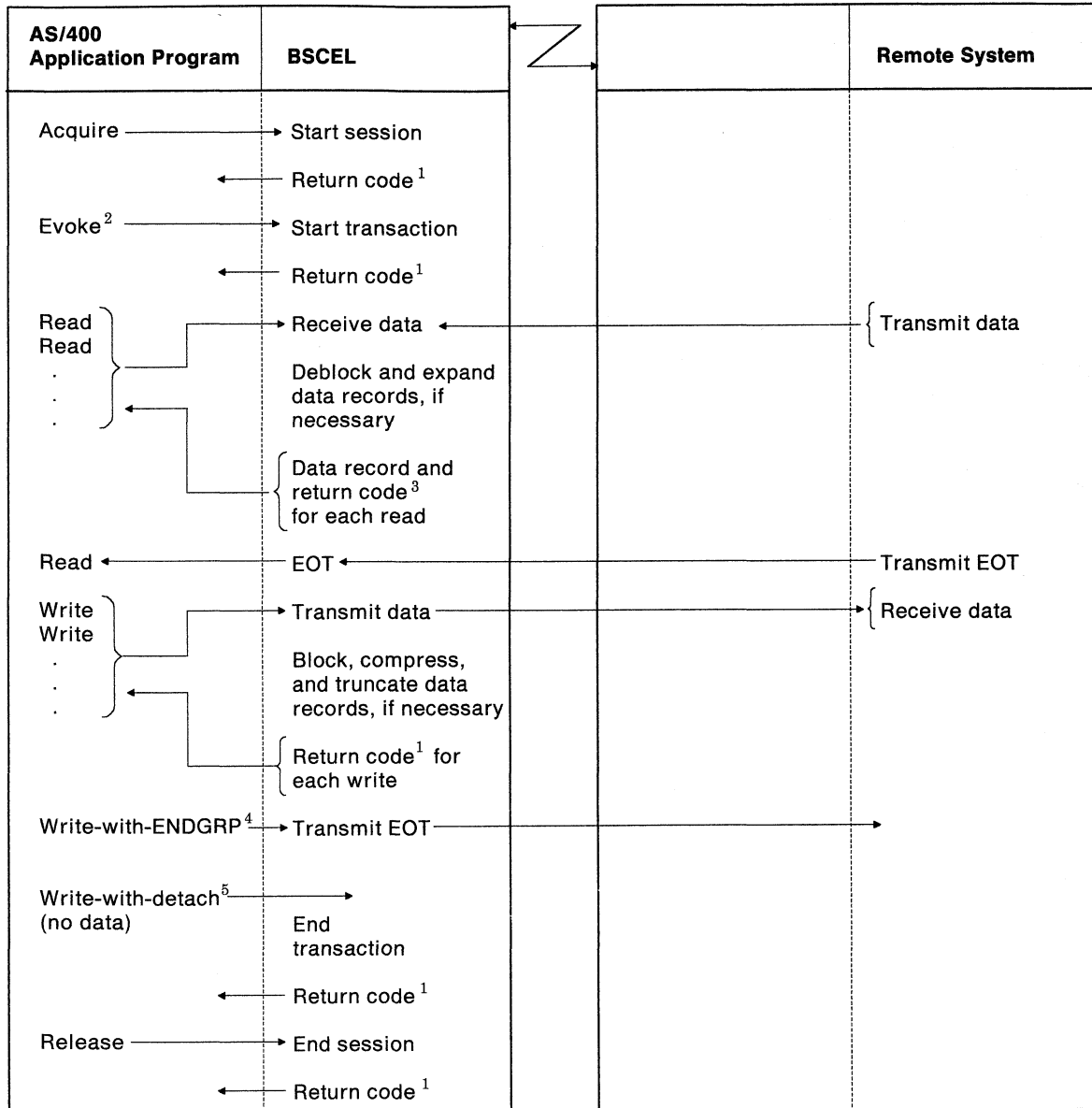


¹ Normal return code is 0000, unless an error occurs.
² Normal return code is 0001, unless an error occurs.
³ The detach function ends both the transaction and the session.

RSLS355-5

Figure D-1. Communications between Two AS/400 Application Programs

Figure D-2 is an example of communications between an AS/400 application program and a remote device that is not using BSCCEL (RMTBSCCEL(*NO)). This example uses GRPSEP(*EOT).

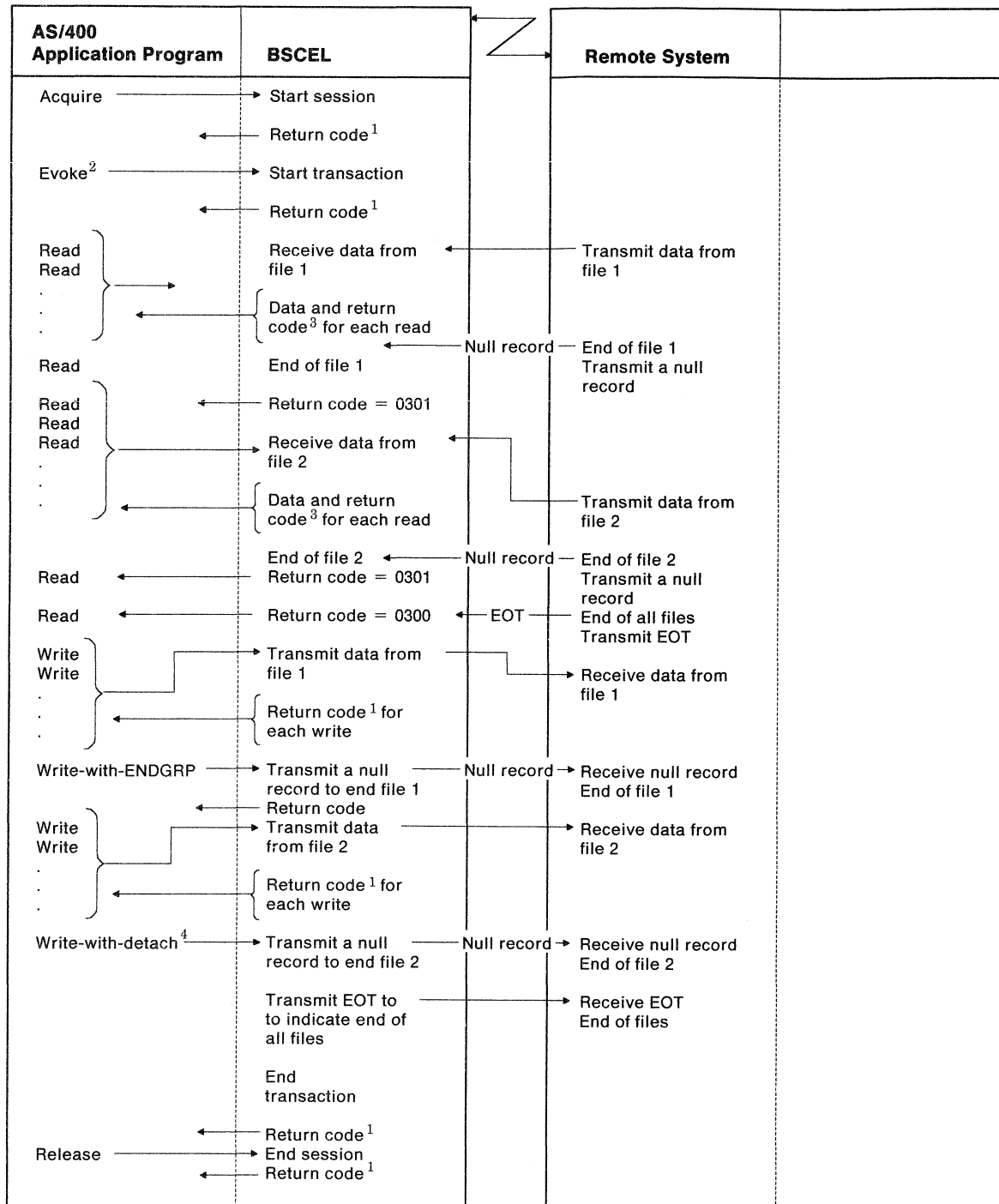


¹Normal return code is 0000, unless an error occurs.
²The evoke function is optional for RMTBSCCEL (*NO). The first input or output operation starts a transaction if the evoke function is not used.
³Normal return code is 0001, unless an error occurs.
⁴The end-of-group function is optional in this example. However, either the allow-write function or the detach function must be used to force BSCCEL to transmit EOT if end-of-group is not used.
⁵The write operation with detach function is optional, since an evoke function is not required for RMTBSCCEL (*NO).

RSLS356-4

Figure D-2. Communications between an AS/400 System and a Remote Device Not Using BSCCEL

Figure D-3 on page D-4 is an example of communication between an AS/400 application program and a remote system using 3740 multiple files (RMTBSCCEL(*NO)). This example uses GRPSEP(*DEV3740).



¹Normal return code is 0000, unless an error occurs.

²The evoke function is optional for RMTBSCCEL (*NO). The first input or output operation starts a transaction if the evoke function is not used.

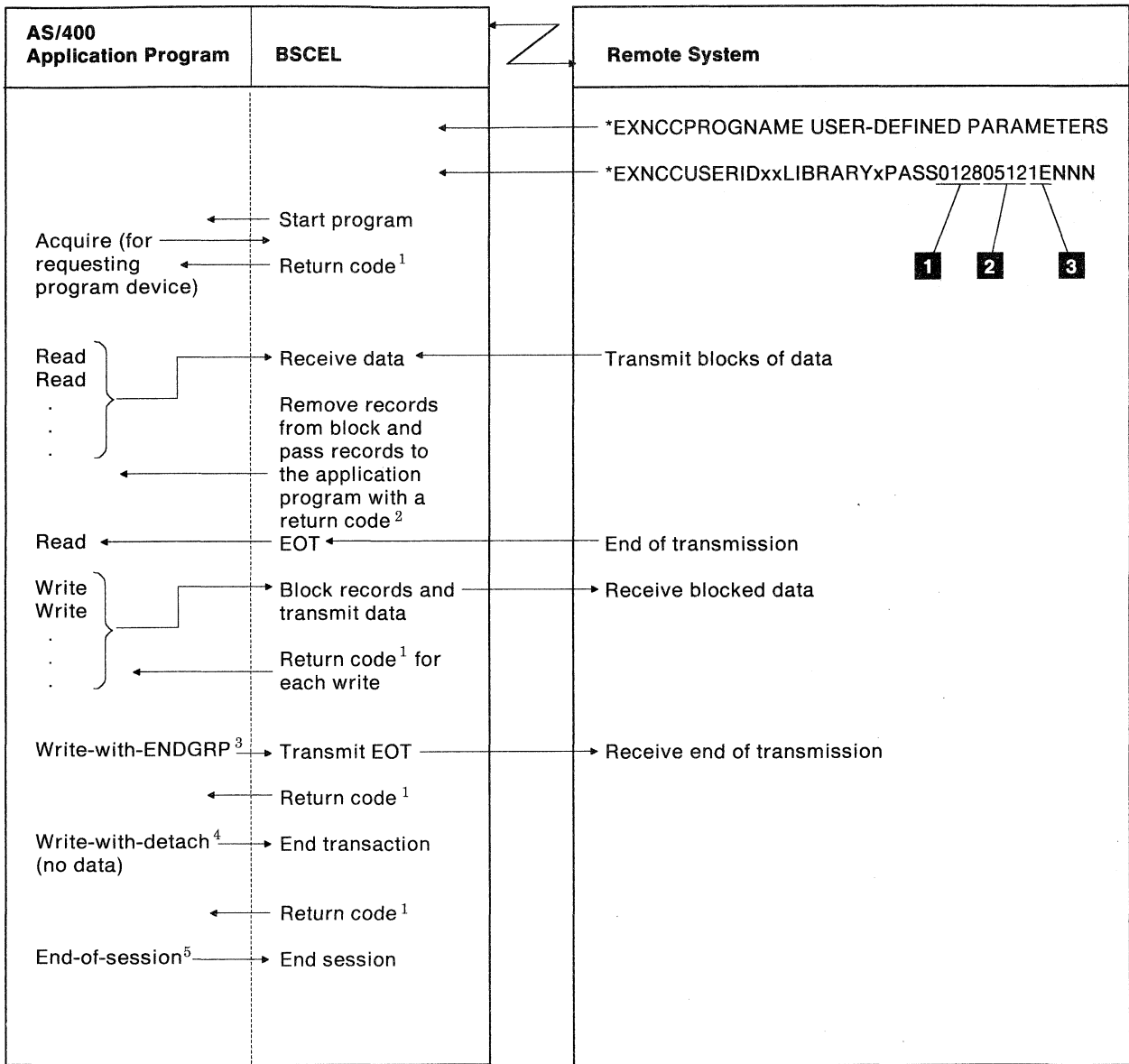
³Normal return code is 0001, unless an error occurs.

⁴The write operation with detach function is optional, since an evoke function is not required for RMTBSCCEL (*NO). If you do not use the detach function, you must then use the allow-write function to force BSCCEL to transmit a null record to end file 2, followed by an EOT.

RSL357-4

Figure D-3. Communications between an AS/400 System and a Remote System Using Multiple Files

Figure D-4 is an example of starting an AS/400 program from a remote system and the communication between an AS/400 system and the remote system following the program start. This example uses GRPSEP(*EOT).



¹ Normal return code is 0000, unless an error occurs.

² Normal return code is 0001, unless an error occurs.

³ The end-of-group function is optional in this example. However, either the allow-write function or the detach function must be used to force BSCEL to transmit EOT if end-of-group is not used.

⁴ The write operation with detach function is optional for an *EXNC program start request.

When the detach function is used, it ends both the transaction and the session

⁵ When the detach function is not used:

- Use the end-of-session function if you do not want to pass the requesting program on to another program.
- Use the release operation if you do want to pass the requesting device on to another program.

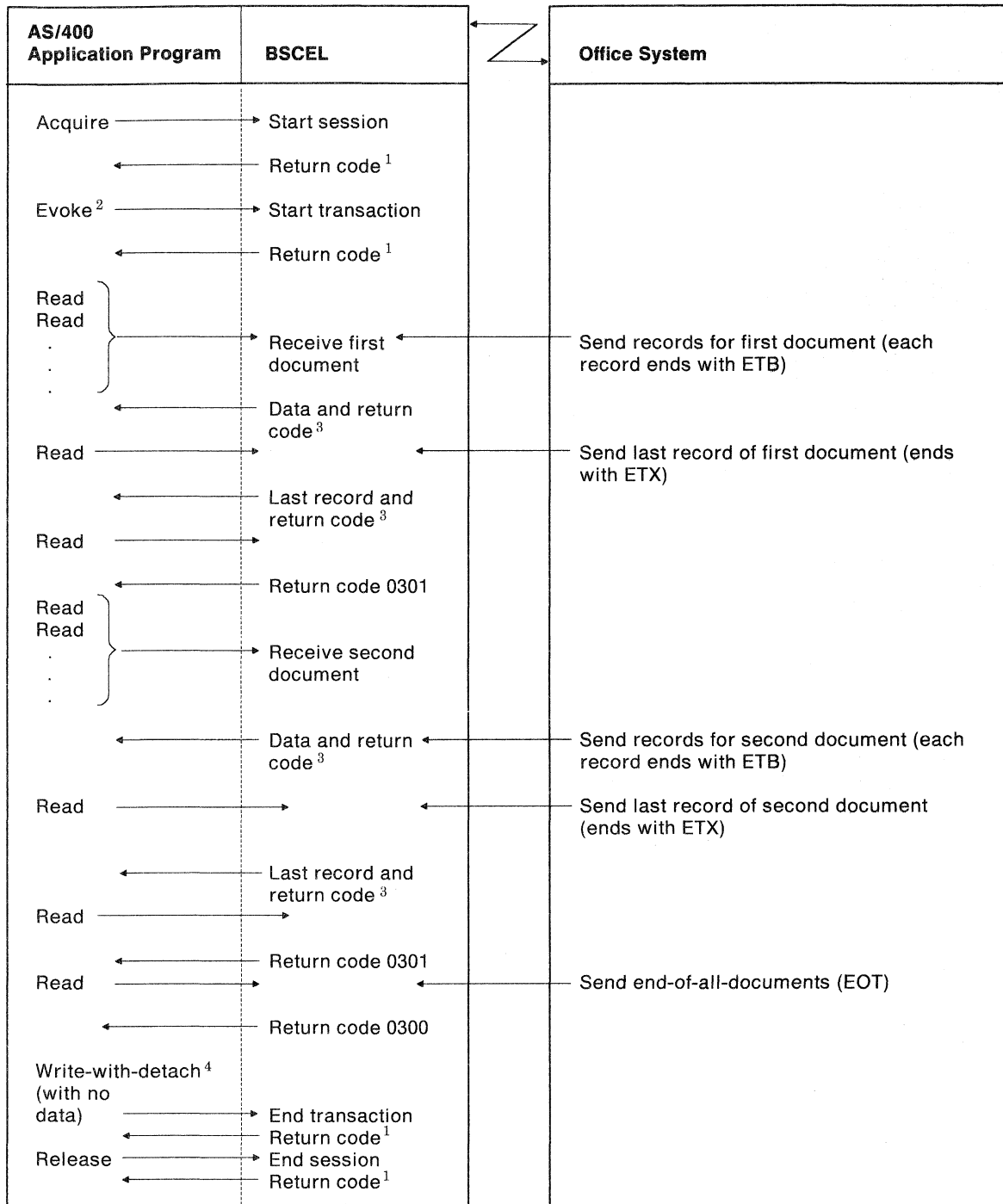
RSL358-6

Figure D-4. Starting an AS/400 System from a Remote System

The remote system in Figure D-4 on page D-5 describes the session with the following limits:

- 1** The maximum user record length is 128 bytes.
- 2** The block length is 512 bytes.
- 3** The record separator character is hex 1E.

Figure D-5 on page D-7 is an example of an AS/400 system receiving data from an office system (RMTBSCSEL(*NO)). This example uses GRPSEP(*OFCSYS).



¹ Normal return code is 0000, unless an error occurs.

² The evoke function is optional for RMTBSCCL (*NO). The first input or output operation starts a transaction if the evoke function is not used.

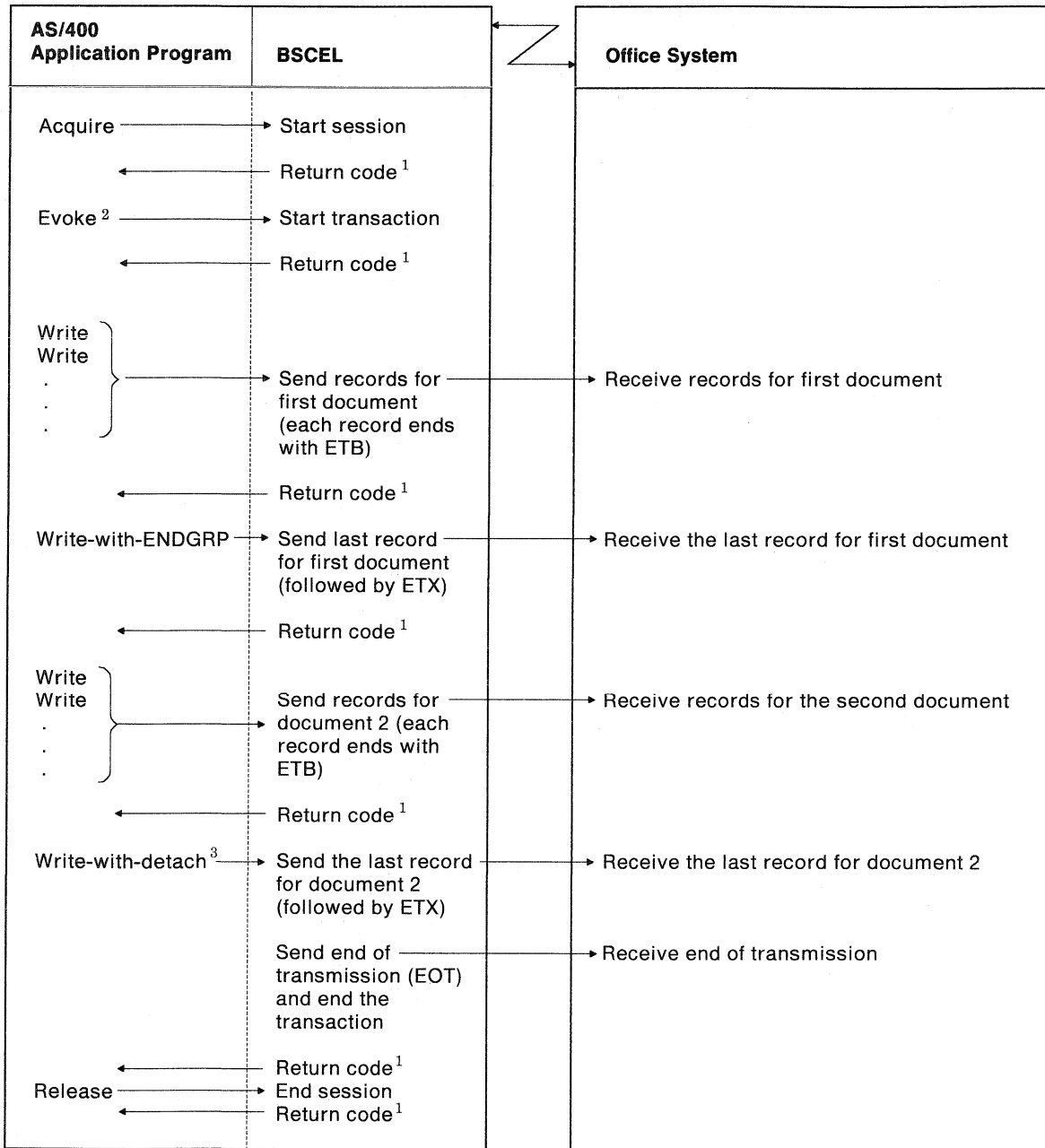
³ Normal return code is 0001, unless an error occurs.

⁴ The write operation with detach function is optional, since an evoke function is not required for RMTBSCCL (*NO).

RSLS359-7

Figure D-5. An AS/400 System Receiving Data from an Office System

Figure D-6 is an example of an AS/400 system sending data to an office system (RMTBSCSEL(*NO)). This example uses GRPSEP(*OFCSYS).



¹Normal return code is 0000, unless an error occurs.

²The evoke function is optional for RMTBSCSEL (*NO). The first input or output operation starts a transaction if the evoke function is not used.

³The write operation with detach function is optional, since an evoke function is not required for RMTBSCSEL (*NO). (If your file was opened for output only, you must use the detach function.) If you do not use the detach function, you must then use the allow-write function to force BSCSEL to send the last record for document 2 (followed by ETX), followed by an EOT.

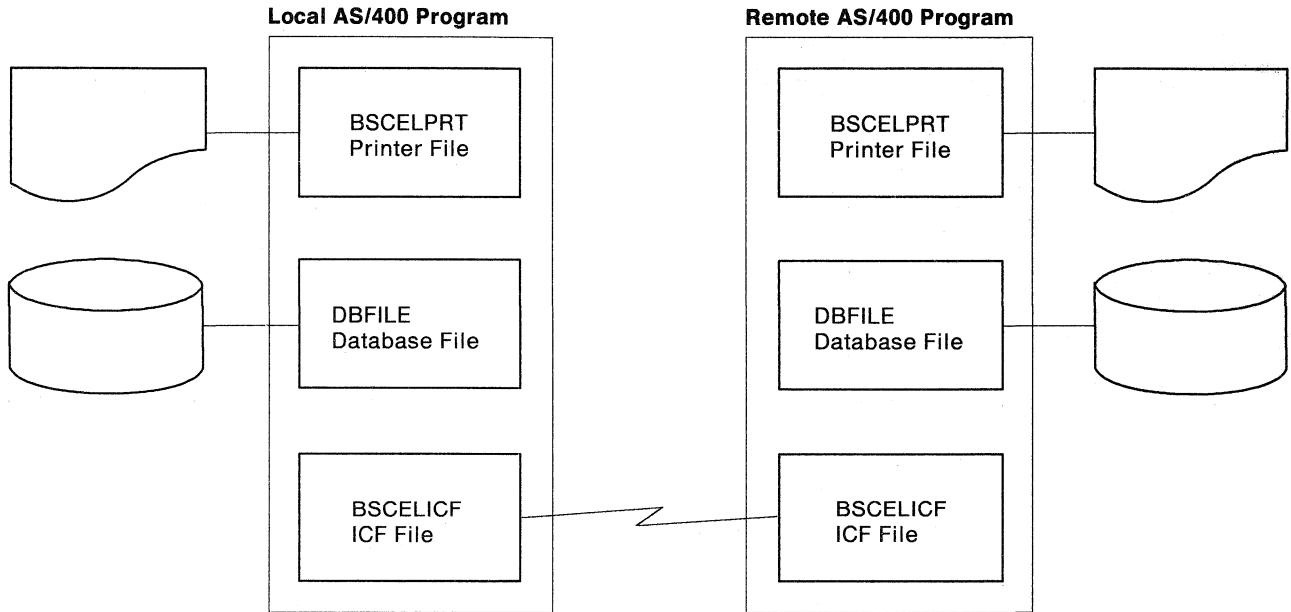
RSLS360-5

Figure D-6. An AS/400 System Sending Data to an Office System

Appendix E. Example Programs

This appendix provides sample programs to demonstrate how BSCCEL is used.

Figure E-1 shows a block diagram of a local AS/400 system program which communicates with a remote AS/400 system. The local program sends a database file, then receives and prints a file from the remote system. The remote program receives the database file and prints it, then sends a database file.



RSL362-1

Figure E-1. Communication between a Local AS/400 Program and a Remote AS/400 Program

COBOL/400 Source Program for Local System

The following describes the objects on the local system needed to run the COBOL/400 BSCCEL program.

Configuration

The following configuration commands are used to create the bisynchronous line, controller, and device descriptions used by the local system:

```
CRTLINBSC LIND(BSCELSRC) RSRNAME(LIN011) ONLINE(*NO)

CRTCTLBSC CTLD(BSCELSRC) ONLINE(*NO) LINE(BSCELSRC)

CRTDEVBSC DEVD(BSCELSRC) LOCADR(00) RMTLOCNAME(TARGET)
ONLINE(*NO) CTL(BSCELSRC) CTNWIN(*PRI)
```

Program Files

The following files are used by the local system:

BSCELICF The ICF file used to send and receive records from the remote system. This file was created by using the following command:

```
CRTICFF FILE(BSCELIB/BSCELICF) SRCFILE(BSCELIB/QDDSSRC)
        SRCMBR(BSCELICF) TEXT('ICF FILE FOR BSCEL')
```

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


```

                                Data Description Source
SEQNBR *...+...1...+...2...+...3...+...4...+...5...+...6...+...7.
100    A*****
200    A*
300    A*          ICF FILE
400    A*          USED IN BSCSEL BATCH DATA TRANSFER PROGRAM.
500    A*
600    A*****
700    A*
800    A*  FILE LEVEL INDICATORS:
900    A*
1000   A          INDARA
1100   A*
1200   A          RCVENGRP(40 'RCVD ENDGRP')
1300   A*
1400   A          RCVDETACH(35 'RCVD DETACH')
1500   A*
1600   A*****
1700   A*          BSCSEL RECORD FORMATS
1800   A*****
1900   A*
2000   A          REF(DBFILE)
2100   A          R RCVDATA
2200   A          CUSNUM      R
2300   A          CUSNAM      R
2400   A          ADDR        R
2500   A          CITY        R
2600   A          STATE       R
2700   A          ZIP         R
2800   A          CRDLMT      R
2900   A          CRDAMT      R
3000   A          R SNDDATA
3100   A          CUSNUM      R
3200   A          CUSNAM      R
3300   A          ADDR        R
3400   A          CITY        R
3500   A          STATE       R
3600   A          ZIP         R
3700   A          CRDLMT      R
3800   A          CRDAMT      R
3900   A          R EVOKPGM
4000   A          SECURITY(2 &PASS 3 &USERID)
4100   A          EVOKE(&LIB/&PGMID)
4200   A          PASS          4A  P
4300   A          USERID       8A  P
4400   A          LIB           8A  P
4500   A          PGMID        8A  P
4600   A          R DETACH      DETACH
          * * * * *   E N D   O F   S O U R C E   * * * * *

```

Figure E-2. DDS for the BSCSELICF File Used by the Local System (COBOL/400)

The command needed to define the program device entry is:

```
ADDICFDEVE FILE(BSCSELIB/BSCSELICF) PGMDEV(SOURCE) RMTLOCNAME(TARGET)
```

An OVRICFDEVE command could also be used, with the same parameters.

DBFILE The database file whose data is sent to the remote system.

The DDS for this file is illustrated in Figure E-3.

```

                                Data Description Source
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 .
100    A*****
200    A*
300    A*          * * * DBFILE * * *
400    A* DDS FOR DATABASE FILE USED IN BSCSEL DATA TRANSFER PROGRAMS.
500    A*
600    A*
700    A*****
800    A*
900    A          R DBFMT
1000   A          CUSNUM      7A      COLHDG('Customer' 'Number')
1100   A          CUSNAM      25A     COLHDG('Customer' 'Name')
1200   A          ADDR        15A     COLHDG('Address')
1300   A          CITY        15A     COLHDG('City')
1400   A          STATE       2A      COLHDG('State')
1500   A          ZIP          5S 0    COLHDG('Zip')
1600   A          CRDLMT      5S 0    COLHDG('Credit' 'Limit')
1700   A          CRDAMT      5S 0    COLHDG('Credit' 'Amount')
1800   A          K CUSNUM
                                * * * * E N D O F S O U R C E * * * *
```

Figure E-3. DDS for the Database File Used in BSCSEL Data Transfer on the Local System (COBOL/400)

BSCSELPR The printer file used to format output to a printer.

The DDS for this file is shown in Figure E-4 on page E-5.

SEQNBR	*...+...1...+...2...+...3...+...4...+...5...+...6...+...7.	Data Description Source	
100	A		REF(DBFILE)
200	A	R HDG	SKIPB(5)
300	A		5DATE EDTCDE(Y)
400	A		20TIME
500	A		60'DATABASE RECORDS RECEIVED
600	A		120'PAGE'
700	A		+1PAGNBR EDTCDE(Z)
800	A		5'CUSTOMER' SPACEB(2)
900	A		20'CUSTOMER'
1000	A		110'CREDIT'
1100	A		120'CREDIT'
1200	A		5'NUMBER' SPACEB(1)
1300	A		20'NAME'
1400	A		50'ADDRESS'
1500	A		70'CITY'
1600	A		90'STATE'
1700	A		100'ZIP'
1800	A		110'LIMIT'
1900	A		120'AMOUNT'
2000	A	R DTL	SPACEB(1)
2100	A	CUSNUM R	5
2200	A	CUSNAM R	20
2300	A	ADDR R	50
2400	A	CITY R	70
2500	A	STATE R	92
2600	A	ZIP R	100
2700	A	CRDLMT R	110EDTCDE(J)
2800	A	CRDAMT R	120EDTCDE(J)
2900	A	R ERROR	SPACEB(3)
3000	A		5'PROGRAM TERMINATED ABNORMALLY'
3100	A		5'PROGRAM DEVICE:' SPACEB(2)
3200	A	PGMDEV 10	+1
3300	A		5'RECORD FORMAT:' SPACEB(2)
3400	A	FMTNM 8	+1
3500	A		5'MAJOR CODE:' SPACEB(2)
3600	A	MAJCOD 2	+1
3700	A		5'MINOR CODE:' SPACEB(2)
3800	A	MINCOD 2	+1

* * * * * E N D O F S O U R C E * * * * *

Figure E-4. DDS for the Printer File on the Local System (COBOL/400)

Program Explanation

The following describes the COBOL/400 program on the local system, which is shown in Figure E-5 on page E-8.

- 1** The three files used in this program are specified in the input output section (file-control) portion of the program.
 - DBFILE** The name of the database file that contains the data which will be sent to the remote system.
 - BSCELICF** The name of the ICF file used to send and receive data between the local and remote systems.
 - BSCELPRT** The name of the printer file that will format output received from the remote system to a printer device.
- 2** FEEDBACK-DATA is the name of the data structure used to provide error information associated with the BSCELICF file. It contains the following information after every input or output operation to the file:
 - Record format name (FMTNM)
 - Program device name (PGMDEV)
 - Major/minor return code (MAJCOD,MINCOD)
- 3** The files are opened and the program device named SOURCE is acquired. This program device was previously added to the ICF file (BSCELICF) by the ADDICFDEVE command.
- 4** The next routine builds the evoke request and issues the write operation to (start) the program on the remote system. The word is set as the literal BSCE in this program. This is the password for the user ID (profile name on another AS/400 system) BSCEL on the remote system.

Note: With BSCEL, the maximum length of the password is 4 bytes, the maximum length of the user ID is 8 bytes, the maximum length of the library is 8 bytes, and the maximum length of the program name is 8 bytes.

When the program start request is received on the remote AS/400 system, the profile named BSCEL is verified with password BSCE, then the library BSCELIB is searched for program C85ELTGT. The profile BSCEL must specify a job description which includes the BSCELIB library in its library list.
- 5** Routine 100-SEND-DATA is called to read records from the database file and immediately write the data to the remote system, until the end-of-file condition (indicator 99) is met. The IF statement is used in the routine to avoid sending a blank record when end-of-file is reached on the database file.
- 6** Routine 110-PAGE-HEADING is called to print headings. Then routine 120-READ-DATA is called to read data from the remote system. The read operation to the ICF file causes the previous sending function to end. Any data left in buffers is sent followed by an end-of-transmission character. The program on the remote system must be prepared to send now. The read operation completes when data is available from the remote system. Detail data is printed (with a check for page overflow) until program indicator 40 (RCVENDGRP) is on. If indicator 66 comes on while printing, the heading lines are printed again. Program indicator 40 comes on when the specified end group (end-of-transmission in this case) has been reached.

- 7** This part of the program does the end-of-job processing. First, a write to the ICF file with the DETACH format causes the session with the remote system to end. Since no additional processing is needed in this program, the files are closed and the program is ended.
- 8** This routine (100-SEND-DATA) is called from **5** to read data from the data-base file and send it to the remote system.
- 9** This routine (120-READ-DATA) is called from **6** to read data from the remote system and print it. If page overflow occurs, headings are printed again.
- 10** This routine (ERROR-PARAGRAPH) is automatically called when an exception occurs during run time. Feedback data is moved to a printer record (ERROR) and the record is printed. The files are then closed and the program is ended.

STMT SEQNBR -A 1 B...2...+...3...+...4...+...5...+...6...+...7..

```

1 000100 IDENTIFICATION DIVISION.
2 000200 PROGRAM-ID.      C85ELSRC.
3 000300 AUTHOR.  JSP.
4 000400 DATE-WRITTEN. 11/87.
5 000500 DATE-COMPILED.
      06/06/88 09:01:27
6 000600 ENVIRONMENT DIVISION.
7 000700 CONFIGURATION SECTION.
8 000800 SOURCE-COMPUTER. IBM-S3X.
9 000900 OBJECT-COMPUTER. IBM-S3X.
10 001000 SPECIAL-NAMES.  I-O-FEEDBACK IS FEEDBACK-AREA.
11 001100 INPUT-OUTPUT SECTION.
12 001200 FILE-CONTROL.
13 001300     SELECT DBFILE          ASSIGN TO DATABASE-DBFILE.
14 001400     SELECT BSCELICF        ASSIGN TO WORKSTATION-BSCELICF-SI
15 001500                               ORGANIZATION IS TRANSACTION
16 001600                               CONTROL-AREA IS TRAN-CTL-AREA
17 001700                               FILE STATUS IS STATUS-IND MAJ-MIN.
18 001800     SELECT BSCELPRT      ASSIGN TO FORMATFILE-BSCELPRT
19 001900                               ORGANIZATION IS SEQUENTIAL.
20 002000 DATA DIVISION.
21 002100 FILE SECTION.
22 002200 FD  DBFILE
23 002300     LABEL RECORDS ARE STANDARD.
24 002400 01  DBREC. COPY DDS-ALL-FORMATS-I OF DBFILE.
25 +000001     05  DBFILE-RECORD PIC X(79).
+000002*     I-O FORMAT:DBFMT      FROM FILE DBFILE      OF LIBRARY BSCELIB
+000003*
+000004*THE KEY DEFINITIONS FOR RECORD FORMAT  DBFMT
+000005*  NUMBER          NAME          RETRIEVAL      TYPE      A
+000006*  0001     CUSNUM          ASCENDING     AN
26 +000007     05  DBFMT          REDEFINES DBFILE-RECORD.
27 +000008     06  CUSNUM          PIC X(7).
+000009*          Customer Number
28 +000010     06  CUSNAM          PIC X(25).
+000011*          Customer Name
29 +000012     06  ADDR           PIC X(15).
+000013*          Address
30 +000014     06  CITY           PIC X(15).
+000015*          City
31 +000016     06  STATE          PIC X(2).
+000017*          State
32 +000018     06  ZIP            PIC S9(5).
+000019*          Zip
33 +000020     06  CRDLMT          PIC S9(5).
+000021*          Credit Limit
34 +000022     06  CRDAMT          PIC S9(5).
+000023*          Credit Amount
35 002500 FD  BSCELICF
36 002600     LABEL RECORDS ARE STANDARD.
37 002700 01  ICFREC. COPY DDS-ALL-FORMATS OF BSCELICF.
38 +000001     05  BSCELICF-RECORD PIC X(79).

```

Figure E-5 (Part 1 of 7). COBOL/400 Source Program for the Local System

```

5728CB1 R01 M00 880819          COBOL SOURCE LISTING
STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..
+000002* INPUT FORMAT:RCVDATA FROM FILE BSCELICF OF LIBRARY BSCELIB
+000003*
39 +000004      05 RCVDATA-I REDEFINES BSCELICF-RECORD.
40 +000005      06 CUSNUM      PIC X(7).
+000006*      Customer Number
41 +000007      06 CUSNAM      PIC X(25).
+000008*      Customer Name
42 +000009      06 ADDR        PIC X(15).
+000010*      Address
43 +000011      06 CITY        PIC X(15).
+000012*      City
44 +000013      06 STATE       PIC X(2).
+000014*      State
45 +000015      06 ZIP         PIC S9(5).
+000016*      Zip
46 +000017      06 CRDLMT      PIC S9(5).
+000018*      Credit Limit
47 +000019      06 CRDAMT      PIC S9(5).
+000020*      Credit Amount
+000021* OUTPUT FORMAT:RCVDATA FROM FILE BSCELICF OF LIBRARY BSCELIB
+000022*
48 +000023      05 RCVDATA-0 REDEFINES BSCELICF-RECORD.
49 +000024      06 CUSNUM      PIC X(7).
+000025*      Customer Number
50 +000026      06 CUSNAM      PIC X(25).
+000027*      Customer Name
51 +000028      06 ADDR        PIC X(15).
+000029*      Address
52 +000030      06 CITY        PIC X(15).
+000031*      City
53 +000032      06 STATE       PIC X(2).
+000033*      State
54 +000034      06 ZIP         PIC S9(5).
+000035*      Zip
55 +000036      06 CRDLMT      PIC S9(5).
+000037*      Credit Limit
56 +000038      06 CRDAMT      PIC S9(5).
+000039*      Credit Amount
+000040* INPUT FORMAT:SNDDATA FROM FILE BSCELICF OF LIBRARY BSCELIB
+000041*
57 +000042      05 SNDDATA-I REDEFINES BSCELICF-RECORD.
58 +000043      06 CUSNUM      PIC X(7).
+000044*      Customer Number
59 +000045      06 CUSNAM      PIC X(25).
+000046*      Customer Name
60 +000047      06 ADDR        PIC X(15).
+000048*      Address
61 +000049      06 CITY        PIC X(15).
+000050*      City
62 +000051      06 STATE       PIC X(2).
+000052*      State
63 +000053      06 ZIP         PIC S9(5).
+000054*      Zip
64 +000055      06 CRDLMT      PIC S9(5).
+000056*      Credit Limit

```

Figure E-5 (Part 2 of 7). COBOL/400 Source Program for the Local System

```

5728CB1 R01 M00 880819          COBOL SOURCE LISTING
STMT SEQNBR -A 1 B..+...2....+...3....+...4....+...5....+...6....+...7..
65 +000057          06 CRDAMT          PIC S9(5).
    +000058*                Credit Amount
    +000059* OUTPUT FORMAT:SNDDATA    FROM FILE BSCELICF  OF LIBRARY BSCELIB
    +000060*
66 +000061          05 SNDDATA-0      REDEFINES BSCELICF-RECORD.
67 +000062          06 CUSNUM          PIC X(7).
    +000063*                Customer Number
68 +000064          06 CUSNAM          PIC X(25).
    +000065*                Customer Name
69 +000066          06 ADDR            PIC X(15).
    +000067*                Address
70 +000068          06 CITY            PIC X(15).
    +000069*                City
71 +000070          06 STATE           PIC X(2).
    +000071*                State
72 +000072          06 ZIP             PIC S9(5).
    +000073*                Zip
73 +000074          06 CRDLMT         PIC S9(5).
    +000075*                Credit Limit
74 +000076          06 CRDAMT         PIC S9(5).
    +000077*                Credit Amount
    +000078* INPUT FORMAT:EVOKPGM    FROM FILE BSCELICF  OF LIBRARY BSCELIB
    +000079*
    +000080*          05 EVOKPGM-I      REDEFINES BSCELICF-RECORD.
    +000081* OUTPUT FORMAT:EVOKPGM    FROM FILE BSCELICF  OF LIBRARY BSCELIB
    +000082*
75 +000083          05 EVOKPGM-0      REDEFINES BSCELICF-RECORD.
76 +000084          06 PASS            PIC X(4).
77 +000085          06 USERID         PIC X(8).
78 +000086          06 LIB            PIC X(8).
79 +000087          06 PGMID          PIC X(8).
    +000088* INPUT FORMAT:DETACH     FROM FILE BSCELICF  OF LIBRARY BSCELIB
    +000089*
    +000090*          05 DETACH-I      REDEFINES BSCELICF-RECORD.
    +000091* OUTPUT FORMAT:DETACH     FROM FILE BSCELICF  OF LIBRARY BSCELIB
    +000092*
    +000093*          05 DETACH-0      REDEFINES BSCELICF-RECORD.
80 002800 FD BSCELPRT
81 002900 LABEL RECORDS ARE STANDARD
82 003000 DATA RECORD IS PRINT-RECORD.
83 003100 01 PRINT-RECORD.
84 003200 COPY DDS-ALL-FORMATS-0 OF BSCELPRT.
85 +000001          05 BSCELPRT-RECORD PIC X(79).
    +000002* I-0 FORMAT:HDG          FROM FILE BSCELPRT  OF LIBRARY BSCELIB
    +000003*
    +000004*          05 HDG              REDEFINES BSCELPRT-RECORD.
    +000005* OUTPUT FORMAT:DTL        FROM FILE BSCELPRT  OF LIBRARY BSCELIB
    +000006*
86 +000007          05 DTL-0          REDEFINES BSCELPRT-RECORD.
87 +000008          06 CUSNUM          PIC X(7).
    +000009*                Customer Number
88 +000010          06 CUSNAM          PIC X(25).
    +000011*                Customer Name
89 +000012          06 ADDR            PIC X(15).
    +000013*                Address

```

Figure E-5 (Part 3 of 7). COBOL/400 Source Program for the Local System


```

5728CB1 R01 M00 880819          COBOL SOURCE LISTING
STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..
 90 +000014          06 CITY          PIC X(15).
    +000015*          City
 91 +000016          06 STATE         PIC X(2).
    +000017*          State
 92 +000018          06 ZIP           PIC S9(5).
    +000019*          Zip
 93 +000020          06 CRDLMT        PIC S9(5).
    +000021*          Credit Limit
 94 +000022          06 CRDAMT        PIC S9(5).
    +000023*          Credit Amount
    +000024* OUTPUT FORMAT:ERROR     FROM FILE BSCELPRT  OF LIBRARY BSCELIB
    +000025*
 95 +000026          05 ERROR-0       REDEFINES BSCELPRT-RECORD.
 96 +000027          06 PGMDEV        PIC X(10).
 97 +000028          06 FMTNM         PIC X(10).
 98 +000029          06 MAJCOD        PIC X(2).
 99 +000030          06 MINCOD        PIC X(2).
    +000031* OUTPUT FORMAT:RCVEND     FROM FILE BSCELPRT  OF LIBRARY BSCELIB
    +000032*
100 +000033          05 RCVEND-0      REDEFINES BSCELPRT-RECORD.
101 +000034          06 PGMDEV        PIC X(10).
102 +000035          06 FMTNM         PIC X(10).
103 +000036          06 MAJCOD        PIC X(2).
104 +000037          06 MINCOD        PIC X(2).
105 003300 WORKING-STORAGE SECTION.
106 003400 77 STATUS-IND             PIC XX.
107 003500 77 INDON                  PIC 1 VALUE B"1".
108 003600 77 INDOFF                 PIC 1 VALUE B"0".
    003700
109 003800 01 PGM-INDIC-AREA.
110 003900 05 PGM-INDIC              PIC 1 OCCURS 99 TIMES
111 004000                               INDICATOR 1.
112 004100 01 TRAN-CTL-AREA.
113 004200 05 FILLER                  PIC X(2).
114 004300 05 PGM-DEV-NAME           PIC X(10).
115 004400 05 REC-FOR-NAME           PIC X(10).
2 004500
116 004600 01 FEEDBACK-DATA.
117 004700 05 FILLER                  PIC X(37).
118 004800 05 FMTNM                   PIC X(10).
119 004900 05 FILLER                  PIC X(225).
120 005000 05 PGMDEV                  PIC X(10).
121 005100 05 FILLER                  PIC X(84).
122 005200 05 FILLER                  PIC X(34).
123 005300 05 MAJCOD                  PIC X(2).
124 005400 05 MINCOD                  PIC X(2).
    005500
125 005600 01 MAJ-MIN.
126 005700 05 MAJ                     PIC X(2).
127 005800 05 MIN                     PIC X(2).
    005900

```

Figure E-5 (Part 4 of 7). COBOL/400 Source Program for the Local System

```

5728CB1 R01 M00 880819                COBOL SOURCE LISTING
STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..
128 006000 PROCEDURE DIVISION.
    006100
    006200*****
    006300* THE FOLLOWING DECLARATIVES SECTION IS AN ERROR ROUTINE THAT *
    006400* IS RUN WHEN AN ERROR OCCURS ON THE READ OR THE WRITE OF THE *
    006500* ICF FILE "BSCSELICF". THE ROUTINE MOVES DATA FROM THE *
    006600* I-O FEEDBACK AREA TO THE "ERROR" FORMAT OF THE PRINT FILE *
    006700* "BSCSELPR". ALL OF THE FILES ARE THEN CLOSED AND THE PROGRAM *
    006800* IS ENDED. *
    006900*****
    007000 DECLARATIVES.
    007100 ERROR-SECTION SECTION.
    007200     USE AFTER STANDARD ERROR PROCEDURE ON OUTPUT.
10 007300 ERROR-PARAGRAPH.
129 007400     ACCEPT FEEDBACK-DATA FROM FEEDBACK-AREA.
130 007500     MOVE CORRESPONDING FEEDBACK-DATA TO ERROR-O.
131 007600     WRITE PRINT-RECORD FORMAT IS "ERROR".
132 007700     CLOSE BSCSELICF
    007800         DBFILE
    007900         BSCSELPR.
133 008000     STOP RUN.
    008100 END DECLARATIVES.
    008200
    008300 MAIN-PROGRAM SECTION.
    008400
    008500 000-PROCESS-GROUP.
3 008600
134 008700     OPEN I-O BSCSELICF.
135 008800     OPEN INPUT DBFILE.
136 008900     OPEN OUTPUT BSCSELPR.
    009000
137 009100     ACQUIRE "SOURCE " FOR BSCSELICF.
138 009200     MOVE "SOURCE " TO PGM-DEV-NAME.
    009300
139 009400     MOVE ZEROES TO PGM-INDIC-AREA.
    009500
    009600*****
    009700* *
4 009800* THE FOLLOWING ROUTINE EVOKES THE PROGRAM 'C85ELTGT' ON THE *
    009900* TARGET SYSTEM. *
    010000* *
    010100*****
    010200
140 010300     MOVE "BSCSELIB " TO LIB OF EVOKPGM-O.
141 010400     MOVE "C85ELTGT" TO PGMID OF EVOKPGM-O.
142 010500     MOVE "BSCE" TO PASS OF EVOKPGM-O.
143 010600     MOVE "BSCSEL " TO USERID OF EVOKPGM-O.
144 010700     WRITE ICFREC     FORMAT IS "EVOKPGM"
    010800         TERMINAL IS PGM-DEV-NAME.
    010900
    011000*****
5 011100* SEND DATABASE FILE TO THE TARGET SYSTEM. *
    011200*****
    011300
145 011400     PERFORM 100-SEND-DATA UNTIL PGM-INDIC(99) = INDON.

```

Figure E-5 (Part 5 of 7). COBOL/400 Source Program for the Local System

```

5728CB1 R01 M00 880819                COBOL SOURCE LISTING
STMT SEQNBR -A 1 B...2...3...4...5...6...7..
 011500
 011600
 011700*****
 6 011800* READ DATA FROM TARGET SYSTEM AND PRINT IT.          *
 011900*****
146 012000    PERFORM 110-PAGE-HEADING.
 012100
147 012200    PERFORM 120-READ-DATA UNTIL PGM-INDIC(40) = INDON.
 012300
 012400*****
 7 012500* DETACH FROM TARGET SYSTEM AND CLOSE FILES.          *
 012600*****
 012700
148 012800    WRITE ICFREC    FORMAT IS "DETACH"
 012900        TERMINAL IS PGM-DEV-NAME.
 013000
 013100 050-CLOSE-END.
 013200
149 013300    CLOSE BSCCELICF
 013400        DBFILE
 013500        BSCCLPRT.
 013600
150 013700    STOP RUN.
 013800
 013900
 014000* SUBROUTINES.
 014100*****
 014200*
 014300* THE FOLLOWING ROUTINE READS RECORDS FROM THE DATABASE FILE *
 014400* 'DBFILE'. WHEN 'DBFILE' REACHES 'END OF FILE' INDICATOR 99 *
 014500* IS TURNED ON.
 014600*
 8 014700*****
 014800 100-SEND-DATA.
151 014900    READ DBFILE
152 015000        AT END MOVE INDON TO PGM-INDIC(99).
153 015100    IF PGM-INDIC(99) EQUAL INDOFF THEN
154 015200        MOVE CORRESPONDING DBFMT TO SNDDATA-0
155 015300        WRITE ICFREC FORMAT IS "SNDDATA"
 015400        TERMINAL IS PGM-DEV-NAME.
 015500
 015600 110-PAGE-HEADING.
156 015700    WRITE PRINT-RECORD FORMAT IS "HDG".
 015800
 015900*****
 016000*
 016100* THE FOLLOWING ROUTINE RECEIVES RECORDS FROM THE TARGET SYSTEM *
 016200* UNTIL INDICATOR 40 IS TURNED ON. INDICATOR 40 SIGNALS THAT A *
 016300* 'RCVENDGRP' HAS BEEN RECEIVED, AND THE TARGET PROGRAM HAS *
 016400* STOPPED SENDING RECORDS. AS RECORDS ARE RECEIVED, THEY ARE *
 016500* PRINTED IN THE PRINT FILE 'BSCCLPRT'.
 016600*
 016700*****

```

Figure E-5 (Part 6 of 7). COBOL/400 Source Program for the Local System

```

5728CB1 R01 M00 880819          COBOL SOURCE LISTING
STMT SEQNBR -A 1 B...2...+...3...+...4...+...5...+...6...+...7..
 9 016800
 016900 120-READ-DATA.
157 017000  READ BSCELICF FORMAT IS "RCVDATA"
     017100  TERMINAL IS PGM-DEV-NAME
     017200  INDICATORS ARE PGM-INDIC-AREA.
158 017300  IF PGM-INDIC(40) EQUAL INDOFF THEN
159 017400  MOVE CORRESPONDING RCVDATA-I TO DTL-0
160 017500  WRITE PRINT-RECORD FORMAT IS "DTL"
161 017600  AT EOP PERFORM 110-PAGE-HEADING.
          * * * * * E N D O F S O U R C E * * * * *

5728CB1 R01 M00 880819          COBOL MESSAGES
STMT
* 22 MSGID: LBL0650 SEVERITY: 00 SEQNBR: 002200
    Message . . . . : Blocking/Deblocking for file 'DBFILE' will
    be performed by compiler-generated code.
* 37 MSGID: LBL0600 SEVERITY: 10 SEQNBR: 002700
    Message . . . . : No INPUT fields found for format EVOKPGM.
* 37 MSGID: LBL0600 SEVERITY: 10 SEQNBR: 002700
    Message . . . . : No INPUT fields found for format DETACH.
* 37 MSGID: LBL0600 SEVERITY: 10 SEQNBR: 002700
    Message . . . . : No OUTPUT fields found for format DETACH.
* 84 MSGID: LBL0600 SEVERITY: 10 SEQNBR: 003200
    Message . . . . : No INPUT fields found for format HDG.
          MESSAGE SUMMARY
TOTAL  INFO(0-4)  WARNING(5-19)  ERROR(20-29)  SEVERE(30-39)  TERMINAL
      5          1          4          0          00
          * * * * * E N D O F C O B O L M E S S A G E S * * * * *

176 source records read
153 copy records read
3 copy members processed
0 sequence errors
10 was the highest severity message issued
LBL0901 00 Program C85ELSRC created in library BSCELIB.
          * * * * * E N D O F C O M P I L A T I O N * * * * *

```

Figure E-5 (Part 7 of 7). COBOL/400 Source Program for the Local System

COBOL/400 Source Program for Remote System

The following describes the objects on the remote system needed to run the COBOL/400 BSECEL target program.

Configuration

The following configuration commands are used to create the bisynchronous line, controller, and device descriptions used by the remote system:

```
CRTLINBSC LIND(BSELTGT) RSRNAME(LIN021) ONLINE(*NO)
```

```
CRTCTLBSC CTLD(BSELTGT) ONLINE(*NO) LINE(BSELTGT)
```

```
CRTDEVBSC DEVD(BSELTGT) LOCADR(00) RMTLOCNAME(SOURCE) ONLINE(*NO)  
CTL(BSELTGT) CTNWIN(*SEC)
```

Program Files

The following files are used by the remote system:

BSELICF The ICF file used to receive then send records to the source program. This file is created by using the following command:

```
CRTICFF FILE(BSELIB/BSELICF) SRCFILE(BSELIB/QDDSSRC)  
SRCMBR(BSELICF) TEXT('ICF FILE FOR BSECEL')
```

The DDS used by this file is shown in Figure E-6 on page E-16.

```

                                Data Description Source
SEQNBR *...+...1...+...2...+...3...+...4...+...5...+...6...+...7.
100    A*****
200    A*
300    A*          ICF FILE
400    A*          USED IN BSCSEL BATCH DATA TRANSFER PROGRAM.
500    A*
600    A*****
700    A*
800    A*  FILE LEVEL INDICATORS:
900    A*
1000   A          INDARA
1100   A*
1200   A          RCVENGRP(40 'RCVD ENDGRP')
1300   A*
1400   A          RCVDETACH(35 'RCVD DETACH')
1500   A*
1600   A*****
1700   A*          BSCSEL RECORD FORMATS
1800   A*****
1900   A*
2000   A          REF(DBFILE)
2100   A          R RCVDATA
2200   A          CUSNUM    R
2300   A          CUSNAM    R
2400   A          ADDR      R
2500   A          CITY      R
2600   A          STATE     R
2700   A          ZIP       R
2800   A          CRDLMT    R
2900   A          CRDAMT    R
3000   A          R SNDDATA
3100   A          CUSNUM    R
3200   A          CUSNAM    R
3300   A          ADDR      R
3400   A          CITY      R
3500   A          STATE     R
3600   A          ZIP       R
3700   A          CRDLMT    R
3800   A          CRDAMT    R
3900   A          R EVOKPGM
4000   A          SECURITY(2 &PASS 3 &USERID)
4100   A          EVOKE(&LIB/&PGMID)
4200   A          PASS      4A P
4300   A          USERID   8A P
4400   A          LIB       8A P
4500   A          PGMID    8A P
4600   A          R DETACH  DETACH
          * * * * *   E N D   O F   S O U R C E   * * * * *

```

Figure E-6. DDS for the BSCSELICF Program File Used by the Remote System (COBOL/400)

The command needed to define the program device entry is:

```
ADDICFDEVE FILE(BSCSELIB/BSCSELICF) PGMDEV(TARGET) RMTLOCNAME(*REQUESTER)
```

An OVRICFDEVE command could also be used, with the same parameters.

DBFILE The database file whose data is sent to the source program.

The DDS for this file is shown in Figure E-7 on page E-17.

```

                                Data Description Source
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 .
100    A*****
200    A*
300    A*          * * * DBFILE * * *
400    A* DDS FOR DATABASE FILE USED IN BSCCEL DATA TRANSFER PROGRAMS.
500    A*
600    A*
700    A*****
800    A*
900    A          R DBFMT
1000   A          CUSNUM          7A          COLHDG('Customer' 'Number')
1100   A          CUSNAM          25A         COLHDG('Customer' 'Name')
1200   A          ADDR            15A         COLHDG('Address')
1300   A          CITY            15A         COLHDG('City')
1400   A          STATE           2A          COLHDG('State')
1500   A          ZIP              5S 0       COLHDG('Zip')
1600   A          CRDLMT           5S 0       COLHDG('Credit' 'Limit')
1700   A          CRDAMT           5S 0       COLHDG('Credit' 'Amount')
1800   A          K CUSNUM
                                * * * * E N D O F S O U R C E * * * *

```

Figure E-7. DDS for the Database File Used in BSCCEL Data Transfer Program on the Remote System (COBOL/400)

BSCCELPR The printer file used to format output to a printer.

The DDS for this file is shown in Figure E-8 on page E-18.

SEQNBR	*...+....1....+....2....+....3....+....4....+....5....+....6....+....7.	Data Description Source	
100	A		REF(DBFILE)
200	A	R HDG	SKIPB(5)
300	A		5DATE EDTCDE(Y)
400	A		20TIME
500	A		60'DATABASE RECORDS RECEIVED'
600	A		120'PAGE'
700	A		+1PAGNBR EDTCDE(Z)
800	A		5'CUSTOMER' SPACEB(2)
900	A		20'CUSTOMER'
1000	A		110'CREDIT'
1100	A		120'CREDIT'
1200	A		5'NUMBER' SPACEB(1)
1300	A		20'NAME'
1400	A		50'ADDRESS'
1500	A		70'CITY'
1600	A		90'STATE'
1700	A		100'ZIP'
1800	A		110'LIMIT'
1900	A		120'AMOUNT'
2000	A	R DTL	SPACEB(1)
2100	A	CUSNUM R	5
2200	A	CUSNAM R	20
2300	A	ADDR R	50
2400	A	CITY R	70
2500	A	STATE R	92
2600	A	ZIP R	100
2700	A	CRDLMT R	110EDTCDE(J)
2800	A	CRDAMT R	120EDTCDE(J)
2900	A	R ERROR	SPACEB(3)
3000	A		5'PROGRAM TERMINATED ABNORMALLY'
3100	A		5'PROGRAM DEVICE:' SPACEB(2)
3200	A	PGMDEV 10	+1
3300	A		5'RECORD FORMAT:' SPACEB(2)
3400	A	FMTNM 8	+1
3500	A		5'MAJOR CODE:' SPACEB(2)
3600	A	MAJCOD 2	+1
3700	A		5'MINOR CODE:' SPACEB(2)
3800	A	MINCOD 2	+1

* * * * * E N D O F S O U R C E * * * * *

Figure E-8. DDS for the Printer File on the Remote System (COBOL/400)

Program Explanation

The following describes the COBOL/400 program on the remote system, which is shown in Figure E-9 on page E-20.

- 1** The three files used in this program are specified in the input output section (file-control) portion of the program.
 - DBFILE** The name of the database file that contains the data which will be sent to the remote system.
 - BSCELICF** The name of the ICF file used to send and receive data between the local and remote systems.
 - BSCELPRT** The name of the printer file that will format output received from the remote system to a printer device.
- 2** FEEDBACK-DATA is the name of the data structure used to provide error information associated with the BSCELICF file. It contains the following information after every input or output operation to the file:
 - Record format name (FMTNM)
 - Program device name (PGMDEV)
 - Major/minor return code (MAJCOD,MINCOD)
- 3** The files are opened and the program device named TARGET is acquired. This program device was previously added to the ICF file (BSCELICF) by the ADDICFDEVE command.
- 4** Routine 100-PAGE-HEADING is called to print headings. Then routine 110-READ-DATA is called to read data from the other system. The read operation completes when data is available from the other system. Detail data is printed (with a check for page overflow) until program indicator 40 (RCVENDGRP) is on. If program indicator 66 comes on while printing, the heading lines are printed again. Program indicator 40 comes on when the specified end group (end-of-transmission in this case) has been reached.
- 5** Routine 120-SEND-DATA is called to read records from the database file and immediately write the data to the remote system, until the end-of-file condition (indicator 99) is met. The IF statement is used in the routine to avoid sending a blank record when end-of-file is reached on the database file.
- 6** This part of the program does the end-of-job processing. First, a read to the ICF file with the RCVDATA format causes the last group of data to be sent along with the end-of-transmission. The condition the read expects to receive is detach. Since no additional processing is needed in this program, all files are closed and the program is ended.
- 7** This routine (110-READ-DATA) is called from **4** to read data from the remote system and print it. If page overflow occurs, headings are printed again.
- 8** This routine (120-SEND-DATA) is called from **5** to read data from the database file and send it to the remote system.
- 9** This routine (ERROR-PARAGRAPH) is automatically called when an exception occurs during run time. Feedback data is moved to a printer record (ERROR) and the record is printed. The files are then closed and the program is ended.

```

5728CB1 R01 M00 880819                COBOL SOURCE LISTING
STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..
 1 000100 IDENTIFICATION DIVISION.
 2 000200 PROGRAM-ID.          C85ELTGT.
 3 000300 AUTHOR. JSP.
 4 000400 DATE-WRITTEN. 11/87.
 5 000500 DATE-COMPILED. 06/06/88 22:47:20
 6 000600 ENVIRONMENT DIVISION.
 7 000700 CONFIGURATION SECTION.
 8 000800 SOURCE-COMPUTER. IBM-S3X.
 9 000900 OBJECT-COMPUTER. IBM-S3X.
10 001000 SPECIAL-NAMES.      I-O-FEEDBACK IS FEEDBACK-AREA.
11 001100 INPUT-OUTPUT SECTION.
12 001200 FILE-CONTROL.
13 001300     SELECT DBFILE          ASSIGN TO DATABASE-DBFILE.
14 001400     SELECT BSCELICF        ASSIGN TO WORKSTATION-BSCELICF-SI
15 001500                                ORGANIZATION IS TRANSACTION
16 001600                                CONTROL-AREA IS TRAN-CTL-AREA
17 001700                                FILE STATUS IS STATUS-IND MAJ-MIN.
18 001800     SELECT BSCELPRT       ASSIGN TO FORMATFILE-BSCELPRT
19 001900                                ORGANIZATION IS SEQUENTIAL.
20 002000 DATA DIVISION.
21 002100 FILE SECTION.
22 002200 FD  DBFILE
23 002300     LABEL RECORDS ARE STANDARD.
24 002400 01  DBREC. COPY DDS-ALL-FORMATS-I OF DBFILE.
25 +000001     05  DBFILE-RECORD PIC X(79).
+000002*     I-O FORMAT:DBFMT      FROM FILE DBFILE      OF LIBRARY BSCELIB
+000003*
+000004*THE KEY DEFINITIONS FOR RECORD FORMAT DBFMT
+000005*  NUMBER          NAME          RETRIEVAL      TYPE      A
+000006*  0001      CUSNUM          ASCENDING      AN
26 +000007      05  DBFMT          REDEFINES DBFILE-RECORD.
27 +000008      06  CUSNUM          PIC X(7).
+000009*          Customer Number
28 +000010      06  CUSNAM          PIC X(25).
+000011*          Customer Name
29 +000012      06  ADDR           PIC X(15).
+000013*          Address
30 +000014      06  CITY           PIC X(15).
+000015*          City
31 +000016      06  STATE          PIC X(2).
+000017*          State
32 +000018      06  ZIP            PIC S9(5).
+000019*          Zip
33 +000020      06  CRDLMT          PIC S9(5).
+000021*          Credit Limit
34 +000022      06  CRDAMT          PIC S9(5).
+000023*          Credit Amount
35 002500 FD  BSCELICF
36 002600     LABEL RECORDS ARE STANDARD.
37 002700 01  ICFREC. COPY DDS-ALL-FORMATS OF BSCELICF.
38 +000001     05  BSCELICF-RECORD PIC X(79).

```

Figure E-9 (Part 1 of 7). COBOL/400 Source Program for the Remote System

```

5728CB1 R01 M00 880819          COBOL SOURCE LISTING
STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..
+000002* INPUT FORMAT:RCVDATA FROM FILE BSCELICF OF LIBRARY BSCELIB
+000003*
39 +000004      05 RCVDATA-I REDEFINES BSCELICF-RECORD.
40 +000005      06 CUSNUM PIC X(7).
+000006* Customer Number
41 +000007      06 CUSNAM PIC X(25).
+000008* Customer Name
42 +000009      06 ADDR PIC X(15).
+000010* Address
43 +000011      06 CITY PIC X(15).
+000012* City
44 +000013      06 STATE PIC X(2).
+000014* State
45 +000015      06 ZIP PIC S9(5).
+000016* Zip
46 +000017      06 CRDLMT PIC S9(5).
+000018* Credit Limit
47 +000019      06 CRDAMT PIC S9(5).
+000020* Credit Amount
+000021* OUTPUT FORMAT:RCVDATA FROM FILE BSCELICF OF LIBRARY BSCELIB
+000022*
48 +000023      05 RCVDATA-0 REDEFINES BSCELICF-RECORD.
49 +000024      06 CUSNUM PIC X(7).
+000025* Customer Number
50 +000026      06 CUSNAM PIC X(25).
+000027* Customer Name
51 +000028      06 ADDR PIC X(15).
+000029* Address
52 +000030      06 CITY PIC X(15).
+000031* City
53 +000032      06 STATE PIC X(2).
+000033* State
54 +000034      06 ZIP PIC S9(5).
+000035* Zip
55 +000036      06 CRDLMT PIC S9(5).
+000037* Credit Limit
56 +000038      06 CRDAMT PIC S9(5).
+000039* Credit Amount
+000040* INPUT FORMAT:SNDDATA FROM FILE BSCELICF OF LIBRARY BSCELIB
+000041*
57 +000042      05 SNDDATA-I REDEFINES BSCELICF-RECORD.
58 +000043      06 CUSNUM PIC X(7).
+000044* Customer Number
59 +000045      06 CUSNAM PIC X(25).
+000046* Customer Name
60 +000047      06 ADDR PIC X(15).
+000048* Address
61 +000049      06 CITY PIC X(15).
+000050* City
62 +000051      06 STATE PIC X(2).
+000052* State
63 +000053      06 ZIP PIC S9(5).
+000054* Zip

```

Figure E-9 (Part 2 of 7). COBOL/400 Source Program for the Remote System

```

5728CB1 R01 M00 880819          COBOL SOURCE LISTING
STMT SEQNBR -A 1 B...2....+....3....+....4....+....5....+....6....+....7..
 64 +000055          06 CRDLMT          PIC S9(5).
    +000056*          Credit Limit
 65 +000057          06 CRDAMT          PIC S9(5).
    +000058*          Credit Amount
    +000059* OUTPUT FORMAT:SNDDATA      FROM FILE BSCELICF  OF LIBRARY BSCELIB
    +000060*
 66 +000061          05 SNDDATA-0      REDEFINES BSCELICF-RECORD.
 67 +000062          06 CUSNUM          PIC X(7).
    +000063*          Customer Number
 68 +000064          06 CUSNAM          PIC X(25).
    +000065*          Customer Name
 69 +000066          06 ADDR           PIC X(15).
    +000067*          Address
 70 +000068          06 CITY           PIC X(15).
    +000069*          City
 71 +000070          06 STATE          PIC X(2).
    +000071*          State
 72 +000072          06 ZIP            PIC S9(5).
    +000073*          Zip
 73 +000074          06 CRDLMT          PIC S9(5).
    +000075*          Credit Limit
 74 +000076          06 CRDAMT          PIC S9(5).
    +000077*          Credit Amount
    +000078* INPUT FORMAT:EVOKPGM      FROM FILE BSCELICF  OF LIBRARY BSCELIB
    +000079*
    +000080*          05 EVOKPGM-I      REDEFINES BSCELICF-RECORD.
    +000081* OUTPUT FORMAT:EVOKPGM      FROM FILE BSCELICF  OF LIBRARY BSCELIB
    +000082*
 75 +000083          05 EVOKPGM-0      REDEFINES BSCELICF-RECORD.
 76 +000084          06 PASS           PIC X(4).
 77 +000085          06 USERID          PIC X(8).
 78 +000086          06 LIB            PIC X(8).
 79 +000087          06 PGMID          PIC X(8).
    +000088* INPUT FORMAT:DETACH        FROM FILE BSCELICF  OF LIBRARY BSCELIB
    +000089*
    +000090*          05 DETACH-I        REDEFINES BSCELICF-RECORD.
    +000091* OUTPUT FORMAT:DETACH        FROM FILE BSCELICF  OF LIBRARY BSCELIB
    +000092*
    +000093*          05 DETACH-0        REDEFINES BSCELICF-RECORD.
80 002800 FD BSCELPRT
81 002900 LABEL RECORDS ARE STANDARD
82 003000 DATA RECORD IS PRINT-RECORD.
83 003100 01 PRINT-RECORD.
84 003200 COPY DDS-ALL-FORMATS-0 OF BSCELPRT.
85 +000001          05 BSCELPRT-RECORD PIC X(79).
    +000002* I-0 FORMAT:HDG           FROM FILE BSCELPRT  OF LIBRARY BSCELIB
    +000003*
    +000004*          05 HDG               REDEFINES BSCELPRT-RECORD.
    +000005* OUTPUT FORMAT:DTL          FROM FILE BSCELPRT  OF LIBRARY BSCELIB
    +000006*
 86 +000007          05 DTL-0          REDEFINES BSCELPRT-RECORD.
 87 +000008          06 CUSNUM          PIC X(7).
    +000009*          Customer Number
 88 +000010          06 CUSNAM          PIC X(25).
    +000011*          Customer Name

```

Figure E-9 (Part 3 of 7). COBOL/400 Source Program for the Remote System

```

STMT SEQNBR -A 1 B...+....2....+....3....+....4....+....5....+....6....+....7..
 89 +000012      06 ADDR          PIC X(15).
    +000013*      Address
 90 +000014      06 CITY          PIC X(15).
    +000015*      City
 91 +000016      06 STATE        PIC X(2).
    +000017*      State
 92 +000018      06 ZIP          PIC S9(5).
    +000019*      Zip
 93 +000020      06 CRDLMT       PIC S9(5).
    +000021*      Credit Limit
 94 +000022      06 CRDAMT       PIC S9(5).
    +000023*      Credit Amount
    +000024* OUTPUT FORMAT:ERROR FROM FILE BSCELPRT  OF LIBRARY BSCELIB
    +000025*
 95 +000026      05 ERROR-0      REDEFINES BSCELPRT-RECORD.
 96 +000027      06 PGMDEV       PIC X(10).
 97 +000028      06 FMTNM        PIC X(10).
 98 +000029      06 MAJCOD       PIC X(2).
 99 +000030      06 MINCOD       PIC X(2).
    +000031* OUTPUT FORMAT:RCVEND FROM FILE BSCELPRT  OF LIBRARY BSCELIB
    +000032*
100 +000033      05 RCVEND-0     REDEFINES BSCELPRT-RECORD.
101 +000034      06 PGMDEV       PIC X(10).
102 +000035      06 FMTNM        PIC X(10).
103 +000036      06 MAJCOD       PIC X(2).
104 +000037      06 MINCOD       PIC X(2).
105 003300 WORKING-STORAGE SECTION.
106 003400 77 STATUS-IND        PIC XX.
107 003500 77 INDON             PIC 1 VALUE B"1".
108 003600 77 INDOFF           PIC 1 VALUE B"0".
109 003700 01 PGM-INDIC-AREA.
110 003800 05 PGM-INDIC        PIC 1 OCCURS 99 TIMES
111 003900                      INDICATOR 1.
112 004000 01 TRAN-CTL-AREA.
113 004100 05 FILLER           PIC X(2).
114 004200 05 PGM-DEV-NAME     PIC X(10).
115 004300 05 RCD-FMT-NAME     PIC X(10).
 2 004400
116 004500 01 FEEDBACK-DATA.
117 004600 05 FILLER           PIC X(37).
118 004700 05 FMTNM           PIC X(10).
119 004800 05 FILLER           PIC X(225).
120 004900 05 PGMDEV           PIC X(10).
121 005000 05 FILLER           PIC X(84).
122 005100 05 FILLER           PIC X(34).
123 005200 05 MAJCOD           PIC X(2).
124 005300 05 MINCOD           PIC X(2).
    005400
125 005500 01 MAJ-MIN.
126 005600 05 MAJ             PIC X(2).
127 005700 05 MIN             PIC X(2).
    005800

```

Figure E-9 (Part 4 of 7). COBOL/400 Source Program for the Remote System

```

5728CB1 R01 M00 880819                COBOL SOURCE LISTING
STMT SEQNBR -A 1 B..+....2....+....3....+....4....+....5....+....6....+....7..
128 005900 PROCEDURE DIVISION.
    006000
    006100*****
    006200* THE FOLLOWING DECLARATIVE SECTION IS AN ERROR ROUTINE THAT *
    006300* IS RUN WHEN AN ERROR OCCURS ON THE READ OR THE WRITE OF THE *
    006400* ICF FILE "BSCELICF". THE ROUTINE MOVES DATA FROM THE *
    006500* I-O FEEDBACK AREA TO THE "ERROR" FORMAT OF THE PRINT FILE *
    006600* "BSCELPRT". ALL OF THE FILES ARE THEN CLOSED AND THE PROGRAM *
    006700* IS ENDED. *
    006800*****
    006900 DECLARATIVES.
    007000 ERROR-SECTION SECTION.
    007100     USE AFTER STANDARD ERROR PROCEDURE ON OUTPUT.
    9 007200 ERROR-PARAGRAPH.
129 007300     ACCEPT FEEDBACK-DATA FROM FEEDBACK-AREA.
130 007400     MOVE CORRESPONDING FEEDBACK-DATA TO ERROR-0.
131 007500     WRITE PRINT-RECORD FORMAT IS "ERROR".
132 007600     CLOSE BSCELICF
    007700         DBFILE
    007800         BSCELPRT.
133 007900     STOP RUN.
    008000 END DECLARATIVES.
    008100
    008200 MAIN-PROGRAM SECTION.
    008300
    008400 000-PROCESS-GROUP.
    3 008500
134 008600     OPEN I-O BSCELICF.
135 008700     OPEN INPUT DBFILE.
136 008800     OPEN OUTPUT BSCELPRT.
    008900
137 009000     ACQUIRE "TARGET " FOR BSCELICF.
138 009100     MOVE "TARGET " TO PGM-DEV-NAME.
    009200
139 009300     MOVE ZEROES TO PGM-INDIC-AREA.
    009400
    009500
    009600*****
    4 009700* READ DATA FROM SOURCE SYSTEM. *
    009800*****
    009900
140 010000     PERFORM 100-PAGE-HEADING.
141 010100     PERFORM 110-READ-DATA UNTIL PGM-INDIC(40) = INDON.
    010200
    010300*****
    5 010400* READ DATABASE FILE AND SEND IT TO SOURCE SYSTEM. *
    010500*****
    010600
142 010700     PERFORM 120-SEND-DATA UNTIL PGM-INDIC(99) = INDON.
    010800

```

Figure E-9 (Part 5 of 7). COBOL/400 Source Program for the Remote System

```

5728CB1 R01 M00 880819                COBOL SOURCE LISTING
STMT SEQNBR -A 1 B...2...+...3...+...4...+...5...+...6...+...7..
010900*****
011000*
011100* THE FOLLOWING ROUTINE RECEIVES THE "DETACH" KEYWORD FROM THE *
011200* SOURCE PROGRAM, CLOSES ALL OF THE FILES AND ENDS THE *
011300* PROGRAM. *
011400* *
6 011500*****
143 011600 READ BSCSELICF FORMAT IS "RCVDATA".
144 011700 CLOSE BSCSELICF
011800 BSCSELPR
011900 DBFILE.
012000
145 012100 STOP RUN.
012200
012300* SUBROUTINES.
012400
012500 100-PAGE-HEADING.
146 012600 WRITE PRINT-RECORD FORMAT IS "HDG".
012700
012800*****
012900*
013000* THE FOLLOWING ROUTINE RECEIVES RECORDS FROM THE SOURCE SYSTEM *
013100* UNTIL INDICATOR 40 IS TURNED ON. INDICATOR 40 SIGNALS THAT A *
013200* 'RCVENDGRP' HAS BEEN RECEIVED, AND THE SOURCE PROGRAM HAS *
013300* STOPPED SENDING RECORDS. AS RECORDS ARE RECEIVED THEY ARE *
013400* PRINTED IN THE PRINT FILE 'BSCSELPR'. *
013500* *
013600*****
7 013700 110-READ-DATA.
147 013800 READ BSCSELICF FORMAT IS "RCVDATA"
013900 TERMINAL IS PGM-DEV-NAME
014000 INDICATORS ARE PGM-INDIC-AREA.
148 014100 IF PGM-INDIC(40) EQUAL INDOFF THEN
149 014200 MOVE CORRESPONDING RCVDATA-I TO DTL-0
150 014300 WRITE PRINT-RECORD FORMAT IS "DTL"
151 014400 AT EOP PERFORM 100-PAGE-HEADING.
014500
014600*****
014700*
014800* THE FOLLOWING ROUTINE READS RECORDS FROM THE DATABASE FILE *
014900* 'DBFILE'. WHEN 'DBFILE' REACHES 'END OF FILE', INDICATOR 99 *
015000* IS TURNED ON. *
015100* *
015200*****
8 015300 120-SEND-DATA.
152 015400 READ DBFILE
153 015500 AT END MOVE INDON TO PGM-INDIC(99).
154 015600 MOVE CORRESPONDING DBFMT TO SNDDATA-0.
155 015700 IF PGM-INDIC(99) EQUAL INDOFF THEN
156 015800 WRITE ICFREC FORMAT IS "SNDDATA"
015900 TERMINAL IS PGM-DEV-NAME.
016000
* * * * * E N D O F S O U R C E * * * * *

```

Figure E-9 (Part 6 of 7). COBOL/400 Source Program for the Remote System

```

5728CB1 R01 M00 880819                COBOL MESSAGES
STMT
* 22 MSGID: LBL0650 SEVERITY: 00 SEQNBR: 002200
    Message . . . . : Blocking/Deblocking for file 'DBFILE' will
                        be performed by compiler-generated code.
* 37 MSGID: LBL0600 SEVERITY: 10 SEQNBR: 002700
    Message . . . . : No INPUT fields found for format EVOKPGM.
* 37 MSGID: LBL0600 SEVERITY: 10 SEQNBR: 002700
    Message . . . . : No INPUT fields found for format DETACH.
* 37 MSGID: LBL0600 SEVERITY: 10 SEQNBR: 002700
    Message . . . . : No OUTPUT fields found for format DETACH.
* 84 MSGID: LBL0600 SEVERITY: 10 SEQNBR: 003200
    Message . . . . : No INPUT fields found for format HDG.
                                MESSAGE SUMMARY
TOTAL      INFO(0-4)    WARNING(5-19)    ERROR(20-29)    SEVERE(30-39)    TERMINAL
      5             1             4             0             00
                                * * * * * E N D O F C O B O L M E S S A G E S * * * * *
160 source records read
153 copy records read
3 copy members processed
0 sequence errors
10 was the highest severity message issued
  LBL0901 00 Program C85ELTGT created in library BSCELIB.
                                * * * * * E N D O F C O M P I L A T I O N * * * * *

```

Figure E-9 (Part 7 of 7). COBOL/400 Source Program for the Remote System

RPG/400 Source Program for Local System

The following describes the objects on the local system needed to run the RPG/400 BSCEL program.

Configuration

The following configuration commands are used to create the bisynchronous line, controller, and device descriptions used by the local system:

```
CRTLINBSC LIND(BSCELSRC) RSRNAME(LIN011) ONLINE(*NO)
```

```
CRTCTLBSC CTLD(BSCELSRC) ONLINE(*NO) LINE(BSCELSRC)
```

```
CRTDEVBSC DEVD(BSCELSRC) LOCADR(00) RMTLOCNAME(TARGET)
  ONLINE(*NO) CTL(BSCELSRC) CTNWIN(*PRI)
```

Program Files

The following files are used by the local system:

BSCELICF The ICF file used to send and receive records from the remote system. This file was created by using the following command:

```
CR TICFF FILE(BSCELIB/BSCELICF) SRCFILE(BSCELIB/QDSSRC)
  SRCMBR(BSCELICF) TEXT('ICF FILE FOR BSCEL')
```

The DDS used by this file is shown in Figure E-10.


```

                                Data Description Source
SEQNBR *...+...1...+...2...+...3...+...4...+...5...+...6...+...7.
100     A*****
200     A*
300     A*                ICF FILE
400     A*                USED IN BSCSEL BATCH DATA TRANSFER PROGRAM.
500     A*
600     A*****
700     A*
800     A*  FILE LEVEL INDICATORS:
900     A*
1000    A                INDARA
1100    A*
1200    A                RCVENDGRP(40 'RCVD ENDGRP')
1300    A*
1400    A                RCVDETACH(35 'RCVD DETACH')
1500    A*
1600    A*****
1700    A*                BSCSEL RECORD FORMATS
1800    A*****
1900    A*
2000    A                REF(DBFILE)
2100    A                R RCVDATA
2200    A                CUSNUM      R
2300    A                CUSNAM      R
2400    A                ADDR        R
2500    A                CITY        R
2600    A                STATE       R
2700    A                ZIP         R
2800    A                CRDLMT      R
2900    A                CRDAMT      R
3000    A                R SNDDATA
3100    A                CUSNUM      R
3200    A                CUSNAM      R
3300    A                ADDR        R
3400    A                CITY        R
3500    A                STATE       R
3600    A                ZIP         R
3700    A                CRDLMT      R
3800    A                CRDAMT      R
3900    A                R EVOKPGM
4000    A                SECURITY(2 &PASS 3 &USERID)
4100    A                EVOKE(&LIB/&PGMID)
4200    A                PASS        4A
4300    A                USERID      8A
4400    A                LIB         8A
4500    A                PGMID       8A
4600    A                R DETACH      DETACH
                                * * * * *
                                E N D   O F   S O U R C E   * * * * *

```

Figure E-10. DDS for the BSCSELICF File Used by the Local System (RPG/400)

The command needed to define the program device entry is:

```
ADDICFDEVE FILE(BSCSELIB/BSCSELICF) PGMDEV(SOURCE) RMTLOCNAME(TARGET)
```

An OVRICFDEVE command could also be used, with the same parameters.

DBFILE The database file whose data is sent to the remote system.

The DDS for this file is shown in Figure E-11 on page E-28.

```

                                Data Description Source
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 .
100    A*****
200    A*
300    A*                * * * DBFILE * * *
400    A*  DDS FOR DATABASE FILE USED IN BSCCL DATA TRANSFER PROGRAMS. *
500    A*
600    A*
700    A*****
800    A*
900    A          R DBFMT
1000   A          CUSNUM          7A          COLHDG('Customer' 'Number')
1100   A          CUSNAM          25A         COLHDG('Customer' 'Name')
1200   A          ADDR            15A         COLHDG('Address')
1300   A          CITY            15A         COLHDG('City')
1400   A          STATE           2A          COLHDG('State')
1500   A          ZIP              5S 0       COLHDG('Zip')
1600   A          CRDLMT           5S 0       COLHDG('Credit' 'Limit')
1700   A          CRDAMT           5S 0       COLHDG('Credit' 'Amount')
1800   A          K CUSNUM
                                * * * *  E N D  O F  S O U R C E  * * * *

```

Figure E-11. DDS for the Database File Used in BSCCL Data Transfer on the Local System (RPG/400)

BSCELPRT The printer file used to format output to a printer.

The DDS for this file is shown in Figure E-12.

SEQNBR	*...+...1...+...2...+...3...+...4...+...5...+...6...+...7.	Data Description Source	
100	A		REF(DBFILE)
200	A	R HDG	SKIPB(5)
300	A		5DATE EDTCDE(Y)
400	A		20TIME
500	A		60'DATABASE RECORDS RECEIVED'
600	A		120'PAGE'
700	A		+1PAGNBR EDTCDE(Z)
800	A		5'CUSTOMER' SPACEB(2)
900	A		20'CUSTOMER'
1000	A		110'CREDIT'
1100	A		120'CREDIT'
1200	A		5'NUMBER' SPACEB(1)
1300	A		20'NAME'
1400	A		50'ADDRESS'
1500	A		70'CITY'
1600	A		90'STATE'
1700	A		100'ZIP'
1800	A		110'LIMIT'
1900	A		120'AMOUNT'
2000	A	R DTL	SPACEB(1)
2100	A	CUSNUM R	5
2200	A	CUSNAM R	20
2300	A	ADDR R	50
2400	A	CITY R	70
2500	A	STATE R	92
2600	A	ZIP R	100
2700	A	CRDLMT R	110EDTCDE(J)
2800	A	CRDAMT R	120EDTCDE(J)
2900	A	R ERROR	SPACEB(3)
3000	A		5'PROGRAM TERMINATED ABNORMALLY'
3100	A		5'PROGRAM DEVICE:' SPACEB(2)
3200	A	PGMDEV 10	+1
3300	A		5'RECORD FORMAT:' SPACEB(2)
3400	A	FMTNM 8	+1
3500	A		5'MAJOR CODE:' SPACEB(2)
3600	A	MAJCOD 2	+1
3700	A		5'MINOR CODE:' SPACEB(2)
3800	A	MINCOD 2	+1
		*****	END OF SOURCE *****

Figure E-12. DDS for the Printer File on the Local System (RPG/400)

Program Explanation

The following describes the RPG III program on the local system, which is shown in Figure E-13 on page E-32.

- 1** The three files used in this program are specified in the file description section. BSCELICF is the ICF file used to send and receive records from the remote system. The continuation lines for this file define the following:
 - KINFDS** Specifies that the data structure named FEEDBK be used for error handling.
 - KINFSR** Specifies the subroutine named FAIL be used when an error exception condition occurs.
 - KNUM** Specifies the maximum number of program devices to be acquired.
 - KID** Specifies that the program device name will be specified in field PGMDEV.

DBFILE is the name of the database file that contains the data which will be sent to the remote system.

BSCELPRT is the name of the printer file that will format output received from the remote system to a printer device.

- 2** FEEDBK is the name of the file information data structure used with the BSCELICF file. It contains the following information after every input or output operation to the file:

- Record format name (FMTNM)
- Program device name (PGMDEV)
- Major/minor return code (MAJCOD,MINCOD)

- 3** The program device named SOURCE is acquired by the program. This program device was previously added to the BSCELICF file with the ADDICFDEVE command.
- 4** The next routine builds the evoke request and issues the write operation to evoke (start) the program on the remote system. The password is set as the literal BSCE in this program. This is the password for the user ID (profile name on another AS/400 system) BSCEL on the remote system.

Note: With BSCEL, the maximum length of the password is 4 bytes, the maximum length of the user ID is 8 bytes, the maximum length of the library is 8 bytes, and the maximum length of the program name is 8 bytes.

When the program start request is received on the remote AS/400 system, the profile named BSCEL is verified with password BSCE, then the library BSCELIB is searched for program RPGELTGT. The profile BSCEL must specify a job description which includes the BSCELIB library in its library list.

- 5** The next routine reads records from the database file using format DBFMT and immediately writes the data to the remote system, until the end-of-file condition (indicator 99) is met. The condition N99 is used with the write statement to avoid sending a blank record when end-of-file is reached on the database file.
- 6** This routine begins by writing the printer headings. The read operation to the ICF file causes the previous sending function to end. Any data left in buffers is sent followed by an end-of-transmission character. The program on the remote system must be prepared to send now. The read operation completes when data is available from the remote system. A *print then read* loop follows, with detail data being printed with a check for page overflow. If indicator 66

comes on while printing, the heading lines are printed again. The loop is ended when indicator 40 comes on, meaning the end group (end-of-transmission) has been reached.

- 7** This part of the program does the end-of-job processing. First, a write to the ICF file with a DETACH function causes the session with the remote system to end. Since no additional processing is needed in this program, the LR indicator is set on and all files are closed implicitly. The program then ends.
- 8** This subroutine (FAIL) is automatically called when an exception occurs during run time. A write operation to the printer file is used to record information regarding the error condition in the printer output file. The *CANCL option on the ENDSR operation causes the processing of the program to end and all files to close.

SEQUENCE

NUMBER *...1...+...2...+...3...+...4...+...5...+...6...+...7...*

Source Listing

```

H
1 100 FBSCCELICFCF E          WORKSTN
    200 F                      KINFDS FEEDBK
    300 F                      KINFSR FAIL
    400 F                      KNUM      1
    500 F                      KID       PGMDEV
        RECORD FORMAT(S):  LIBRARY BSCELIB FILE BSCELICF.
        EXTERNAL FORMAT RCVDATA RPG NAME RCVDATA
        EXTERNAL FORMAT SNDDATA RPG NAME SNDDATA
        EXTERNAL FORMAT EVOKPGM RPG NAME EVOKPGM
        EXTERNAL FORMAT DETACH RPG NAME DETACH
600 FDBFILE IF E          K          DISK
        RECORD FORMAT(S):  LIBRARY BSCELIB FILE DBFILE.
        EXTERNAL FORMAT DBFMT RPG NAME DBFMT
700 FBSCELPRTO E          66        PRINTER
        RECORD FORMAT(S):  LIBRARY BSCELIB FILE BSCELPRT.
        EXTERNAL FORMAT HDG RPG NAME HDG
        EXTERNAL FORMAT DTL RPG NAME DTL
        EXTERNAL FORMAT ERROR RPG NAME ERROR
A000000 INPUT FIELDS FOR RECORD RCVDATA FILE BSCELICF FORMAT RCVDATA.
A000001          1  7 CUSNUM
A000002          8 32 CUSNAM
A000003         33 47 ADDR
A000004         48 62 CITY
A000005         63 64 STATE
A000006         65 690ZIP
A000007         70 740CRDLMT
A000008         75 790CRDAMT
B000000 INPUT FIELDS FOR RECORD SNDDATA FILE BSCELICF FORMAT SNDDATA.
B000001          1  7 CUSNUM
B000002          8 32 CUSNAM
B000003         33 47 ADDR
B000004         48 62 CITY
B000005         63 64 STATE
B000006         65 690ZIP
B000007         70 740CRDLMT
B000008         75 790CRDAMT
C000000 INPUT FIELDS FOR RECORD EVOKPGM FILE BSCELICF FORMAT EVOKPGM.
C000001          1  4 PASS
C000002          5 12 USERID
C000003         13 20 LIB
C000004         21 28 PGMID
D000000 INPUT FIELDS FOR RECORD DETACH FILE BSCELICF FORMAT DETACH.
E000000 INPUT FIELDS FOR RECORD DBFMT FILE DBFILE FORMAT DBFMT.
E000001          1  7 CUSNUM
E000002          8 32 CUSNAM
E000003         33 47 ADDR

```

Figure E-13 (Part 1 of 5). RPG/400 Source Program for the Local System

SEQUENCE

```

NUMBER      *...1....+....2....+....3....+....4....+....5....+....6....+....7...*
E000004          48 62 CITY
E000005          63 64 STATE
E000006          65 690ZIP
E000007          70 740CRDLMT
E000008          75 790CRDAMT
2 800  IFEDBK      DS
   900  I          38 45 FMTNM
 1000  I          273 282 PGMDEV
 1100  I          401 402 MAJCOD
 1200  I          403 404 MINCOD
 1300  C*****
 1400  C* START PROGRAM *
 1500  C* *
 1600  C* FILES ARE IMPLICITLY OPENED, AND THE BSCCL PROGRAM DEVICE *
 1700  C* IS ACQUIRED. *
 1800  C* *
 1900  C*****
3 2000  C          'SOURCE' ACQ BSCCLICF
 2100  C          MOVE'SOURCE' PGMDEV 10
 2200  C*****
 2300  C* *
 2400  C* THE FOLLOWING ROUTINE EVOKES THE PROGRAM 'RPGELTGT' ON THE *
 2500  C* TARGET SYSTEM. *
 2600  C* *
 2700  C*****
4 2800  C          MOVE'BSCE' PASS
 2900  C          MOVE'BSCEL' USERID
 3000  C          MOVE'BSCELIB' LIB
 3100  C          MOVE'RPGEL' PGMID
 3200  C          MOVE 'TGT' PGMID
 3300  C          WRITEEVOKPGM
 3400  C*****
 3500  C* *
 3600  C* THE FOLLOWING ROUTINE READS RECORDS FROM THE DATABASE FILE *
 3700  C* 'DBFILE'. WHEN 'DBFILE' REACHES END OF FILE, INDICATOR 99 *
 3800  C* IS TURNED ON. CONDITION N99 IS USED ON THE WRITE TO AVOID *
 3900  C* SENDING A BLANK RECORD. *
 4000  C* *
 4100  C*****
5 4200  C          *IN99 DOUEQ'1' B001
 4300  C          READ DBFMT 99
 4400  C N99      WRITESNDDATA
 4500  C          END E001
 4600  C*****
 4700  C* *
 4800  C* THE READ RCVDATA IN THE NEXT SECTION OF THE PROGRAM WILL *
 4900  C* CAUSE THE END-OF-TRANSMISSION TO BE SENT BEFORE RECEIVING *
 5000  C* THE FIRST DATA RECORD. RECORDS WILL BE RECEIVED FROM THE *
 5100  C* TARGET SYSTEM UNTIL AN END-OF-TRANSMISSION IS RECEIVED. THIS *
 5200  C* IS DEFINED AS A RCVENDGRP, SO INDICATOR 40 WILL BE SET ON. *
 5300  C* AS RECORDS ARE RECEIVED THEY ARE PRINTED IN THE PRINT FILE *
 5400  C* 'BSCCLPRT'. INDICATOR 66 SIGNALS PAGE OVERFLOW. *

```

Figure E-13 (Part 2 of 5). RPG/400 Source Program for the Local System

SEQUENCE

```

NUMBER *...1....+...2....+...3....+...4....+...5....+...6....+...7...*
5500 C*
5600 C*****
6 5700 C WRITEHDG
5800 C READ RCVDATA 98
5900 C *IN40 DOWEQ '0' B001
6000 C WRITEDTL
6100 C *IN66 IFEQ '1' B002
6200 C WRITEHDG
6300 C MOVE '0' *IN66
6400 C END E002
6500 C READ RCVDATA 98
6600 C END E001
6700 C*****
6800 C*
6900 C* THE FOLLOWING ROUTINE WRITES A DETACH TO THE TARGET SYSTEM. *
7000 C* WHEN THE TARGET SYSTEM RECEIVES THE DETACH, THE CONVERSATION *
7100 C* BETWEEN THE SOURCE AND TARGET SYSTEMS WILL BE ENDED. *
7200 C* SINCE NO ADDITIONAL PROCESSING IS NEEDED IN THIS PROGRAM, THE *
7300 C* LR INDICATOR IS SET ON AND THE PROGRAM IS ENDED. *
7400 C*
7500 C*****
7 7600 C WRITEDETACH
7700 C MOVE '1' *INLR
7800 C*****
7900 C*
8000 C*FAIL SUBROUTINE
8100 C*
8200 C*****
8 8300 C FAIL BEGSR
8400 C WRITEERROR
8500 C ENDSR '*CANCL'
F000000 OUTPUT FIELDS FOR RECORD SNDDATA FILE BSCCELICF FORMAT SNDDATA.
F000001 CUSNUM 7 CHAR 7
F000002 CUSNAM 32 CHAR 25
F000003 ADDR 47 CHAR 15
F000004 CITY 62 CHAR 15
F000005 STATE 64 CHAR 2
F000006 ZIP 69 ZONE 5,0
F000007 CRDLMT 74 ZONE 5,0
F000008 CRDAMT 79 ZONE 5,0
G000000 OUTPUT FIELDS FOR RECORD EVOKPGM FILE BSCCELICF FORMAT EVOKPGM.
G000001 PASS 4 CHAR 4
G000002 USERID 12 CHAR 8
G000003 LIB 20 CHAR 8
G000004 PGMID 28 CHAR 8
H000000 OUTPUT FIELDS FOR RECORD DETACH FILE BSCCELICF FORMAT DETACH.
I000000 OUTPUT FIELDS FOR RECORD HDG FILE BSCCLPRT FORMAT HDG.
J000000 OUTPUT FIELDS FOR RECORD DTL FILE BSCCLPRT FORMAT DTL.
J000001 CUSNUM 7 CHAR 7
J000002 CUSNAM 32 CHAR 25
J000003 ADDR 47 CHAR 15
J000004 CITY 62 CHAR 15
J000005 STATE 64 CHAR 2
J000006 ZIP 69 ZONE 5,0

```

Figure E-13 (Part 3 of 5). RPG/400 Source Program for the Local System


```

5728RG1 R01M00 880819          IBM AS/400 RPG/400
SEQUENCE
NUMBER *...1....+....2....+....3....+....4....+....5....+....6....+....7....*
J000007          CRDLMT   74  ZONE  5,0
J000008          CRDAMT   79  ZONE  5,0
K000000  OUTPUT  FIELDS FOR RECORD ERROR FILE BSCELPRT FORMAT ERROR.
K000001          PGMDEV   10  CHAR  10
K000002          FMTNM    18  CHAR   8
K000003          MAJCOD   20  CHAR   2
K000004          MINCOD   22  CHAR   2
* * * * *  E N D  O F  S O U R C E  * * * * *

```

```

Additional Diagnostic Messages
* 7089      100  RPG PROVIDES SEPARATE INDICATOR AREA FOR FILE BSCELICF.
* 7086      600  RPG PROVIDES BLOCK OR UNBLOCK SUPPORT FOR FILE DBFILE.
5728RG1 R01M00 880819          IBM AS/400 RPG/400

```

```

Key Field Information
          PHYSICAL  LOGICAL
FILE/RCD FIELD      FIELD      ATTRIBUTES
02 DBFILE
    DBFMT
          CUSNUM          CHAR  7

```

```

5728RG1 R01M00 880819          IBM AS/400 RPG/400
Cross Reference

```

File and Record References:

FILE/RCD	DEV/RCD	REFERENCES (D=DEFINED)
01 BSCELICF	WORKSTN	100D 2000
	DETACH	100D D000000 7600 H000000
	EVOKPGM	100D C000000 3300 G000000
	RCVDATA	100D A000000 5800 6500
	SNDDATA	100D B000000 4400 F000000
03 BSCELPRT	PRINTER	700D
	DTL	700D 6000 J000000
	ERROR	700D 8400 K000000
	HDG	700D 5700 6200 I000000
02 DBFILE	DISK	600D
	DBFMT	600D E000000 4300

Field References:

FIELD	ATTR	REFERENCES (M=MODIFIED D=DEFINED)
*INLR	A(1)	7700M
*IN40	A(1)	5900
*IN66	A(1)	6100 6300M
*IN99	A(1)	4200
ADDR	A(15)	A000003D B000003D E000003D F000003D J000003D
CITY	A(15)	A000004D B000004D E000004D F000004D J000004D
CRDAMT	P(5,0)	A000008D B000008D E000008D F000008D J000008D
CRDLMT	P(5,0)	A000007D B000007D E000007D F000007D J000007D
CUSNAM	A(25)	A000002D B000002D E000002D F000002D J000002D
CUSNUM	A(7)	A000001D B000001D E000001D F000001D J000001D
FAIL	BEGSR	100 8300D
FEEDBK	DS(404)	100 800D
FMTNM	A(8)	900D K000002D
LIB	A(8)	C000003D 3000M G000003D
MAJCOD	A(2)	1100D K000003D
MINCOD	A(2)	1200D K000004D
PASS	A(4)	C000001D 2800M G000001D
PGMDEV	A(10)	1000D 2100D K000001D
PGMID	A(8)	C000004D 3100M 3200M G000004D

Figure E-13 (Part 4 of 5). RPG/400 Source Program for the Local System

```

5728RG1 R01M00 880819          IBM AS/400 RPG/400
STATE      A(2)    A000005D B000005D E000005D F000005D J000005D
USERID     A(8)    C000002D 2900M G000002D
ZIP        P(5,0) A000006D B000006D E000006D F000006D J000006D
'*CANCL'   LITERAL 8500
'BSCE'     LITERAL 2800
'BSCEL'    LITERAL 2900      3100
'BSCELIB'  LITERAL 3000
'SOURCE'   LITERAL 2000      2100
'TGT'      LITERAL 3200
'0'        LITERAL 5900      6300
'1'        LITERAL 4200      6100      7700

```

Indicator References:

```

INDICATOR REFERENCES (M=MODIFIED D=DEFINED)
*IN          4200      5900      6100      6300M
LR           7700M
* 7031 35
40           5900
66           700D      6100      6300M
* 7031 88
* 7031 98     5800M     6500M
99           4200      4300M     4400

```

***** END OF CROSS REFERENCE *****

```

5728RG1 R01M00 880819          IBM AS/400 RPG/400

```

Message Summary

```

* QRG7031 Severity: 00 Number: 3
Message . . . . : The Name or indicator is not referenced.
* QRG7086 Severity: 00 Number: 1
Message . . . . : The RPG handles blocking function for file.
INFDS contents updated only when blocks of data transferred.
* QRG7089 Severity: 00 Number: 1
Message . . . . : The RPG provides Separate-Indicator area for
file.

```

***** END OF MESSAGE SUMMARY *****

```

5728RG1 R01M00 880819          IBM AS/400 RPG/400

```

Final Summary

```

Message Count: (by Severity Number)
TOTAL  00  10  20  30  40  50
       5   5   0   0   0   0

```

Program Source Totals:

```

Records . . . . . : 85
Specifications . . . . . : 39
Table Records . . . . . : 0
Comments . . . . . : 46

```

PRM has been called.

Program RPGELSRC is placed in library BSCELIB. 00 highest Error-Severity-Code.

***** END OF COMPILATION *****

Figure E-13 (Part 5 of 5). RPG/400 Source Program for the Local System

RPG/400 Source Program for Remote System

The following describes the objects on the remote system needed to run the RPG/400 BSCEL target program.

Configuration

The following configuration commands are used to create the bisynchronous line, controller, and device descriptions used by the remote system:

```
CRTLINBSC LIND(BSCELTGT) RSRNAME(LIN021) ONLINE(*NO)
```

```
CRTCTLBSC CTLD(BSCELTGT) ONLINE(*NO) LINE(BSCELTGT)
```

```
CRTDEVBSC DEVD(BSCELTGT) LOCADR(00) RMTLOCNAME(SOURCE) ONLINE(*NO)  
CTL(BSCELTGT) CTNWIN(*SEC)
```

Program Files

The following files are used by the remote system:

BSCELICF The ICF file used to receive and then send records to the source program. This file is created by using the following command:

```
CRTICFF FILE(BSCELIB/BSCELICF) SRCFILE(BSCELIB/QDSSRC)  
SRCMBR(BSCELICF) TEXT('ICF FILE FOR BSCEL')
```

The DDS used by this file is shown in Figure E-14 on page E-38.

```

                                Data Description Source
SEQNBR *...+....1....+....2....+....3....+....4....+....5....+....6....+....7.
 100    A*****
 200    A*
 300    A*                                ICF FILE
 400    A*                                USED IN BSCCEL BATCH DATA TRANSFER PROGRAM.
 500    A*
 600    A*****
 700    A*
 800    A*  FILE LEVEL INDICATORS:
 900    A*
1000    A                                INDARA
1100    A*
1200    A                                RCVENDGRP(40 'RCVD ENDGRP')
1300    A*
1400    A                                RCVDETACH(35 'RCVD DETACH')
1500    A*
1600    A*****
1700    A*                                BSCCEL RECORD FORMATS
1800    A*****
1900    A*
2000    A                                REF(DBFILE)
2100    A      R RCVDATA
2200    A      CUSNUM      R
2300    A      CUSNAM      R
2400    A      ADDR        R
2500    A      CITY        R
2600    A      STATE       R
2700    A      ZIP         R
2800    A      CRDLMT      R
2900    A      CRDAMT      R
3000    A      R SNDDATA
3100    A      CUSNUM      R
3200    A      CUSNAM      R
3300    A      ADDR        R
3400    A      CITY        R
3500    A      STATE       R
3600    A      ZIP         R
3700    A      CRDLMT      R
3800    A      CRDAMT      R
3900    A      R EVOKPGM
4000    A                                SECURITY(2 &PASS 3 &USERID)
4100    A                                EVOKE(&LIB/&PGMID)
4200    A      PASS         4A
4300    A      USERID      8A
4400    A      LIB         8A
4500    A      PGMID       8A
4600    A      R DETACH      DETACH
          * * * * *  E N D   O F   S O U R C E   * * * * *

```

Figure E-14. DDS for the BSCCELICF Program File Used by the Remote System (RPG/400)

The command needed to define the program device entry is:

```
ADDICFDEVE FILE(BSCSELIB/BSCSELICF) PGMDEV(TARGET) RMTLOCNAME(*REQUESTER)
```

An OVRICFDEVE command could also be used, with the same parameters.

DBFILE The database file whose data is sent to the source program.

The DDS for this file is shown in Figure E-15.

```

                                Data Description Source
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 .
100  A*****
200  A*
300  A*          * * * DBFILE * * *
400  A* DDS FOR DATABASE FILE USED IN BSCSEL DATA TRANSFER PROGRAMS. *
500  A*
600  A*
700  A*****
800  A*
900  A          R DBFMT
1000 A          CUSNUM          7A          COLHDG('Customer' 'Number')
1100 A          CUSNAM          25A         COLHDG('Customer' 'Name')
1200 A          ADDR           15A         COLHDG('Address')
1300 A          CITY           15A         COLHDG('City')
1400 A          STATE          2A          COLHDG('State')
1500 A          ZIP            5S 0        COLHDG('Zip')
1600 A          CRDLMT         5S 0        COLHDG('Credit' 'Limit')
1700 A          CRDAMT         5S 0        COLHDG('Credit' 'Amount')
1800 A          K CUSNUM
                                * * * * E N D O F S O U R C E * * * *
```

Figure E-15. DDS for Database File Used in BSCSEL Data Transfer Programs on the Remote System (RPG/400)

BSCCLPRT The printer file used to format output to a printer.

The DDS for this file is shown in Figure E-16.

SEQNBR	*	1	2	3	4	5	6	7
100	A					REF(DBFILE)		
200	A		R HDG			SKIPB(5)		
300	A					5DATE EDTCDE(Y)		
400	A					20TIME		
500	A					60'DATABASE RECORDS RECEIVED		
600	A					120'PAGE'		
700	A					+1PAGNBR EDTCDE(Z)		
800	A					5'CUSTOMER' SPACEB(2)		
900	A					20'CUSTOMER'		
1000	A					110'CREDIT'		
1100	A					120'CREDIT'		
1200	A					5'NUMBER' SPACEB(1)		
1300	A					20'NAME'		
1400	A					50'ADDRESS'		
1500	A					70'CITY'		
1600	A					90'STATE'		
1700	A					100'ZIP'		
1800	A					110'LIMIT'		
1900	A					120'AMOUNT'		
2000	A		R DTL			SPACEB(1)		
2100	A		CUSNUM	R		5		
2200	A		CUSNAM	R		20		
2300	A		ADDR	R		50		
2400	A		CITY	R		70		
2500	A		STATE	R		92		
2600	A		ZIP	R		100		
2700	A		CRDLMT	R		110EDTCDE(J)		
2800	A		CRDAMT	R		120EDTCDE(J)		
2900	A		R ERROR			SPACEB(3)		
3000	A					5'PROGRAM TERMINATED ABNORMALLY'		
3100	A					5'PROGRAM DEVICE:' SPACEB(2)		
3200	A		PGMDEV		10	+1		
3300	A					5'RECORD FORMAT:' SPACEB(2)		
3400	A		FMTNM		8	+1		
3500	A					5'MAJOR CODE:' SPACEB(2)		
3600	A		MAJCOD		2	+1		
3700	A					5'MINOR CODE:' SPACEB(2)		
3800	A		MINCOD		2	+1		
			*****			END OF SOURCE		*****

Figure E-16. DDS for the Printer File on the Remote System (RPG/400)

Program Explanation

The following describes the RPG III program on the remote system, which is shown in Figure E-17 on page E-42.

- 1** The three files used in this program are specified in the file description section. BSCELICF is the ICF file used to receive then send records to the source program. The continuation lines for this file define the following:
 - KINFDS** Specifies that the data structure named FEEDBK be used for error handling.
 - KINFSR** Specifies the subroutine named FAIL be used when an error exception condition occurs.
 - KNUM** Specifies the maximum number of program devices to be acquired.
 - KID** Specifies that the program device name will be specified in field PGMDEV.

DBFILE is the name of the database file that contains the data which will be sent to the source program.

BSCELPRT is the name of the printer file that will format output received from the source program to a printer device.

- 2** FEEDBK is the name of the file information data structure used with the BSCELICF file. It contains the following information after every input or output operation to the file:

- Record format name (FMTNM)
- Program device name (PGMDEV)
- Major/minor return code (MAJCOD,MINCOD)

- 3** The program device named TARGET is acquired by the program. This program device was previously added to the BSCELICF file with the ADDICFDEVE command.
- 4** This routine begins by writing the printer headings. The program on the source system must be prepared to send now. The read operation completes when data is available. A *print then read* loop follows, with the detail data being printed, and a check for page overflow. If indicator 66 comes on while printing, the heading lines are printed again. The loop is ended when indicator 40 comes on, meaning the end group (end-of-transmission) has been reached.
- 5** The next routine reads records from the database file using format DBFMT and immediately writes the data to the other system, until the end-of-file condition (indicator 99) is met. The condition N99 is used with the write statement to avoid sending a blank record when end-of-file is reached on the database file. The source system must be prepared to receive data.
- 6** This part of the program does the end-of-job processing. First, a read to the ICF file causes the last group of data to be sent along with the end-of-transmission. The condition the read expects to receive is detach. Since no additional processing is needed in this program, the LR indicator is set on and all files are closed implicitly. The program will then end.
- 7** This subroutine (FAIL) is automatically called when an exception occurs during program execution. A write operation to the printer file is used to record information regarding the error condition in the printer output file. The *CANCL option on the ENDSR operation causes the processing of the program to end and all files to close.

SEQUENCE

NUMBER *...1....+....2....+....3....+....4....+....5....+....6....+....7...*

Source Listing

```

H
1 100 FBSCELICFCF E          WORKSTN
200 F                      KINFDS FEEDBK
300 F                      KINFSR FAIL
400 F                      KNUM          1
    RECORD FORMAT(S): LIBRARY BSCSELIB FILE BSCELICF.
    EXTERNAL FORMAT RCVDATA RPG NAME RCVDATA
    EXTERNAL FORMAT SNDDATA RPG NAME SNDDATA
    EXTERNAL FORMAT EVOKPGM RPG NAME EVOKPGM
    EXTERNAL FORMAT DETACH RPG NAME DETACH
500 FDBFILE IF E          K          DISK
    RECORD FORMAT(S): LIBRARY BSCSELIB FILE DBFILE.
    EXTERNAL FORMAT DBFMT RPG NAME DBFMT
600 FBSCELPRTO E          66          PRINTER
    RECORD FORMAT(S): LIBRARY BSCSELIB FILE BSCELPRT.
    EXTERNAL FORMAT HDG RPG NAME HDG
    EXTERNAL FORMAT DTL RPG NAME DTL
    EXTERNAL FORMAT ERROR RPG NAME ERROR
A000000 INPUT FIELDS FOR RECORD RCVDATA FILE BSCELICF FORMAT RCVDATA.
A000001          1  7 CUSNUM
A000002          8 32 CUSNAM
A000003         33 47 ADDR
A000004         48 62 CITY
A000005         63 64 STATE
A000006         65 690ZIP
A000007         70 740CRDLMT
A000008         75 790CRDAMT
B000000 INPUT FIELDS FOR RECORD SNDDATA FILE BSCELICF FORMAT SNDDATA.
B000001          1  7 CUSNUM
B000002          8 32 CUSNAM
B000003         33 47 ADDR
B000004         48 62 CITY
B000005         63 64 STATE
B000006         65 690ZIP
B000007         70 740CRDLMT
B000008         75 790CRDAMT
C000000 INPUT FIELDS FOR RECORD EVOKPGM FILE BSCELICF FORMAT EVOKPGM.
C000001          1  4 PASS
C000002          5 12 USERID
C000003         13 20 LIB
C000004         21 28 PGMID
D000000 INPUT FIELDS FOR RECORD DETACH FILE BSCELICF FORMAT DETACH.
E000000 INPUT FIELDS FOR RECORD DBFMT FILE DBFILE FORMAT DBFMT.
E000001          1  7 CUSNUM
E000002          8 32 CUSNAM
E000003         33 47 ADDR
E000004         48 62 CITY

```

Figure E-17 (Part 1 of 5). RPG/400 Source Program for the Remote System

SEQUENCE

```

NUMBER *...1...+...2...+...3...+...4...+...5...+...6...+...7...*
E000005          63 64 STATE
E000006          65 690ZIP
E000007          70 740CRDLMT
E000008          75 790CRDAMT
2 700 IFEEDBK      DS
800 I              38 45 FMTNM
900 I              273 282 PGMDEV
1000 I             401 402 MAJCOD
1100 I             403 404 MINCOD
1200 C*****
1300 C* START PROGRAM. *
1400 C* ACQUIRE PROGRAM DEVICE. *
3 1500 C          'TARGET' ACQ BSCSELICF *
1600 C*****
1700 C* *
1800 C* THE FOLLOWING ROUTINE RECEIVES RECORDS FROM THE SOURCE SYSTEM *
1900 C* UNTIL INDICATOR 40 IS TURNED ON. INDICATOR 40 SIGNALS THAT A *
2000 C* 'RCVENDGRP' HAS BEEN RECEIVED, AND THE SOURCE PROGRAM HAS *
2100 C* STOPPED SENDING RECORDS. AS RECORDS ARE RECEIVED THEY ARE *
2200 C* PRINTED IN THE PRINT FILE 'BSCSELPR'. INDICATOR 66 SIGNALS *
2300 C* PAGE OVERFLOW. *
2400 C* *
2500 C*****
4 2600 C          WRITEHDG
2700 C          READ RCVDATA          99
2800 C          *IN40  DOWEQ'0'          B001
2900 C          WRITEDTL
3000 C          *IN66  IFEQ '1'          B002
3100 C          WRITEHDG
3200 C          MOVE '0'          *IN66
3300 C          END          E002
3400 C          READ RCVDATA          99
3500 C          END          E001
3600 C*****
3700 C* *
3800 C* THE FOLLOWING ROUTINE READS RECORDS FROM THE DATABASE FILE *
3900 C* 'DBFILE'. WHEN 'DBFILE' REACHES 'END OF FILE' INDICATOR 99 *
4000 C* IS TURNED ON. CONDITION N99 IS USED ON THE WRITE TO AVOID *
4100 C* SENDING A BLANK RECORD. *
4200 C* *
4300 C*****
5 4400 C          *IN99  DOUEQ'1'          B001
4500 C          READ DBFMT          99
4600 C N99          WRITESNDDATA
4700 C          END          E001
4800 C*****
4900 C* *
5000 C* THE FOLLOWING READ CAUSES THE SENDING OF END-OF-TRANSMISSION *
5100 C* TO THE SOURCE SYSTEM, AND RECEIVES THE 'DETACH' FROM THE *
5200 C* SOURCE SYSTEM. AFTER DETACH IS RECEIVED, NO OTHER PROCESSING *
5300 C* IS NEEDED SO THE LR INDICATOR IS SET ON AND THE PROGRAM ENDS. *
5400 C* *
5500 C*****

```

Figure E-17 (Part 2 of 5). RPG/400 Source Program for the Remote System

```

SEQUENCE
NUMBER *...1...+...2...+...3...+...4...+...5...+...6...+...7...*
6 5600 C READ RCVDATA 99
5700 C MOVE '1' *INLR
5800 C*****
5900 C* *
6000 C* FAIL SUBROUTINE *
6100 C* *
6200 C*****'*****
7 6300 C FAIL BEGSR
6400 C WRITEERROR
6500 C ENDSR'*CANCL'
F000000 OUTPUT FIELDS FOR RECORD SNDDATA FILE BSCSELICF FORMAT SNDDATA.
F000001 CUSNUM 7 CHAR 7
F000002 CUSNAM 32 CHAR 25
F000003 ADDR 47 CHAR 15
F000004 CITY 62 CHAR 15
F000005 STATE 64 CHAR 2
F000006 ZIP 69 ZONE 5,0
F000007 CRDLMT 74 ZONE 5,0
F000008 CRDAMT 79 ZONE 5,0
G000000 OUTPUT FIELDS FOR RECORD HDG FILE BSCSELPRF FORMAT HDG.
H000000 OUTPUT FIELDS FOR RECORD DTL FILE BSCSELPRF FORMAT DTL.
H000001 CUSNUM 7 CHAR 7
H000002 CUSNAM 32 CHAR 25
H000003 ADDR 47 CHAR 15
H000004 CITY 62 CHAR 15
H000005 STATE 64 CHAR 2
H000006 ZIP 69 ZONE 5,0
H000007 CRDLMT 74 ZONE 5,0
H000008 CRDAMT 79 ZONE 5,0
I000000 OUTPUT FIELDS FOR RECORD ERROR FILE BSCSELPRF FORMAT ERROR.
I000001 PGMDEV 10 CHAR 10
I000002 FMTNM 18 CHAR 8
I000003 MAJCOD 20 CHAR 2
I000004 MINCOD 22 CHAR 2
***** END OF SOURCE *****

```

Additional Diagnostic Messages

- * 7089 100 RPG PROVIDES SEPARATE INDICATOR AREA FOR FILE BSCSELICF.
- * 7086 500 RPG PROVIDES BLOCK OR UNBLOCK SUPPORT FOR FILE DBFILE.

Key Field Information

FILE/RCD	PHYSICAL FIELD	LOGICAL FIELD	ATTRIBUTES
02 DBFMT		CUSNUM	CHAR 7

Figure E-17 (Part 3 of 5). RPG/400 Source Program for the Remote System

5728RG1 R01M00 880819 IBM AS/400 RPG/400 BSCELIB
 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 BSCELICF	WORKSTN	100D	1500			
	DETACH	100D	D000000			
	EVOKPGM	100D	C000000			
	RCVDATA	100D	A000000	2700	3400	5600
	SNDDATA	100D	B000000	4600	F000000	
03 BSCELPRT	PRINTER	600D				
	DTL	600D	2900	H000000		
	ERROR	600D	6400	I000000		
	HDG	600D	2600	3100	G000000	
02 DBFILE	DISK	500D				
	DBFMT	500D	E000000	4500		

Field References:

FIELD	ATTR	REFERENCES (M=MODIFIED D=DEFINED)				
*INLR	A(1)	5700M				
*IN40	A(1)	2800				
*IN66	A(1)	3000	3200M			
*IN99	A(1)	4400				
ADDR	A(15)	A000003D	B000003D	E000003D	F000003D	H000003D
CITY	A(15)	A000004D	B000004D	E000004D	F000004D	H000004D
CRDAMT	P(5,0)	A000008D	B000008D	E000008D	F000008D	H000008D
CRDLMT	P(5,0)	A000007D	B000007D	E000007D	F000007D	H000007D
CUSNAM	A(25)	A000002D	B000002D	E000002D	F000002D	H000002D
CUSNUM	A(7)	A000001D	B000001D	E000001D	F000001D	H000001D
FAIL	BEGSR	100	6300D			
FEEDBK	DS(404)	100	700D			
FMTNM	A(8)	800D	I000002D			
* 7031 LIB	A(8)	C000003D				
	MAJCOD	A(2)	1000D	I000003D		
	MINCOD	A(2)	1100D	I000004D		
* 7031 PASS	A(4)	C000001D				
	PGMDEV	A(10)	900D	I000001D		
* 7031 PGMID	A(8)	C000004D				
	STATE	A(2)	A000005D	B000005D	E000005D	F000005D
* 7031 USERID	A(8)	C000002D				
	ZIP	P(5,0)	A000006D	B000006D	E000006D	F000006D
	'*CANCL'	LITERAL	6500			
	'TARGET'	LITERAL	1500			
	'0'	LITERAL	2800	3200		
	'1'	LITERAL	3000	4400	5700	

5728RG1 R01M00 880819 IBM AS/400 RPG/400 BSCELIB

Indicator References:

INDICATOR	REFERENCES (M=MODIFIED D=DEFINED)				
*IN	2800	3000	3200M	4400	
LR	5700M				
* 7031 35					
	40	2800			
	66	600D	3000	3200M	
	99	2700M	3400M	4400	4500M 4600 5600M

* * * * * E N D O F C R O S S R E F E R E N C E * * * * *

Figure E-17 (Part 4 of 5). RPG/400 Source Program for the Remote System

```

5728RG1 R01M00 880819          IBM AS/400 RPG/400          BSCELIB
                          M e s s a g e   S u m m a r y
* QRG7031 Severity: 00   Number: 5
    Message . . . . : The Name or indicator is not referenced.
* QRG7086 Severity: 00   Number: 1
    Message . . . . : The RPG handles blocking function for file.
                      INFDS contents updated only when blocks of data transferred.
* QRG7089 Severity: 00   Number: 1
    Message . . . . : The RPG provides Separate-Indicator area for
                      file.
      * * * * *   E N D   O F   M E S S A G E   S U M M A R Y   * * * * *
5728RG1 R01M00 880819          IBM AS/400 RPG/400          BSCELIB
                          F i n a l   S u m m a r y
Message Count: (by Severity Number)
      TOTAL    00    10    20    30    40    50
           7     7     0     0     0     0     0
Program Source Totals:
  Records . . . . . : 65
  Specifications . . . . . : 31
  Table Records . . . . . : 0
  Comments . . . . . : 34
PRM has been called.
Program RPGELTGT is placed in library BSCELIB. 00 highest Error-Severity-Code.
      * * * * *   E N D   O F   C O M P I L A T I O N   * * * * *

```

Figure E-17 (Part 5 of 5). RPG/400 Source Program for the Remote System

Glossary

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

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

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

alphabetic character. (1) Any one of the letters A through Z or a through z or one of the characters #, \$, or @. (2) In COBOL, a character that is one of the 26 uppercase letters of the alphabet, or a space. (3) (DDS) (IDDU) Any one of the uppercase letters A through Z or one of the characters

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

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

American National Standard Code for Information Interchange (ASCII). The code developed by American National Standards Institute for information exchange among data processing systems, data communications systems, and associated equipment. The ASCII character set consists of 7-bit control characters and symbolic characters, plus one parity-check bit.

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

ASCII. See *American National Standard Code for Information Interchange (ASCII)*.

asynchronous. (1) Not occurring in a regular or predictable pattern. (2) Without regular time relationship.

asynchronous communications. A method of communications supported by the operating system that allows an exchange of data with a remote device, using either a start-stop line or an X.25 line. Asynchronous communications includes the file transport support and the interactive terminal facility support.

asynchronous processing. A series of operations that are done separately from the job in which they were requested; for example, submitting a batch job from an interactive job at a work station. Contrast with *synchronous processing*.

asynchronous transmission. A method of transmission in which the sending and receiving of data is controlled by control characters instead of by a timing sequence. Contrast with *synchronous transmission*.

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

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

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

BASIC (beginner's all-purpose symbolic instruction code). A programming language with a small list of commands and a simple syntax, primarily designed for numeric applications.

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

binary. A numbering system with a base of two (0 and 1).

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

binary synchronous communications equivalence link (BSC) support. The system support that provides BSC communication with another AS/400 system and many other BSC computers and devices.

block. (1) A group of records that are recorded or processed as a unit. (2) A set of adjacent records stored as a unit on a disk, diskette, or magnetic tape. (3) In data communications, a group of records that are received, processed, or sent as a unit. (4) A sequential group of statements (defined using line commands) that are processed as a unit. (5) In COBOL, a unit of data that is moved into or out of the computer storage.

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

BSC) support. See *binary synchronous communications equivalence link (BSC) support*.

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

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

CL. See *control language (CL)*.

close. The function that ends the connection between a file and a program, and ends the processing. Contrast with *open*.

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

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

column separator. A symbol on each side of a position of a field on a display. This symbol does not occupy a position on the display.

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

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

command line. The blank line on a display where commands, option numbers, or selections can be entered.

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

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

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

communications line. The physical link (such as a wire or a telephone circuit) that connects one or more work stations to a communications controller unit, or con-

nects one controller to another. Contrast with *data link protocol*.

communications type. A method for application programs to communicate on a local AS/400 system, or between a local AS/400 system and a remote system using the intersystem communications function (ICF). Examples of these communications methods include (a) Systems Network Architecture (SNA) such as advanced program-to-program communications (APPC) and SNA upline facility (SNUF), (b) binary synchronous communications (BSC), and (c) asynchronous communications.

compression. A function that removes duplicate characters from the data being processed and replaces the duplicate characters with control characters. Compression reduces the amount of storage space required for the data. See also *decompression*.

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

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

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

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

control language (CL) program. A program that is created from source statements consisting entirely of control language commands.

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

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

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

conversation. In interactive communications, the communication between the application program and a spe-

cific item (usually another application program) at the remote system.

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

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

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

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

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

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

data terminal equipment (DTE). That part of a data link that sends data, receives data, and provides the data communications control function according to protocols.

data-link-escape (DLE) character. A BSC transmission control character used to indicate that the next character is a control character, not a data character.

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

database file. An object that contains descriptions of how input data is to be presented to a program from internal storage and how output data is to be presented

to internal storage from a program. See also *physical file* and *logical file*.

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

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

decompression. A function that exchanges control characters for actual data. See also *compression*.

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

DEVD. See *device description*.

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

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

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

device name. The symbolic name of an individual device.

display file. A device file created by the user to support a display station.

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

Displaywriter. An independent work station that uses the Electronic Document Distribution licensed program to communicate with other office products.

DLE. See *data-link-escape (DLE) character*.

do group. (1) A set of commands in a control language program defined by a DO command and an ENDDO command that is conditionally processed as a group. (2) In RPG/400, a group of calculations done one or more times based on the results of comparing factor 1 and factor 2 of certain calculation operations (for example, DOUXX). A DO operation and an END operation are the delimiters for a do group.

document library. The system library named QDOC that contains all documents and folders.

DTE. See *data terminal equipment (DTE)*.

EBCDIC. See *extended binary-coded decimal interchange code (EBCDIC)*.

EBCDIC character. Any one of the symbols included in the 8-bit EBCDIC set.

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

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

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

EOT. See *end-of-transmission (EOT) character*.

ETB. See *end-of-transmission-block (ETB) character*.

ETX. See *end-of-text (ETX) character*.

extended binary-coded decimal interchange code (EBCDIC). A coded character set of 256 eight-bit characters.

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

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

file name. (1) The name used by a program to identify a file. See also *label*. (2) (COBOL) A name associated with a file and defined in a file description entry or in a sort-merge file description entry.

fold. To continue data onto the next line. Contrast with *truncate*.

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

general-purpose library. The library shipped with the system that contains IBM-provided objects required for many system functions and user-created objects that are not explicitly placed in a different library when they are created. Named QGPL.

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

hex. See *hexadecimal*.

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

high-level language (HLL). A programming language, such as RPG, BASIC, PL/I, Pascal, COBOL, and C used to write computer programs.

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

HLL. See *high-level language (HLL)*.

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

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

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

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

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

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

intermediate block check. A check that verifies each record, rather than the contents of the total block, when large blocks of data are received.

intermediate-text-block (ITB) character. The BSC transmission control character used to divide a block of text into smaller groups of text for an intermediate block check. See *intermediate block check*.

interrecord-separator character (IRS). A transmission control character that is used to separate records within a block of data.

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

IPL. See *initial program load (IPL)*.

IRS. See *interrecord-separator character (IRS)*.

ITB. See *intermediate-text-block (ITB) character*.

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

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

justify. To adjust text so that line endings are even. See *left-justify* and *right-justify*.

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

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

label. (1) The name of a file on a diskette or tape. (2) An identifier of a command or program statement generally used for branching. (3) In RPG/400, a symbolic name that represents a specific location in a program. A label can serve as the destination point for one or more branching operations.

left-justify. To print text with an even left margin. The right margin is printed unevenly.

library. (1) An object on disk that serves as a directory to other objects. A library groups related objects, and allows the user to find objects by name. Compare with *folder* and *document library*. (2) The set of publications for a system.

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

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

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

line description. An object that contains information describing a particular communications line that is attached to the system.

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

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

local location name. The name by which your system is known to other systems in an SNA network. Equiv-

alent to an SNA local logical unit name. Contrast with *remote location name*.

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

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

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

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

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

member. Different sets of data within one file. See also *source member*.

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

mode description. A system object created for advanced-program-to-program communications (APPC) devices that describe the session limits and the characteristics of the session, such as the maximum number of sessions allowed, maximum number of conversations allowed, the pacing value for incoming and outgoing request/response units, and other controlling information for the session.

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

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

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

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

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

null record. In binary synchronous communications, a record that contains no data; only the data link control characters STX ETX.

null value. A parameter position for which no value is specified.

offline. Pertaining to the operation of a functional unit that is not under the continual control of the system. Contrast with *online*.

online. Pertaining to the operation of a functional unit that is under the continual control of the system. Contrast with *offline*.

online information. Information, read on the display screen, that explains displays, messages, and programs.

open. The function that connects a file to a program for processing. Contrast with *close*.

Operating System/400 (OS/400). (Operating System/400 and OS/400 are trademarks of the International Business Machines Corporation.) The operating system used by the AS/400 system.

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

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

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

packed decimal format. Representation of a decimal value in which each byte within a field represents two numeric digits except the far right byte, which contains one digit in bits 0 through 3 and the sign in bits 4 through 7. For all other bytes, bits 0 through 3 represent one digit; bits 4 through 7 represent one digit. For example, the decimal value +123 is represented as 0001 0010 0011 1111. Contrast with *zoned decimal format*.

packed field. A field that contains data in the packed decimal format.

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

parameter list. A list of values that provide a means of associating addressability of data defined in a called program with data in the calling program. It contains parameter names and the order in which they are to be associated in the calling and called program.

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

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

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

positional parameter. A parameter that must appear in a specified location, relative to other parameters.

program name. A user-defined word that identifies a COBOL source program.

prompt. (1) A reminder or a displayed request for information or user action. The user must respond to allow the program to proceed. (2) A list of values or a request for information provided by the system as a reminder of the type of information or action required.

protocol. A set of rules controlling the communication and transfer of data between two or more devices in a communications system.

QGPL. See *general-purpose library*.

read-from-invited-program-devices operation. An input operation that waits for input from any one of the invited program devices for a user-specified time. Contrast with *read-from-one-program-device operation*.

read-from-one-program-device operation. An input operation that will not complete until the specified device has responded with input. Contrast with *read-from-invited-program-devices operation*.

release-program-device operation. An operation that makes a program device not available for input/output operations. Contrast with *acquire-program-device operation*.

remote. Pertaining to a device, system, or file that is connected to another device, system, or file through a communications line. Contrast with *local*.

remote controller. A device or system, attached to a communications line, that controls the operation of one or more remote devices. Contrast with *local controller*.

remote location name. Any other system with which your system can communicate in an SNA network. This corresponds to the remote location name specified in the communications configuration. Contrast with *local location name*.

remote system. Any other system in the network with which your system can communicate.

remote work station. A work station that is connected to the system by data communications. Contrast with *local work station*.

return code. In data communications, a value sent by the system to a program to indicate the results of an operation by that program.

reverse-interrupt (RVI) character. The BSC transmission control character that is sent as a request from the receiving station to the sending station to stop transmitting and begin receiving a message.

right-justify. To print text with an even right margin by adding extra space throughout a line.

RPG. Report Program Generator. A programming language designed for writing application programs for business data processing requirements. The application programs range from report writing and inquiry programs to applications such as payroll, order entry, and production planning.

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

RVI. See *reverse-interrupt (RVI) character*.

SDLC. See *synchronous data link control (SDLC)*.

session. (1) The length of time that starts when a user signs on and ends when the user signs off at a display station. (2) In communications, the logical connection by which a program or device can communicate with a program or device at a remote location. (3) In SNA, a logical connection between two network locations that can be started, tailored to provide various connection protocols, and stopped, as requested. Each session is uniquely identified in a header by a pair of network addresses identifying the origin and destination of any transmissions exchanged during the session. See also *half-session*.

SOH. See *start-of-header (SOH) character*.

source file. (1) A file of programming code that is not compiled into machine language. Contrast with *data file*. (2) A file created by the specification of FILETYPE(*SRC) on the Create command. A source file can contain source statements for such items as high-level language programs and data description specifications.

source member. A member of a database source file that contains source statements such as RPG/400, COBOL, BASIC, PL/I, or DDS statements. See also *member*.

source program. (1) A set of instructions that are written in a programming language and must be translated to machine language before the program can be run. (2) In communications, the program that starts a session with a remote system. Contrast with *target program*.

source system. The system that issues a request to establish communications with another system. (DDM) The system on which an application program issues a request to use a remote file. Contrast with *target system*.

start-of-header (SOH) character. In binary synchronous communications, the transmission control character indicating that the information that follows is a header.

start-of-text (STX) character. In binary synchronous communications, a transmission control character used to begin a logical set of records that will be ended by the end-of-text character or end-of-transmission-block character.

STX. See *start-of-text (STX) character*.

subsystem. An operating environment, defined by a subsystem description, where the system coordinates processing and resources.

subsystem description. A system object that contains information defining the characteristics of an operating environment controlled by the system.

switched line. In data communications, a connection between computers or devices that is established by dialing. Contrast with *nonswitched line*.

synchronous data link control (SDLC). (1) A form of communications line control that uses commands to control the transfer of data over a communications line. (2) A communications discipline conforming to subsets of the Advanced Data Communication Control Procedures (ADCCP) of the American National Standards Institute (ANSI) and High-Level Data Link Control (HDLC) of the International Standards Organization (ISO), for transferring synchronous, code-transparent, serial-by-bit information over a communications line. Transmission exchanges may be duplex or half-duplex over switched or nonswitched lines. The configuration of the connection may be point-to-point, multipoint, or loop. Compare with *binary synchronous communications (BSC)*.

synchronous processing. A series of operations that are done as part of the job in which they were requested; for example, calling a program in an interactive job at a work station. Contrast with *asynchronous processing*.

synchronous transmission. A method of transmission in which the sending and receiving of data is controlled by timing signals. Contrast with *asynchronous transmission*.

system configuration list. A list of devices that are provided with the system.

system security. A system function that restricts the use of files, libraries, folders, and devices to certain users.

target. In advanced program-to-program communications, the program or system to which a request for processing is sent.

target program. In communications, the program that is started on the remote system at the request of the source system. Contrast with *source program*.

target system. In a distributed data management (DDM) network, the system that receives a request from an application program on another system to use one or more files located on the target system. Contrast with *source system*.

temporary-text-delay (TTD) character. The BSC transmission control character that tells the receiving station that there is a temporary delay in sending the data.

transaction. In communications, an exchange between a program on a local system and a program on a remote system that accomplishes a particular action or result. See also *conversation* and *session*.

transmission control characters. In data communications, special characters that are included in a message to control communications over a data link. For example, the sending station and the receiving station use transmission control characters to exchange information; the receiving station uses transmission control characters to indicate errors in data it receives.

transmission services. A switched, nonswitched, or packet-switched communications line provided by a vendor.

transparency. (BSC) See *transparent text mode*.

transparent data. Data that can contain any hexadecimal value.

transparent text mode. In binary synchronous communications, a method of transmission in which only transmission control characters preceded by the DLE control character are processed as transmission control characters.

truncate. (1) To cut off data that cannot be printed or displayed in the line width specified or available. Contrast with *fold*. (2) To cut off data that does not fit in the specified field length in a field definition.

TS. See *transmission services*.

TTD. See *temporary-text-delay (TTD) character*.

turnaround. Pertaining to changing a communications line from being able to send to being able to receive, or from being able to receive to being able to send.

unbind command. A command used to reset the protocols for a session. Contrast with *bind command*.

user ID. See *user identification (user ID)*.

user identification (user ID). (1) The name used to associate the user profile with a user when a user signs on the system. See also *user profile name*. (2) The first part of a two-part network name used in the system distribution directory and in the office applications to uniquely identify a user. The network name is usually the same as the user profile name, but does not need to be. See also *common user identification (common user ID)*.

user password. A unique string of characters that a system user must enter to identify himself to the system, if the system resources are secured.

user profile. An object with a unique name that contains the user's password, the list of special authorities assigned to a user, and the objects the user owns.

user profile name. The name or code that the system associates with a user when he or she signs on the system. Also known as user ID. See also *user identification (user ID)*.

vary off. To make a device, controller, or line unavailable for its normal, intended use.

vary on. To make a device, controller, or line available for its normal, intended use.

work station. A device used to transmit information to or receive information from a computer; for example, a display station or printer.

write operation. An output operation that sends a processed record to an output device or output file.

zoned decimal format. A format for representing numbers in which the digit is contained in bits 4 through 7 and the sign is contained in bits 0 through 3 of the far right byte; bits 0 through 3 of all other bytes contain 1's (hex F). For example, in zoned decimal format, the decimal value of +123 is represented as 1111 0001 1111 0010 1111 0011. Same as unpacked decimal format. Contrast with *packed decimal format*.

Index

A

acquire operation

- description 4-10
- fails when remote format specified 4-11
- identifying session 5-2
- with multipoint lines 4-10
- with point-to-point lines 4-10

Acquire (*ACQ) command 5-11

Add Intersystem Communications Function Program

Device Entry (ADDICFDEVE) command

- compared to configuration parameters 4-8
- specifying parameters 4-2

ADDICFDEVE command

- See Add Intersystem Communications Function Program Device Entry (ADDICFDEVE) command

allow-write function

- description 4-18
- sending end-of-transmission indicator 4-18
- using switched lines 4-18

application flow examples D-1

ASCII character translation 1-4

AS/400 System manuals iii

B

batch communications 1-1

blocking

- with End-of-Transaction (*EOX) command 5-12
- with write operation 4-14

BSC control codes and protocols C-1

BSCSEL

- application flow examples D-1
- commands
 - Acquire (*ACQ) 5-11
 - End-of-Transaction (*EOX) 5-12
 - Release (*REL) 5-12
- communications lines supported 1-2
- configuring 2-1
- data formats supported 1-3
- entering commands 2-1
- example programs E-1
- online message descriptions 5-16
- online messages 5-15
- programming considerations 5-1
- remote systems supported 1-1
- running 3-1
- using
 - multipoint tributary lines 1-2
 - point-to-point lines 1-2

C

cancel-invite function

- description 4-18
- receiving return code 0412 4-18
- with switched lines 4-18

Change BSC Controller Description (CHGCTLBSC)

command 2-1

Change BSC Device Description (CHGDEVBSC)

command 2-1

Change BSC Line Description (CHGLINBSC)

command 2-1

Change Intersystem Communications Function

Program Device Entry (CHGICFDEVE) command

- specifying parameters 4-2

changing

- BSCSEL configuration 2-1
- controller description 2-1
- device description 2-1
- line description 2-1

CHGCTLBSC (Change BSC Controller Description)

command 2-1

CHGDEVBSC (Change BSC Device Description)

command 2-1

CHGICFDEVE command

- See Change Intersystem Communications Function Program Device Entry (CHGICFDEVE) command

CHGLINBSC (Change BSC Line Description)

command 2-1

COBOL/400 source program (local system)

- configuration
 - Create BSC Controller Description (CRTCTLBSC) command E-1
 - Create BSC Device Description (CRTDEVBSC) command E-1
 - Create BSC Line Description (CRTLINBSC) command E-1
- program files
 - BSCSELICF E-2
 - BSCSELPRT E-4
 - DBFILE E-4

COBOL/400 source program (remote system)

- configuration
 - Create BSC Controller Description (CRTCTLBSC) command E-15
 - Create BSC Device Description (CRTDEVBSC) command E-15
 - Create BSC Line Description (CRTLINBSC) command E-15
- program files
 - BSCSELICF E-15
 - BSCSELPRT E-17
 - DBFILE E-16

- command parameters, VRYCFG** 3-1
- command prompt** 2-1
- commands**
 - Call Program (CALL) 5-2
 - Change BSC Controller Description (CHGCTLBSC) 2-1
 - Change BSC Device Description (CHGDEVBSC) 2-1
 - Change BSC Line Description (CHGLINBSC) 2-1
 - Create BSC Controller Description (CRTCTLBSC) 2-1
 - Create BSC Device Description (CRTDEVBSC) 2-1
 - Create BSC Line Description (CRTLINBSC) 2-1
 - Delete Controller Description (DLTCTLD) 2-2
 - Delete Device Description (DLTDEVD) 2-2
 - Delete Line Description (DLTLIND) 2-2
 - for BSCCEL 5-11
 - Vary Configuration (VRYCFG) 3-1
- communications**
 - ending a session 4-20
 - ending a transaction 4-19
 - I/O feedback area 4-22
 - notifying program of errors 4-16
 - receiving data 4-15
 - return code processing 4-22
 - safe indicator 4-22
 - sending data 4-13
 - starting a session 4-10
 - starting a transaction 4-11
 - using response indicators 4-21
- communications descriptions, defining** 2-1
- communications line**
 - contention state 4-18
 - multipoint tributary 1-2
 - point-to-point 1-2
 - sessions supported per line 1-2
 - sharing with other programs 1-2
 - types supported by BSCCEL 1-2
 - using point-to-point with acquire operation 4-10
- comparing configuration and program device entry command parameters** 4-8
- configuration**
 - changing 2-1
 - compared to command parameters 4-8
 - creating 2-1
 - deleting 2-2
 - retrieving status 2-2
 - varying off 3-1
 - varying on 3-1
- configuring BSCCEL** 2-1
- contention state for lines** 4-18
- continuation format**
 - description 5-5
 - entering 5-5
 - example 5-6
 - using with 3740 Data Entry Systems 5-13
- control characters** C-1
- controller description**
 - changing 2-1

- controller description** (*continued*)
 - creating 2-1
 - deleting 2-2
 - retrieving status 2-2
- Create BSC Controller Description (CRTCTLBSC)**
 - command** 2-1, E-1, E-15, E-26, E-37
- Create BSC Device Description (CRTDEVBSC)**
 - command** 2-1, E-1, E-15, E-26, E-37
- Create BSC Line Description (CRTLINBSC)**
 - command** 2-1, E-1, E-15, E-26, E-37
- creating**
 - BSCCEL configuration 2-1
 - controller description 2-1
 - device description 2-1
 - line description 2-1
- CRTCTLBSC (Create BSC Controller Description)**
 - command** 2-1
- CRTDEVBSC (Create BSC Device Description)**
 - command** 2-1
- CRTICFF command** 4-1
- CRTLINBSC (Create BSC Line Description)**
 - command** 2-1

D

- data description specifications (DDS)**
 - description of keywords A-3
- data files** 5-13
- data formats**
 - parameters 5-7
 - types supported by BSCCEL 1-3
- DDS**
 - See data description specifications (DDS)
- Delete Controller Description (DLTCTLD)**
 - command** 2-2
- Delete Device Description (DLTDEVD) command** 2-2
- Delete Line Description (DLTLIND) command** 2-2
- deleting**
 - BSCCEL configuration 2-2
 - controller description 2-2
 - device description 2-2
 - line description 2-2
- detach function**
 - description 4-19
 - ending a transaction 4-19
 - with RMTBSCCEL(*NO) 4-19
 - with RMTBSCCEL(*YES) 4-19
 - with *EXEC program start request 4-19
 - with *EXNC program start request 4-19
- device description**
 - changing 2-1
 - creating 2-1
 - deleting 2-2
 - retrieving status 2-2
- distributed data processing support** 1-1
- DLTCTLD (Delete Controller Description)**
 - command** 2-2

DLTDEVD (Delete Device Description) command 2-2
DLTLIND (Delete Line Description) command 2-2

E

EBCDIC character translation 1-4

end-of-group function

description 4-15
specifying group separator character 4-15

end-of-session function

description 4-20
using after error occurs 4-20

End-of-Transaction (*EOX) command

description 5-12
with blocking 5-12
with data 5-12
without blocking 5-12
without data 5-12

ending

a communications session
with end-of-session function 4-20
with release operation 4-20
a transaction 4-19

evoke function

parameters specified 4-12
sending program start request with 4-11, 5-4
specifying security function with 4-13
specifying target program 4-12, 5-7
starting a transaction 4-11
using more than one 4-11
with RMTBSCEL(*NO) 4-12
with RMTBSCEL(*YES) 4-12

example programs E-1

F

fail function

description 4-17
receiving data 4-17
sending data 4-17

for program start requests, formats 5-4

format select (FMTSLT) parameter

with acquire operation 4-11
with record-identification function 4-16

formats for program start requests 5-4

functions

allow-write 4-18
cancel-invite 4-18
detach 4-19
end-of-group 4-15
end-of-session 4-20
evoke 4-11
fail 4-17
invite 4-16
record identification 4-16
request-to-write 4-17
security 4-13
subdevice selection 4-14

functions (continued)

timer 4-18
variable length data 4-15, 4-16

G

get-attributes operation 4-19

H

high-level language operations A-1

I

ICF file

opening 4-10

index search, description of v

invite function

canceling with cancel-invite function 4-18
description 4-16
with read-from-invited-program-devices
function 4-16

I/O feedback area

general description 4-22

J

job-level changes 4-8

journal (QHST) 5-15

K

keywords for data description specifications A-3

L

language operations

for programming languages A-2
supported by ICF A-1

line description

changing 2-1
creating 2-1
deleting 2-2
retrieving status 2-2
specifying Idle Disconnect Timer parameter 4-18

line protocols C-1

M

manuals

AS/400 System iii

multiple file formats

ending with null record 5-13
programming considerations 5-13

multipoint lines 4-10

N

notifying program of errors 4-16

null records

- control characters 5-12
- for multiple file formats 5-13
- receiving for remote system 5-12
- receiving for response indicator 4-21

O

office systems

- ending session 5-14
- format 5-13
- handling documents 1-2
- indicating end of documents 5-14
- programming considerations 5-13
- receiving documents 5-14

online education, description of v

online information, types of

- help for control language commands v
- help for displays v
- index search v
- online education v
- question-and-answer function v

online messages

- action BSCEL takes 5-15
- description of messages 5-16
- format 5-15
- from *EXEC or *EXEX program start request 5-15

open operation 4-10

opening an ICF file 4-10

operations

- acquire 4-10
- DDS keywords for A-3
- get attributes 4-19
- open 4-10
- read 4-15
- read-from-invited-program-devices 4-16
- release 4-20
- using A-1
- write 4-13

Override Intersystem Communications Function

Program Device Entry (OVRICFDEVE) command

- compared to configuration parameters 4-8
- specifying parameters 4-2

OVRICFDEVE command

- See Override Intersystem Communications Function Program Device Entry (OVRICFDEVE) command

P

parameters

- data format 5-7
- for evoke function 4-12
- positional 5-2
- security 4-13
- user-defined 5-2
- VRYCFG command 3-1

password parameter 4-13

permanent changes 4-8

point-to-point lines 4-10

positional parameters

- sending security information 5-2
- separating from user-defined 5-2

prestart jobs function 5-10

program device entry

- command parameters 4-8
- commands 4-2

program start request

- coding positions 5-2
- considerations 5-1
- continuations
 - description 5-5
 - entering 5-5
 - example 5-6
 - sending 5-5
 - using with 3740 Data Entry Systems 5-13

example

- *EXEC format 5-5
- *EXNX format 5-6

files for 3740 Data Entry Systems 5-13

formats

- data format 5-7
- description 5-4
- example 5-5, 5-6
- sent with evoke operation 5-4
- types for remote system 5-4
- using *EXNC or *EXNX 5-11

from remote system 5-6

from the AS/400 system 5-7

maximum length 5-2, 5-7

parameters

- positional 5-2
- user-defined 5-2

prestart jobs 5-10

separating parameters 5-2

session activity 5-6

specifying to send online messages 5-15

syntax 5-1

program start requests

failed 5-8

programming considerations

BSCEL online messages 5-15

for AS/400 system 5-15

for office systems 5-13

for remote systems 5-11

for 6580 Displaywriters 5-14

introduction 5-1

receiving null records 5-12

reverse-interrupt indicator 5-16

sending BSCEL commands 5-11

Q

Q & A

See question-and-answer function, description of

QHST journal 5-15

question-and-answer function, description of v

R

read operation

description 4-15
using response indicators 4-21

read-from-invited-program-devices operation

description 4-16
when to use 4-16
with invite function 4-16

reason codes 5-8

receive-detach response indicator

description 4-22
for RMTBSCSEL(*YES) 4-22
for *EXEC program start request 4-22

receive-end-of-group response indicator

description 4-21
end-of-text (ETX) indicator 4-21
end-of-transmission (EOT) indicator 4-21
receiving
null record 4-21

receive-turnaround response indicator

description 4-21
receiving end-of-text (ETX) indicator 4-21

receiving data

during a transaction 4-15
from previously invited program device 4-16
from specific program 4-15

record lengths, determining 4-9

record-identification function

description 4-16
with format select parameter 4-16

release operation

description 4-20
when source program starts session 4-20

Release (*REL) command 5-12

remote location name 4-2

remote system

acquiring a program on 4-10
BSCSEL commands 5-11
ending a transaction 4-19
receiving multiple file formats 5-13
receiving null records 5-12
sending data to 4-13
sending multiple file formats 5-13
sending security information to 4-13
starting transaction on 4-11
two types 5-11
types supported by BSCSEL 1-1
using BSCSEL 4-10
using office systems 5-13
using security 4-11

request-to-write function

description 4-17
receiving return code 0000 4-17
sending reverse interrupt (RVI) indicator 4-17

request-to-write function (continued)

using while receiving data 4-17

requesting device

using acquire operation with 4-11
using with evoke function 4-11

requests, formats for program start 5-4

response indicators

description 4-21
not valid with write operation 4-21
receive-detach 4-22
receive-end-of-group 4-21
receive-turnaround 4-21
using more than one on a read operation 4-21
valid with read operation 4-21

retrieving status

BSCSEL configuration 2-2
controller description 2-2
device description 2-2
line description 2-2

return codes

cancel-invite function 4-18
descriptions B-1
operation completed successfully 4-18
receive-detach response indicator 4-22
receive-end-of-group response indicator 4-21
receive-turnaround response indicator 4-22
request-to-write function 4-17
return code processing 4-22
timer ends 4-18
timer function 4-18

reverse-interrupt indicator

programming considerations 5-16
sending with request-to-write function 4-17

RPG/400 source program (local system)

configuration
Create BSC Controller Description (CRTCTLBSC)
command E-26
Create BSC Device Description (CRTDEVBSC)
command E-26
Create BSC Line Description (CRTLINBSC)
command E-26
program files
BSCSELICF E-26
BSCSELPRT E-29
DBFILE E-27

RPG/400 source program (remote system)

configuration
Create BSC Controller Description (CRTCTLBSC)
command E-37
Create BSC Device Description (CRTDEVBSC)
command E-37
Create BSC Line Description (CRTLINBSC)
command E-37
program files
BSCSELICF E-37
BSCSELPRT E-40
DBFILE E-39

running BSCCL 3-1

S

safe indicator 4-22

security function

- parameters
 - password 4-13
 - user identifier 4-13
- specifying parameters 4-12
- with evoke function 4-13

sending data 4-13

session

- activity from a program start request 5-6
- conducting consecutively 1-2
- ending 4-20
- for multipoint tributary lines 1-2
- for point-to-point lines 1-2
- getting current status of 4-19
- number supported per line 1-2
- starting 4-10

setting up BSCCL

- creating BSCCL configuration 2-1
- entering BSCCL commands 2-1

source program

- notifying of errors 4-16
- starting a session from 4-10
- using release operation 4-20

specifying program device entry commands 4-2

start requests, formats for program 5-4

starting

- a session 4-10
- a transaction 4-11

status message from device 5-16

subdevice selection

- description 4-14
- sending as a separate record 4-15

switched lines

- with allow-write function 4-18
- with cancel-invite function 4-18

system-supplied formats

- description of A-4

T

target program

- data format used 5-7
- maximum length of name 5-7
- specifying with evoke function 4-12, 5-7
- started by evoke parameters 5-8
- starting a session from 4-11
- starting other programs 5-8
- using user-defined parameters 5-2

timer function

- description 4-18
- receiving 0310 return code (timer ends) 4-18

transaction

- ending 4-19

transaction (continued)

- number active at one time 1-2, 4-11
- sending data 4-13
- starting 4-11

U

user identifier parameter 4-13

user-defined parameters

- entering 5-2
- separating from positional 5-2
- specifying with evoke function 4-12

V

variable-length-data-record function

- description 4-15, 4-16

Vary Configuration (VRYCFG) command 3-1

varying off the configuration 3-1

varying on the configuration 3-1

VRYCFG command parameters 3-1

VRYCFG (Vary Configuration) command 3-1

W

write operation

- blocking records 4-14
- combining with other operations 4-13
- description 4-13
- using response indicators 4-21

Numerics

3740 Data Entry System

- programming considerations 5-12
- using multiple file formats 5-13
- using program start requests 5-13

6580 Displaywriter

- programming considerations 5-14
- specifying block size 5-14

Special Characters

***EOX**

- See End-of-Transaction (*EOX) command

READER'S COMMENT FORM

Please use this form only to identify publication errors or to request changes in publications. Direct any requests for additional publications, technical questions about IBM systems, changes in IBM programming support, and so on, to your IBM representative or to your IBM-approved remarketer. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

- If your comment does not need a reply (for example, pointing out a typing error), check this box and do not include your name and address below. If your comment is applicable, we will include it in the next revision of the manual.
- If you would like a reply, check this box. Be sure to print your name and address below.

Page number(s):

Comment(s):

Please contact your IBM representative or your IBM-approved remarketer to request additional publications.

Name

Company or
Organization

Address

City

State

Zip Code

Phone No.

Area Code

No postage necessary if mailed in the U.S.A.

Cut or
Along

Fold and Tape

Please do not staple

Fold and Tape

BUSINESS REPLY MAIL

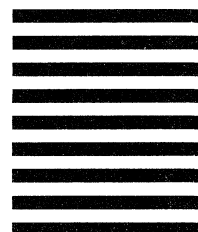
FIRST CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Information Development
Department 245
3605 North Hwy 52
ROCHESTER MN 55901-9986



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES



Fold and Tape

Please do not staple

Fold and Tape



Cut or
Along

READER'S COMMENT FORM

Please use this form only to identify publication errors or to request changes in publications. Direct any requests for additional publications, technical questions about IBM systems, changes in IBM programming support, and so on, to your IBM representative or to your IBM-approved remarketer. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

- If your comment does not need a reply (for example, pointing out a typing error), check this box and do not include your name and address below. If your comment is applicable, we will include it in the next revision of the manual.
- If you would like a reply, check this box. Be sure to print your name and address below.

Page number(s):

Comment(s):

Please contact your IBM representative or your IBM-approved remarketer to request additional publications.

Name

Company or
Organization

Address

City

State

Zip Code

Phone No.

Area Code

No postage necessary if mailed in the U.S.A.

Cut or
Along

Fold and Tape

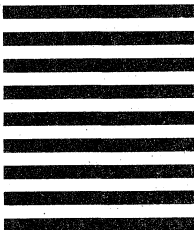
Please do not staple

Fold and Tape



BUSINESS REPLY MAIL
 FIRST CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

NO POSTAGE
 NECESSARY
 IF MAILED IN THE
 UNITED STATES



POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
 Information Development
 Department 245
 3605 North Hwy 52
 ROCHESTER MN 55901-9986



Fold and Tape

Please do not staple

Fold and Tape

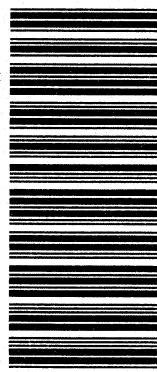


Cut
Along



Program Number
5728-SS1

21F2708



SC21-9593-1

