

AS/400 Advanced Series



# BSC Equivalence Link Programming

*Version 4*



AS/400 Advanced Series



# BSC Equivalence Link Programming

*Version 4*

**Take Note!**

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

**First Edition (August 1997)**

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

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

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

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

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

or you can fax your comments to:

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

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

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

© **Copyright International Business Machines Corporation 1997. All rights reserved.**

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

---

# Contents

<b>Notices</b> . . . . .	vii	Notifying the Remote Program of Problems	4-15
Programming Interface Information . . . . .	vii	Fail Function . . . . .	4-15
Trademarks and Service Marks . . . . .	viii	Using Additional Functions and Operations	4-15
 		Request-to-Write Function . . . . .	4-15
<b>About This Book</b> . . . . .	ix	Allow-Write Function . . . . .	4-16
Who Should Use This Book . . . . .	ix	Cancel-Invite Function . . . . .	4-16
Prerequisite and Related Information . . . . .	ix	Timer Function . . . . .	4-16
Information Available on the World Wide Web	ix	Get-Attributes Operation . . . . .	4-17
 		Ending a Transaction . . . . .	4-17
<b>Chapter 1. Introduction to the Binary Synchronous Communications</b>		Detach Function . . . . .	4-17
<b>Equivalence Link</b> . . . . .	1-1	Ending a Session . . . . .	4-17
Remote Systems Supported . . . . .	1-1	Release Operation . . . . .	4-17
Communications Lines Supported . . . . .	1-2	End-of-Session Function . . . . .	4-18
Data Formats Supported . . . . .	1-3	Using Response Indicators . . . . .	4-18
 		Receive-End-of-Group . . . . .	4-18
<b>Chapter 2. Configuring BSCCL</b> . . . . .	2-1	Receive-Turnaround . . . . .	4-19
Defining the BSCCL Configuration . . . . .	2-1	Receive-Detach . . . . .	4-19
 		Using the Input/Output Feedback Area . . . . .	4-19
<b>Chapter 3. Running BSCCL</b> . . . . .	3-1	Using Return Codes . . . . .	4-19
Vary On and Vary Off Support . . . . .	3-1	 	
 		<b>Chapter 5. BSCCL Considerations</b> . . . . .	5-1
<b>Chapter 4. Writing Application Programs</b> . . . . .	4-1	Program Start Request Considerations . . . . .	5-1
Using an Intersystem Communications Function File . . . . .	4-1	Syntax for Program Start Requests . . . . .	5-1
Specifying the Program Device Entry Commands . . . . .	4-2	Formats for Program Start Requests . . . . .	5-3
Comparing the Program Device Entry Command and Configuration Parameters . . . . .	4-7	Continuation Format for Program Start Requests . . . . .	5-4
Determining Record Lengths . . . . .	4-8	Examples of Program Start Requests . . . . .	5-4
Starting a Session . . . . .	4-9	Program Start Requests Received from Remote Systems . . . . .	5-5
Open or Acquire Operation . . . . .	4-10	Program Start Requests Sent by the AS/400 System . . . . .	5-6
Starting a Transaction . . . . .	4-10	Failed Program Start Requests . . . . .	5-6
Evoke Function . . . . .	4-11	Prestarting Jobs for Program Start Requests . . . . .	5-8
RMTBSCCL Considerations . . . . .	4-11	Remote System Considerations . . . . .	5-8
Security Function . . . . .	4-12	BSCCL Commands . . . . .	5-9
Sending Data . . . . .	4-12	Receiving Null Records . . . . .	5-9
Write Operation . . . . .	4-12	Using 3740 Data Entry Systems . . . . .	5-10
Subdevice Selection Function . . . . .	4-13	Using Office Systems . . . . .	5-10
End-of-Group Function . . . . .	4-13	Programming Considerations . . . . .	5-11
Variable-Length-Data Function . . . . .	4-14	BSCCL Online Messages . . . . .	5-11
Receiving Data . . . . .	4-14	The Reverse-Interrupt Indicator . . . . .	5-12
Read Operation . . . . .	4-14	Binary Synchronous Communications Considerations . . . . .	5-13
Invite Function . . . . .	4-14	 	
Read-from-Invited-Program-Devices Operation . . . . .	4-14	<b>Appendix A. Language Operations, DDS Keywords, and System-Supplied Formats</b>	A-1
Record-Identification Function . . . . .	4-14	Language Operations . . . . .	A-1
Variable-Length-Data Function . . . . .	4-14	Supported DDS Keywords . . . . .	A-3
Waiting for a Display File, an ICF File, and a Data Queue . . . . .	4-15	System-Supplied Formats . . . . .	A-3

<b>Appendix B. Return Codes, Messages, and Sense Codes</b>	B-1
Return Codes	B-1
Major Code 00	B-1
Major Code 02	B-4
Major Code 03	B-7
Major Code 04	B-9
Major Codes 08 and 11	B-10
Major Code 34	B-11
Major Code 80	B-12
Major Code 81	B-16
Major Code 82	B-22
Major Code 83	B-32
<b>Appendix C. BSC Control Codes and Protocols</b>	C-1
<b>Appendix D. EBCDIC and ASCII Character Sets</b>	D-1
EBCDIC Character Set	D-1
ASCII Character Set	D-2
<b>Appendix E. Binary Synchronous Communications Configuration Example</b>	E-1
Configuring for BSCEL Communications	E-1
Configuring the System/36 for BSCEL Communications	E-1
Configuring the AS/400 System for BSCEL Communications	E-2
<b>Appendix F. BSCEL Application Flow Examples</b>	F-1
<b>Appendix G. Example Programs</b>	G-1

COBOL/400 Source Program for Local System	G-1
Configuration	G-1
Program Files	G-1
Program Explanation	G-5
COBOL/400 Source Program for Remote System	G-12
Configuration	G-12
Program Files	G-12
Program Explanation	G-15
RPG/400 Source Program for Local System	G-21
Configuration	G-21
Program Files	G-21
Program Explanation	G-24
RPG/400 Source Program for Remote System	G-29
Configuration	G-29
Program Files	G-29
Program Explanation	G-32
ILE C/400 Source Program for Local System	G-36
Configuration	G-36
Program Files	G-36
Program Explanation	G-40
ILE C/400 Source Program for Remote System	G-44
Configuration	G-44
Program Files	G-45
Program Explanation	G-48
<b>Bibliography</b>	H-1
AS/400 Books	H-1
Other Related Books	H-1
<b>Index</b>	X-1

# Figures

1-1.	BSC/CL Example Communications Network Configuration . . . . .	1-3	G-3.	DDS for the Database File Used in BSC/CL Data Transfer on the Local System (COBOL/400 Language) . . . . .	G-3
4-1.	BSC/CL Control Characters That Cannot Be Used as Record Separator Characters . . . . .	4-5	G-4.	DDS for the Printer File on the Local System (COBOL/400 Language) . . . . .	G-4
4-2.	Comparing Configuration Command Parameters and Program Device Entry Command Parameters . . . . .	4-8	G-5.	COBOL/400 Source Program for the Local System . . . . .	G-7
5-1.	Syntax for Program Start Requests . . . . .	5-1	G-6.	DDS for the BSC/CLICF Program File Used by the Remote System (COBOL/400 Language) . . . . .	G-13
5-2.	Coding Positions for a Program Start Request . . . . .	5-2	G-7.	DDS for the Database File Used in BSC/CL Data Transfer Program on the Remote System (COBOL/400 Language) . . . . .	G-14
5-3.	Continuation Format for Program Start Requests . . . . .	5-4	G-8.	DDS for the Printer File on the Remote System (COBOL/400 Language) . . . . .	G-14
5-4.	*EXEC Format for Program Start Request . . . . .	5-5	G-9.	COBOL/400 Source Program for the Remote System . . . . .	G-16
5-5.	Continuation Format for Program Start Request . . . . .	5-5	G-10.	DDS for the BSC/CLICF File Used by the Local System (RPG/400 Language) . . . . .	G-22
5-6.	Session Activity . . . . .	5-5	G-11.	DDS for the Database File Used in BSC/CL Data Transfer on the Local System (RPG/400 Language) . . . . .	G-23
5-7.	Reason Codes for Rejected Program Start Requests . . . . .	5-7	G-12.	DDS for the Printer File on the Local System (RPG/400 Language) . . . . .	G-23
A-1.	ICF Operations Supported by BSC/CL . . . . .	A-1	G-13.	RPG/400 Source Program for the Local System . . . . .	G-25
A-2.	ICF Operations and Equivalent Language Statements . . . . .	A-2	G-14.	DDS for the BSC/CLICF Program File Used by the Remote System (RPG/400 Language) . . . . .	G-29
B-1.	Actions for Return Code 0000 . . . . .	B-2	G-15.	DDS for Database File Used in BSC/CL Data Transfer Programs on the Remote System (RPG/400 Language) . . . . .	G-30
C-1.	BSC Line Protocols and Control Characters . . . . .	C-1	G-16.	DDS for the Printer File on the Remote System (RPG/400 Language) . . . . .	G-31
D-1.	EBCDIC Character Set . . . . .	D-1	G-17.	RPG/400 Source Program for the Remote System . . . . .	G-33
D-2.	ASCII Character Set . . . . .	D-2	G-18.	DDS for the SRCICFF File Used by the Local System (ILE C/400 Language) . . . . .	G-37
F-1.	Communications between Two AS/400 Application Programs . . . . .	F-1	G-19.	DDS for the Database File Used in BSC/CL Data Transfer on the Local System (ILE C/400 Language) . . . . .	G-38
F-2.	Communications between an AS/400 System and a Remote Device Not Using BSC/CL . . . . .	F-2			
F-3.	Communications between an AS/400 System and a Remote System Using Multiple Files . . . . .	F-3			
F-4.	Starting an AS/400 System from a Remote System . . . . .	F-4			
F-5.	An AS/400 System Receiving Data from an Office System . . . . .	F-6			
F-6.	An AS/400 System Sending Data to an Office System . . . . .	F-7			
G-1.	Communication between a Local AS/400 Program and a Remote AS/400 Program . . . . .	G-1			
G-2.	DDS for the BSC/CLICF File Used by the Local System (COBOL/400 Language) . . . . .	G-2			

<p>G-20. DDS for the Printer File on the Local System (ILE C/400 Language) . . . . . G-39</p> <p>G-21. ILE C/400 Source Program for the Local System . . . . . G-41</p> <p>G-22. DDS for the TGTICFF File Used by the Remote System (ILE C/400 Language) . . . . . G-46</p>	<p>G-23. DDS for Database File Used in BSC/CEL Data Transfer Programs on the Remote System (ILE C/400 Language) . . . . . G-47</p> <p>G-24. DDS for the Printer File on the Remote System (ILE C/400 Language) . . . . . G-47</p> <p>G-25. ILE C/400 Source Program for the Remote System . . . . . G-49</p>
---	--



---

## Notices

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

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

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

Address your questions to:

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

This publication could contain technical inaccuracies or typographical errors.

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

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

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

---

## Programming Interface Information

This Programming provides programming information needed to use the binary synchronous communications equivalence link (BSCCL) with the AS/400 system. The *BSC Equivalence Link Programming* contains no programming interfaces for customers.

---

## Trademarks and Service Marks

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

Application System/400	ILE
AS/400	Operating System/400
C/400	OS/400
COBOL/400	RPG/400
FORTRAN/400	Series/1
IBM	400

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

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

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

C-bus is a trademark of Corollary, Inc.

Java and HotJava are trademarks of Sun Microsystems, Inc.

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

---

## About This Book

This book supplies the programming information you need to use the binary synchronous communications equivalence link (BSCSEL) with the IBM AS/400 system.

For a list of related publications, see the "Bibliography."

---

## Who Should Use This Book

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

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

- General communications concepts. AS/400 communications concepts are covered in the book, *System Operation*.
- 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 Configuration*.

- Intersystem communications function (ICF) support described in the *ICF Programming*.

---

## Prerequisite and Related Information

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

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

---

## Information Available on the World Wide Web

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

<http://www.as400.ibm.com>

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



---

# Chapter 1. Introduction to the Binary Synchronous Communications Equivalence Link

*BSC Equivalence Link Programming* supplies distributed data processing support to the IBM\* Application System/400\* (AS/400\*) system users who want to communicate with a remote system or device using binary synchronous communications (BSC). **Binary synchronous communications (BSC)** is a data communications line protocol that uses a standard set of **transmission control characters**<sup>1</sup> and control character sequences to send binary-coded data over a communications line. **Binary synchronous communications equivalence link (BSCSEL) support** is the inter-system communications function (ICF) support on the AS/400 system that provides binary synchronous communications with a remote system or device. BSCSEL also supplies online and batch communications between application programs on different BSC systems. AS/400 application programs can be written in the Integrated Language Environment (ILE) C/400\*, ILE COBOL/400\*, ILE FORTRAN/400\*, or ILE RPG/400\* programming languages.

This chapter lists the remote systems, communications lines, and data formats supported by BSCSEL and presents an example showing an AS/400 configuration of a BSCSEL communications network.

---

## Remote Systems Supported

BSCSEL communicates with another AS/400 system that has BSC or BSCSEL support. You can also use an AS/400 system with BSCSEL support to communicate with any of the following IBM systems:

- System/38 with BSC support
- System/36 with SSP-ICF BSCSEL support or with SSP BSC telecommunications support
- System/34 with SSP-ICF BSCSEL 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
  - IBM 6580 Displaywriter System
- IBM 3747 Data Converter
- IBM 3750 Switching System (except in the United States and Canada)

BSCSEL 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. BSCSEL 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 BSCSEL, you can start a program on the remote

---

<sup>1</sup> A transmission control character is a special character included in a message to control communications over a data link.

system or the remote system can start a program on the local AS/400 system.

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

BSCEL does not emulate the IBM 3780 data communication terminal. However, BSCEL can communicate with devices that use 3780 implementation.

BSCEL supports some of the standard features of the 3780 protocol, such as point-to-point communications, record blocking, IRS record separator character, EBCDIC transparent text mode, and blank compression when transparent text mode is not used. BSCEL does not support other features of the 3780 protocol, such as audible alarm, conversational mode, and multipoint data link control. Refer to the book, *Component Information for the IBM 3780 Data Communication Terminal*, for additional information about the features of the 3780 terminal.

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:

- BSCEL supports only one communications session at a time on each line.
- A communications program, using BSCEL, cannot share a communications line with another communications program.

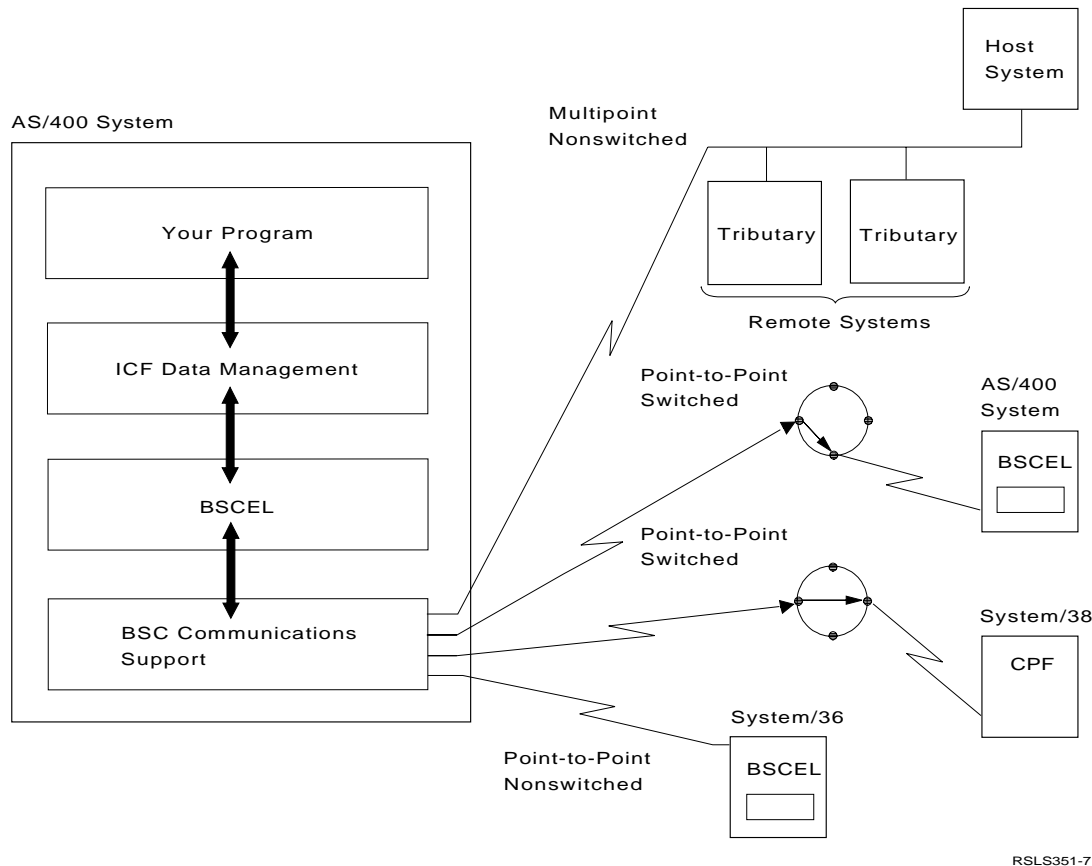
By using multipoint tributary communications lines:

- BSCEL supports more than one communications session on each line. (Each communications session is associated with a different BSC device description.)
- BSCEL 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 BSCEL 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 immediately. 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 4-10.)

Figure 1-1 is an example of a BSCEL network that connects the AS/400 system and remote systems with various line types. (For information on creating a BSCEL configuration, see Chapter 2, and the *Communications Configuration*.)



RSLS351-7

Figure 1-1. BSC Example Communications Network Configuration

## Data Formats Supported

BSCCEL 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 4-7 and “Specifying the Program Device Entry Commands” on page 4-2.)

- Data records that are not blocked either in **transparent text mode**<sup>2</sup> or not in transparent text mode
- Blocked data records in either transparent text mode or not in transparent text mode
- Blocked data records with record separators not in transparent text mode
- Blocked data records with intermediate-text-block (ITB) characters not in transparent text mode

- Blocked data records with blank compression not in transparent text mode
- Blocked data records with blank truncation in either transparent text mode or not in transparent 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.)

BSCCEL 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.)

<sup>2</sup> A method of transmission in which only transmission control characters preceded by the DLE control character are processed as transmission control characters.

The AS/400 system processes all data in EBCDIC characters.

If the remote system uses ASCII characters, BSCEL 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 BSC

This chapter describes the commands used for configuring BSC.

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

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

The following is a brief introduction of the commands you use to configure BSC. For a complete description of these and related commands, see the *Communications Configuration* book and the online help.

---

### Defining the BSC Configuration

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

A BSC 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 (BSC) (CRTLINBSC)
- Create Controller Description (BSC) (CRTCTLBSC)
- Create Device Description (BSC) (CRTDEVBSC)

The Retrieve Configuration Source (RTVCFGSRC) command can also be used to retrieve CL command source to create existing configuration objects.

The Rename Object (RNMOBJ) command can be used to change the name of a configuration object in a library.

To change one or more of the attributes of a BSC 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. 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 (DLTDEV)



---

## 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 (VRYCFG) command, refer to the *Communications Management* book.

The VRYCFG command has the following parameters (the default values are underlined>):

#### 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.

#### 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 between specifying RANGE(\*OBJ) or RANGE(\*NET).

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

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

#### VRYWAIT

Specifies whether the BSC line description is varied on asynchronously or synchronously. Specify how long the system waits for vary on to be completed (for synchronous vary on) after which the communications file is opened and the session is acquired.

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

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

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

#### ASCVRYOFF

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

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

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

**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 intersystem communications function (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 input/output (I/O) feedback area (which contains the results of read and write operations) and return code processing.

---

### Using an Intersystem Communications Function File

The **intersystem communications function (ICF)** is a function of the operating system that allows a program to communicate interactively with another program or system. An 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 data description specifications (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 *ICF Programming* book.

<b>CRTICFF</b>	The Create ICF File command allows you to create an ICF file.
<b>CHGICFF</b>	The Change ICF File command allows you to change the file attributes of the ICF file.
<b>OVRICFF</b>	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.
<b>DLTF</b>	The Delete File command allows you to delete a file from the system.
<b>DSPFD</b>	The Display File Description command displays the file description of any file on the system. This information may be printed or displayed.
<b>DSPFFD</b>	The Display File Field Description command displays the description of the fields in any file on the system. This information may be printed or displayed.
<b>ADDICFDEVE</b>	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 to an ICF file.
<b>CHGICFDEVE</b>	The Change ICF Device Entry command allows you to permanently change the program device attributes previously added with the ADDICFDEVE command.
<b>OVRICFDEVE</b>	The Override ICF Device Entry command allows you to: <ul style="list-style-type: none"><li>• Temporarily add the program device entry, the remote location information, and the session level attributes to the ICF file.</li><li>• Temporarily change a program device entry with the</li></ul>

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 to an ICF file with the ADDICFDEVE command or changed with the CHGICFDEVE command.

---

## Specifying the Program Device Entry Commands

The following describes the parameters for the ADDICFDEVE, CHGICFDEVE, and OVRICFDEVE commands and lists the valid values for each parameter for BSC. 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:** BSC uses the library list to locate the ICF file.

**\*CURLIB:** BSC uses the current library for the job to locate the ICF file. If no current library entry exists in the library list, BSC 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. BSC 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**, the program that is started on the remote system at the request of the source system, always uses

\*REQUESTER as the remote location name in the ICF file to connect to the session that the **source program**, the program that starts a session with a remote system, 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 (CRTDEVBS) 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 BSC remote location may be acquired to the file at any one time.

## FMTSLT

Specifies the record format selection used for input operations.

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

## CMNTYPE

Specifies which type of communications parameters show on the prompt screen. This parameter is used only for the purpose of prompting. You should specify the value \*BSCCEL or \*ALL for this parameter.

**\*BSCCEL:** The prompt for all BSCCEL-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 the ICF file 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:

**\*DEVVD:** BSCCEL 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 CRTDEVVBS command and the CHGDEVVBS 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) character**. This control character divides a block of text into smaller groups of text for an intermediate block check. 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. An **end-of-text (ETX) character** is a BSC transmission control character used to end a logical set of records that began with the start-of-text character. An **end-of-transmission-block (ETB) character** is a BSC transmission control character used to end a block of records. 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. The **interrecord-separator (IRS) character** is a transmission control character used to separate records within a block of data. 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 ICF file 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, BSCSEL scans the buffer for the last non-blank 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. BSCSEL 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, BSCSEL 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 RMTBSCSEL(\*YES).

BLOCK(\*USER) specifies that the application program will fully construct a transmission block (including all record separators, BSC framing characters, transparency characters, and so on). The data in the user's buffer on a write operation will be transmitted as a single transmission block, unaltered except for the adding of leading pad characters, synchronization characters, block check characters, trailing pad characters, and **data-link-escape (DLE) character**<sup>1</sup> insertions (as required for transparent mode). In addition, BSCSEL support removes all trailing blanks in the buffer before sending the data. Therefore, the last non-blank character in the buffer must be a valid BSC end frame character.

When receiving data with BLOCK(\*USER) specified, the data received is passed to the application program unaltered, except for the removal of the BSC pad characters, synchronization characters, block check characters, and transparency control DLE characters (unless the DLE precedes an STX). Any

trailing bytes in the buffer are filled with blanks.

The transparency selection (TRANSPY(\*YES/\*NO)) is ignored when BLOCK(\*USER) is specified. To effect transparent transmission of a block of data, the application program must prefix the block with DLESTX. BSC support will insert any additional DLE characters required in the block. Any other DLEs (except for the first DLE) are treated as data characters by the BSC support.

Variable length transmission blocks are normally required when using the BLOCK(\*USER) option. Therefore, the record format used on write operations should be large enough to contain the maximum block size to be written. The application program must ensure that all trailing bytes are blanks. BSCSEL support then truncates all bytes in the buffer following the ETB or ETX to determine the number of bytes to transmit.

When using BLOCK(\*USER), BSCSEL uses \*EOT for the value of the GRPSEP parameter; therefore, it is the responsibility of the application program to construct or recognize data group boundaries.

The BLOCK(\*USER) option must be used to send header (SOH) data or to recognize the receipt of header data. All other BLOCK options treat an SOH as an STX.

**\*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 CRTDEVBSC command and the CHGDEVBSC 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 by specifying a value ranging from 0 to 9 or A to F, as in

---

<sup>1</sup> A BSC transmission control character used to indicate that the next character is a control character, not a data character.



BLOCK(\*SEP A). If a record separator character is not specified, BSCCEL uses the record separator character hex 1E.

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

Figure 4-1. BSCCEL 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, BSCCEL 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 TRNSPY(\*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 TRNSPY(\*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 maximum record length, an error occurs when you send or receive the record.

**\*DEV D:** BSCCEL 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 4-8.

#### 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). BSCCEL checks to ensure that the maximum block length meets these requirements. If it does not, BSCCEL fails the acquire operation for the current session if RMTBSCCEL(\*YES) is specified. BSCCEL fails the first input or output operation if RMTBSCCEL(\*NO) is specified.

**\*DEV D:** BSCCEL 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 does not include record separator characters when blocking is used. However, because some receiving devices can include those characters in the count, you might have to either lower the BLKLEN value at the AS/400 system or raise the value (if possible) at the remote system or device to match the device requirements and the operating environment.

In addition to the record separator character, the following BSC control characters are not included when determining the block length:

STX  
ETX  
ETB  
SYN  
PAD  
DLE  
BCC

#### 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 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.

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, which can contain any hexadecimal value. 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 inter-group 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), BSCEL ignores the truncation indicator. You cannot specify TRUNC(\*YES) with BLOCK(\*NOSEP) or BLOCK(\*ITB). If you do, BSCEL 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 D:** BSCEL 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. An **end-of-transmission (EOT) character** is the control character used to end transmission with the remote system.

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

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

If you specify BLOCK(\*USER), BSCEL uses \*EOT for the value of the GRPSEP parameter. If you specify either \*DEV3740 or \*OFCSYS, BSCEL ignores this value when BLOCK(\*USER) is specified.

#### **RMTBSCEL**

Specifies the type of BSCEL session with the remote system.

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

**\*NO:** The remote system cannot recognize BSCEL 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 BSCEL 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 BSCEL 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:** BSCEL uses the switch type specified in the controller description (CRTCTLBSC 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 BSCEL session in any ICF file, unless those values are changed by the program device entry commands.

Figure 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 <sup>1</sup>
Program device name	–	PGMDEV
Remote location name	RMTLOCNAME	RMTLOCNAME <sup>2</sup>
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 <sup>3</sup>	GRPSEP <sup>3</sup>
Remote session indicator	RMTBSCSEL	RMTBSCSEL
Switch type for session start	INLCNN <sup>4</sup>	INLCNN
Secure from override indicator	–	SECURE <sup>5</sup>

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

2 If you specify RMTLOCNAME(\*REQUESTER) on the command, BSCSEL 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 RMTBSCSEL.) Instead, the remote program start request determines the data format parameters for this session.

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

3 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.

4 You specify this parameter when you create the controller description (CRTCTLBSC command).

5 This parameter is only valid on the OVRICFDEVE command.

Figure 4-2 shows the relationship between the BSCSEL 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 (CRTDEVBS command).

The ADDICFDEVE and CHGICFDEVE program device entry commands cause permanent changes for any BSCSEL 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 BSCSEL session that uses the specified program device.

For additional information about the relationship of these commands, refer to the *ICF Programming*.

## 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 the ICF file. 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 ICF file, the input length is always 4096, unless you override or change the MAXRCDLEN parameter on the OVRICFF or CHGICFF commands.

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 CRTDEVBSC, CHGDEVBSC, ADDICFDEVE, 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 CRTDEVBSC, CHGDEVBSC, 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 CRTLINBSC and CHGLINBSC 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 4-11 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 RMTBSCEL(\*YES) is specified. See "RMTBSCEL Considerations" on page 4-11 for information on how to specify RMTBSCEL(\*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 or 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, CHGICFDEVE, 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 a program device name or the ACQPGMDEV parameter for the ICF file on the CRTICFF, CHGICFF, or OVRICFF command. For a description of the open operation, see the *ICF Programming*.

**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, CHGICFDEVE, or OVRICFDEVE command. Any values specified for the data format parameters on the ADDICFDEVE, CHGICFDEVE, 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, CHGICFDEVE, or OVRICFDEVE command. You must specify a requesting device for the remote location (RMTLOCNAME(\*REQUESTER)) on the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command for the target program that is started by a program start request.

Any values specified for the ADDICFDEVE, CHGICFDEVE, 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 BSCSEL 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 BSCSEL 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 RMTBSCEL(\*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 BSCEL 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

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

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

## RMTBSCEL 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 RMTBSCEL(\*NO) when your program communicates with other systems or devices (such as a System/38 or a 3741 Data Entry Station) that cannot recognize BSCEL commands and online messages. Specify RMTBSCEL(\*YES) when your program communi-

cates with other systems that are also using BSCEL.

**Specifying RMTBSCEL(\*NO):** When you specify RMTBSCEL(\*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. BSCEL 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 BSCEL. If an evoke function is not used, the first input or output operation starts the transaction.

The evoke function is optional when you specify RMTBSCEL(\*NO).

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

**Specifying RMTBSCEL(\*YES):** When you specify RMTBSCEL(\*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 RMTBSCEL(\*YES) parameter causes BSCEL 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 started. 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 RMTBSCSEL(\*YES) and use an evoke function, you may need to provide security information for the program start request that BSCSEL 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.

BSCSEL does not support the profile ID parameter 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 4-8.)

## Write Operation

The write operation issues a data record from your program to the remote system in this session. BSCSEL sends the record in the data format specified by the configuration parameters or by the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command. When you use a write operation, it issues one data record to BSCSEL 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.

**Note:** If you specify BLOCK(\*USER), you must specify an output record length of at least two or an error will occur.

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 Appendix B 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, BSCSEL sends an end-of-transmission indication to the remote system. If you specify GRPSEP(\*DEV3740) for this session, BSCSEL 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, BSCSEL sends an end-of-text (ETX) control character after the last data record, and then sends the end-of-transmission (EOT) indication. BSCSEL 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. A **read-from-invited-program-devices operation** is an input operation that waits for input from any one of the invited program devices for a user-specified time. Use the timer function or the WAITRCD parameter on the CRTICFF, CHGICFF, or OVRICFF 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**<sup>2</sup>, 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 RMTBSCSEL(\*NO)

**Specifying Record Blocking:** If you do not specify record blocking when you use a write operation, BSCSEL sends one data record to the remote system for each write operation. If you do specify record blocking, BSCSEL 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 RMTBSCSEL(\*NO) for the communications session.

BSCSEL 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.

BSCSEL ignores the subdevice selection if it has been selected at any other time. BSCSEL 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, BSCSEL indicates to the remote system that this is the last record of a user-defined group of records. This function also indicates to BSCSEL that your program is not requesting any input. BSCSEL 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, CHGICFDEVE, 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.

---

<sup>2</sup> In data communications, a contention state is a type of half-duplex line or data link control in which either user may transmit any time the line or link is available. If both users attempt to transmit at the same time, the protocols or the hardware determines who goes first.

## Variable-Length-Data Function

For information about determining the record length for an output operation, refer to “Determining Record Lengths” on page 4-8.

---

## 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 4-8.

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 *ICF Programming* 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, CHGICFDEVE, 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 4-8.

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

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

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

Use these commands to indicate a data queue in which entries are placed when one of the following occurs:

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

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

---

## Notifying the Remote Program of Problems

You can inform the other 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 BSCEL to discard all records not sent in the current transaction. If some records have been sent, BSCEL 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 BSCEL to discard all incoming records. BSCEL 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 operation with BSCEL:

- 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, BSCEL sends a reverse-interrupt (RVI) indicator to the remote system as the response to the next data record received. The **reverse-interrupt (RVI) character** is sent as a request from the receiving station to the sending station to stop transmitting and begin receiving a message. BSCEL sends only one RVI indicator to the remote system. If you issue additional request-to-write functions before the change of direction, BSCEL ignores them.

When your program uses the request-to-write function and the remote system is also using BSCEL, 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 BSCEL receives the RVI indication. 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 Appendix B.)

## 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 BSCEL 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. After a successful cancel-invite function, your program can attempt to either send or receive data again.

If data or a message is received, or if BSCEL 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 Appendix B for more information on return code 0412.)

**Using Switched Lines:** If you are using a switched line and specify RMTBSCEL(\*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 establishes 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. Refer to the *ICF Programming* book for more information about the get-attributes operation.

---

## 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 5-3 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 4-19 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 RMTBSCSEL(\*YES) for this session, BSCSEL sends a \*REL command to the remote system. This informs BSCSEL 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, BSCSEL 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. If the application issues an input/output operation as its next operation, it will not be successful. 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 5-3 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 when RMTBSCSEL(\*YES) is specified for the session. For more information, see Appendix B.

---

## 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 *ICF Programming*.

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 BSCEL, 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 pre-start 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.

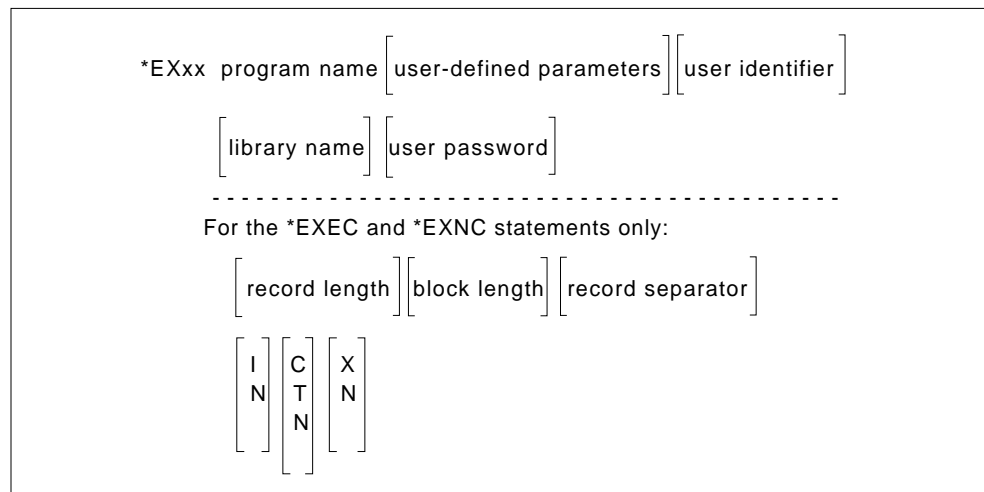
- If you specify RMTBSCCEL(\*NO) for the communications session, you can send data in the proper format for a program start request with your program's first output operation.

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 shows the syntax for BSCCEL program start requests.



RV2P801-0

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 communica-

tions 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.

Figure 5-2 lists the coding positions and describes the fields for a program start request.

Figure 5-2 (Page 1 of 2). Coding Positions for a Program Start Request

Coding Positions	Field	Description
1 to 6	*EXxx	Type of program start request used to start the program (xx = EC, EX, NC, or NX). 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 <sup>1</sup>	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 <sup>2</sup>	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 <sup>2</sup>	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 <sup>2</sup>	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 <sup>3</sup>	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 <sup>3</sup>	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 <sup>3</sup>	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 <sup>3</sup>	Indicates if ITB characters are used to separate records in a block. Specify I to use ITB characters, N to not use them.

Figure 5-2 (Page 2 of 2). Coding Positions for a Program Start Request

Coding Positions	Field	Description
159	Blank control <sup>3</sup> (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 <sup>3</sup> (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 combined length of the specified target program name, blank separators, and user-defined parameters should be 120 bytes in positions 7 through 126 of the program start request.
- 2 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.
- 3 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 BSCEL commands and online messages are passed from BSCEL 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 BSCEL 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 BSCEL to recognize BSCEL 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 BSCEL to recognize BSCEL 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 BSCEL 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 BSCEL commands or online messages.

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

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

- If the evoke function is sent with a detach function, BSCSEL sends \*EXEX.
- If the evoke function is sent without a detach function, BSCSEL 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 BSCSEL 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, BSCSEL does not send BSCSEL 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. BSCSEL allows you to split 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.

**Note:** If the program start request is split into two records, each record must be transmitted as a separate data block.

Figure 5-3 shows the continuation format for a program start request.

Figure 5-3. Continuation Format for Program Start Requests

Position	Field
<b>First Record</b>	
1 through 5	*EXNC or *EXNX
6	C
7 through xx	Program name
xx through 126	User-defined parameters
<b>Second Record</b>	
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). BSCSEL 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 5-4 on page 5-5 shows the record format for \*EXEC in the standard format of 160 bytes.

Figure 5-5 on page 5-5 shows the same information as in Figure 5-4 on page 5-5, but uses the continuation format (with two records) with the \*EXNC statement.

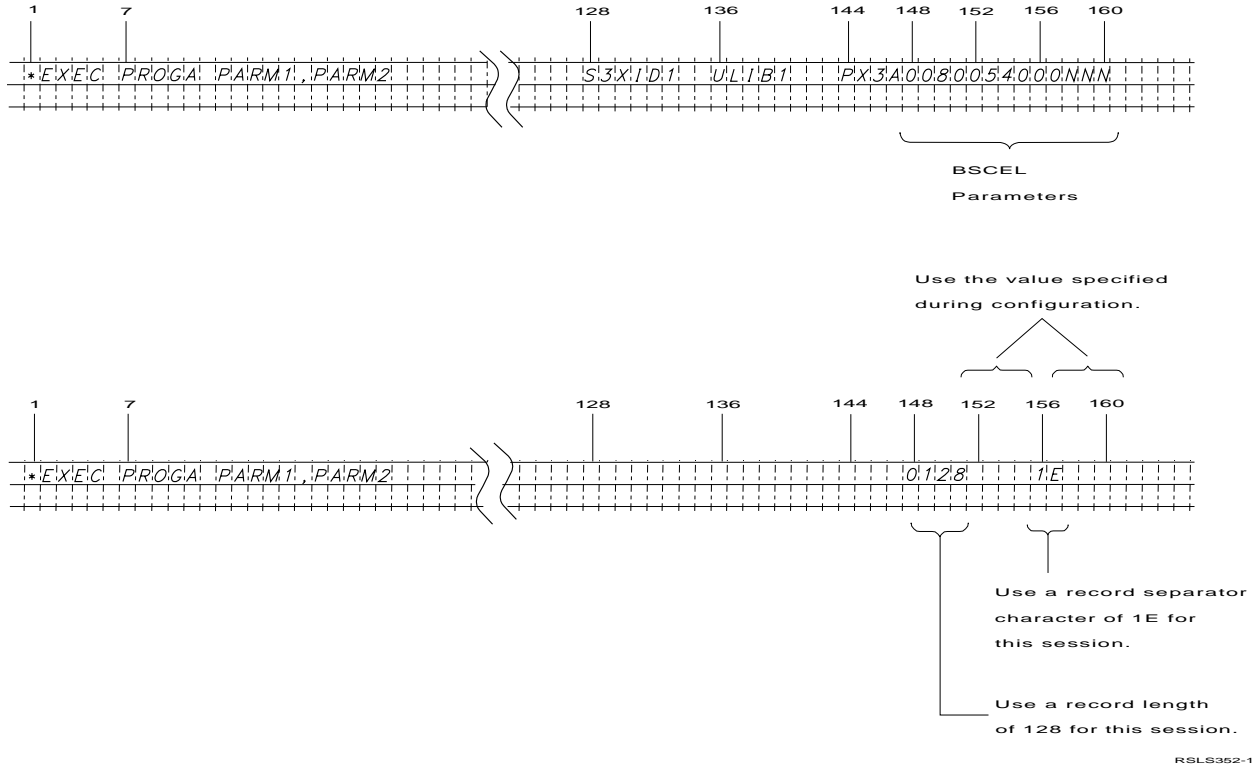


Figure 5-4. \*EXEC Format for Program Start Request

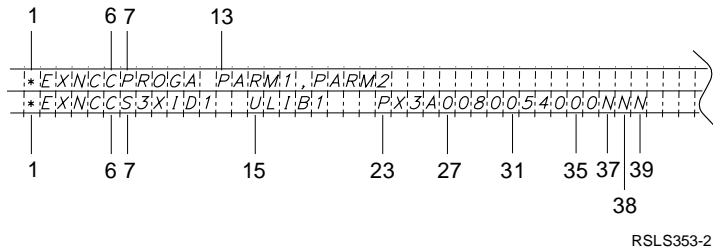


Figure 5-5. Continuation Format for Program Start Request

## 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.

Figure 5-6 describes the session activity for an AS/400 target program started by a program start request.

Figure 5-6 (Page 1 of 2). 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 BSCCEL 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.	BSCCEL ends the session normally.

Figure 5-6 (Page 2 of 2). Session Activity

Activity	Result
Your target program tries to send or receive data after the transaction ends.	BSCCEL 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 BSCCEL. To send a program start request, specify RMTBSCCEL(\*YES) on the BSCCEL configuration or on the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command. When you use an evoke function, BSCCEL 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 started. 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 4-12.

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 BSCCEL 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.

Figure 5-7 describes the reason codes for failed program start requests.

Figure 5-7 (Page 1 of 2). Reason Codes for Rejected Program Start Requests

Reason Code	Reason Description
401	Program start request received to a device that is not allocated to an active subsystem.
402	Requested device is currently being held by a Hold Communications Device (HLDCMNDEV) command.
403	User profile is not accessible.
404	Job description is not accessible.
405	Output queue is not accessible.
406	Maximum number of jobs defined by subsystem description are already active.
407	Maximum number of jobs defined by communications entry are already active.
408	Maximum number of jobs defined by routing entry are already active.
409	Library on library list is exclusively in use by another job.
410	Group profile cannot be accessed.
411	Insufficient storage in machine pool to start job.
412	System values not accessible.
501	Job description was not found.
502	Output queue was not found.
503	Class was not found.
504	Library on initial library list was not found.
505	Job description or job description library is damaged.
506	Library on library list is destroyed.
507	Duplicate libraries were found on library list.
508	Storage-pool defined size is zero.
602	Transaction program-name value is reserved but not supported.
604	Matching routing entry was not found.
605	Program was not found.
704	Password is not valid.

Figure 5-7 (Page 1 of 2). Reason Codes for Rejected Program Start Requests

Reason Code	Reason Description
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.
726	User profile has been disabled.
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.

Figure 5-7 (Page 2 of 2). 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. <sup>1</sup>
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. <sup>2</sup>
1914	Program name not valid.

Figure 5-7 (Page 2 of 2). Reason Codes for Rejected Program Start Requests

Reason Code	Reason Description
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 *ICF Programming* book.

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

Some remote systems use BSCSEL to communicate with the AS/400 system, other remote systems do not use BSCSEL.

If the remote system uses BSCSEL, consider the following:



- If the local AS/400 system starts the session with an acquire operation, specify RMTBSCSEL(\*YES) in the BSCSEL configuration description, or on the ADDICFDEVE, OVRICFDEVE, or CHGICFDEVE 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 BSCSEL, consider the following:

- If the local AS/400 system starts the session with an acquire operation, specify RMTBSCSEL(\*NO) in the BSCSEL configuration description or on the ADDICFDEVE, OVRICFDEVE, or CHGICFDEVE command. If the AS/400 system is expected to receive first, the application program on the AS/400 system 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 BSCSEL.

## BSCSEL Commands

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

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

**Acquire (\*ACQ) Command:** BSCSEL sends the Acquire (\*ACQ) command to the remote system when an acquire operation is used in an AS/400 program. When BSCSEL on the remote system receives the \*ACQ command, the issuing system acquires the session. Neither local nor remote BSCSEL accepts any other acquire oper-

ations (except acquire operations for requesting devices by remote BSCSEL) 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 BSCSEL at a remote system. A \*REL command received by BSCSEL 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 BSCSEL 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. BSCSEL 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 BSCSEL device description or on the ADDICFDEVE,

CHGICFDEVE, 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 BSC device description or on the ADDICFDEVE, CHGICFDEVE, 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 in Appendix B.) 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 BSCSEL to send and receive multiple files in the 3740 format, specify GRPSEP(\*DEV3740) in the BSC device description or on the ADDICFDEVE, CHGICFDEVE, 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 BSCSEL 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.)  
BSCSEL sends the last data record and a null record followed by an EOT control character. BSCSEL 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 Appendix B.)

## Using Office Systems

To configure BSCSEL to send and receive office documents from systems and devices, such as the 6580 Displaywriter system, specify GRPSEP(\*OFCSYS) for the BSC device description or on the ADDICFDEVE, CHGICFDEVE, or 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 document 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 Appendix B.)

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 Appendix B.)

**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<sub>nnn</sub> message-text

where:

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

n<sub>nnn</sub> = the reason code associated with the failure of the program start request. Refer to Figure 5-7 on page 5-7 for the reason codes of the failed program start requests.

ICFE CPI<sub>nnnn</sub> message-text

where:

n<sub>nnnn</sub> = 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 evoke 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 book 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 indicator 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).

## Binary Synchronous Communications Considerations

When using binary synchronous communications, there is no response sent to an EOT (end-of-transmission) control character. The sending station assumes the EOT is received after it sends the last data block. If the EOT is not received, data integrity is not assured.

It is possible to circumvent the binary synchronous handling of the EOT control character by using user-implemented error detection and recovery capabilities. Some of these include:

- Sequential block numbering
- Appropriate checkpoint-restart capabilities
- Job numbering
- Message numbering
- Data format checking



---

## 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

Figure A-1 describes the language operations supported by ICF.

*Figure A-1. ICF Operations Supported by BSCEL*

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.
Release	Attempts to end a session.
Close	Closes the ICF file.

---

Figure A-2 shows all the valid operations for each programming language that supports ICF (ILE

RPG/400, ILE COBOL/400, ILE C/400, ILE FORTRAN/400).

**Note:** C/400 statements are case sensitive.

Figure A-2. ICF Operations and Equivalent Language Statements

ICF Operation	ILE RPG/400 ILE COBOL/400		ILE C/400 Function	ILE FORTRAN/400 Statement
	Operation Code	Procedure Statement		
Open	OPEN	OPEN	fopen, _Ropen	OPEN
Acquire	ACQ	ACQUIRE	QXXACQUIRE, _Racquire	Not supported <sup>2</sup>
Get attributes	POST	ACCEPT	QXXDEVATR, _Rdevatr	Not supported
Read	READ	READ	fread, _Readn	READ
Read-from-invited-program-devices	READ <sup>1</sup>	READ <sup>1</sup>	QXXREADINVDEV, followed by an fread, _Readindv	Not supported
Write	WRITE	WRITE	fwrite, _Rwrite	WRITE
Write/Read	EXFMT	Not supported	_Rwriterd	_Rwriterd
Release	REL	DROP	QXXRELEASE, _Rrelease	Not supported
Close	CLOSE	CLOSE	fclose, _Rclose	CLOSE

<sup>1</sup> 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 information for more information.

<sup>2</sup> To acquire a program device using the ILE FORTRAN/400 language, you must specify the program device on the ACQPGMDEV parameter on the CRTICFF, CHGICFF, or OVRICFF commands. The program device will then be implicitly acquired when the ICF file is opened.



---

## Supported DDS Keywords

Following is a list of data description specifications (DDS) keywords that you can use to specify the communications functions for BSCCL.

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

The following list presents the system-supplied formats that you use to specify the communications functions 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 *ICF Programming* book.



---

## Appendix B. Return Codes, Messages, and Sense Codes

---

### Return Codes

This section 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; they report the results of each I/O operation issued by your application program. Your program should check the return code and act accordingly. Refer to your high-level language book for more information on how to access these return codes.

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

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

#### Notes:

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

---

### Major Code 00

**Major Code 00** – Operation completed successfully.

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

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

Code	Description/Action
0000	<p><b>Description:</b> The last operation issued by your program completed successfully.</p> <p><b>Action:</b> For the actions which can be taken after 0000 is received, refer to the following table:</p>

Figure B-1. Actions for Return Code 0000

Type of Session	Last Operation Issued	Actions Your Program Can Take
Started by a source program	Acquire or open	Issue an evoke <sup>1</sup> or timer function, or a get-attributes operation.
	Evoke with detach or write with detach	Issue another evoke function, issue a release operation, continue local processing, or end.
	Any other output operation	Issue another output operation (except evoke), or issue an input operation.
	End-of-Session	Continue local processing or end.
Started by a remote program start request <sup>2</sup>	Acquire or open	Issue an input or output operation.
	Write with detach	Continue local processing or end. This session has ended.
	Any other output operation	Issue another output operation (except evoke), or issue an input operation.
	End-of-Session	Continue local processing or end.

<sup>1</sup> 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.

<sup>2</sup> A target program (started by a program start request) cannot issue an evoke function in this session; it can issue an evoke function only in a different session that it has first acquired.

- 0001**     **Description:** On a successful input operation, your program received some data. Your program must continue to receive data until it receives an end-of-transmission indication (which allows your program to send data) or a detach indication.
- Action:** Issue another input operation. If your program detects an end-of-transmission indication, it can issue an output operation.
- 0008**     **Description:** On a successful input operation, your program received a detach indication with the last of the data. The communications transaction with the remote program has ended, but the session with the remote system is still active.
- Action:** If your program started the session, it can issue another evoke function (to start another program), issue a release operation (to perform local processing or to start another session), or end. If a program start request from the remote program started the transaction, your program can either issue an end-of-session function or end.
- 0010**     **Description:** On a successful output operation, your program received a reverse-interrupt indication. The remote program wants to send data as soon as possible. You should allow the remote program to send this data.
- Action:** Issue an input operation as soon as possible.
- 0020**     **Description:** On a successful input operation, your program received a remote system message and an end-of-transmission indication. The message is in your program's input buffer and is a result of the pre-

vious unsuccessful evoke function, for which your program received return code 831A.

**Action:** Handle the message in the input buffer (for example, display it). Your program now has control of the session, and can issue another evoke function (to start another program), issue a release operation (to perform local processing or to start another session), or end.

**0021 Description:** On a successful input operation, your program received a remote system message which is now in your program's input buffer. Your program should continue to receive input.

**Action:** Handle the message in the input buffer (for example, display it), and issue another input operation. If your program detects the equivalent of an end-of-transmission indication, it can issue an output operation.

**0028 Description:** On a successful input operation, your program received a detach indication with a remote system message. The communications transaction with the remote program has ended, but the session with the remote system is still active. The system message is in your program's input buffer and describes the status of the transaction that has ended.

**Action:** Handle the message in the input buffer (for example, display it). If your program started the session, it can issue another evoke function (to start another program), issue a release operation (to perform local processing or to start another session), or end. If a program start request from the remote program started the transaction, your program can either issue an end-of-session function or end.

**0030 Description:** On a successful input operation, your program received a truncated remote system message and an end-of-transmission indication. The message is in your program's input buffer, and was truncated because it was too long for the buffer. The message is a result of the previous unsuccessful evoke function, for which your program received return code 831A.

**Action:** Handle the message in the input buffer (for example, display it). Your program now has control of the session, and can issue another evoke function (to start another program), issue a release operation (to perform local processing or to start another session), or end.

**0031 Description:** On a successful input operation, your program received a truncated remote system message. The message is in your program's input buffer, and was truncated because it was too long for the buffer. Your program should continue to receive input.

**Action:** Handle the message in the input buffer (for example, display it), and issue another input operation. If your program detects the equivalent of an end-of-transmission indication, it can issue an output operation.

**0038 Description:** On a successful input operation, your program received a detach indication with a truncated remote system message. The communications transaction with the remote program has ended, but the session with the remote system is still active. The message is in your program's input buffer and describes the status of the transaction

that has ended. The message was truncated because it was too long for the buffer.

**Action:** Handle the message in the input buffer (for example, display it). If your program started the session, it can issue another evoke function (to start another program), issue a release operation (to perform local processing or to start another session), or end. If a program start request from the remote program started the transaction, your program can either issue an end-of-session function or end.

---

## 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	Description/Action
0201	<p><b>Description:</b> On a successful input operation, your program received some data. Also, your job is being ended (controlled). Your program can continue to receive data until it receives an end-of-transmission indication (which allows your program to send data) or a detach indication.</p> <p><b>Action:</b> Your program can issue another input operation. If your program detects the equivalent of an end-of-transmission indication, it can issue an output operation. However, the recommended action is to complete all processing and end your program as soon as possible. The system eventually changes a job ended (controlled) to a job ended (immediate) and forces all processing to stop for your job.</p>
0208	<p><b>Description:</b> On a successful input operation, your program received a detach indication with the last of the data. The communications transaction with the remote program has ended, but the session with the remote system is still active. Also, your job is being ended (controlled).</p>

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

**0220**

**Description:** On a successful input operation, your program received a remote system message and an end-of-transmission indication. The message is in your program's input buffer and is a result of the previous unsuccessful evoke function, for which your program received return code 831A. Also, your job is being ended (controlled).

**Action:** Handle the message in the input buffer (for example, display it). Your program now has control of the session, and can issue another evoke function (to start another program), issue a release operation (to perform local processing or to start another session), or end. However, the recommended action is to complete all processing and end your program as soon as possible. The system eventually changes a job ended (controlled) to a job ended (immediate) and forces all processing to stop for your job.

**0221**

**Description:** On a successful input operation, your program received a remote system message which is now in your program's input buffer. Also, your job is being ended (controlled). Your program should continue to receive input.

**Action:** Handle the message in the input buffer (for example, display it), and issue another input operation. If your program detects the equivalent of an end-of-transmission indication, it can issue an output operation. However, the recommended action is to complete all processing and end your program as soon as possible. The system eventually changes a job ended (controlled) to a job ended (immediate) and forces all processing to stop for your job.

**0228**

**Description:** On a successful input operation, your program received a detach indication with a remote system message. The communications transaction with the remote program has ended, but the session with the remote system is still active. The system message is in your program's input buffer and describes the status of the transaction that has ended. Also, your job is being ended (controlled).

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

- 0230**      **Description:** On a successful input operation, your program received a truncated remote system message and an end-of-transmission indication. The message is in your program's input buffer, and was truncated because it was too long for the buffer. The message is a result of the previous unsuccessful evoke function, for which your program received return code 831A. Also, your job is being ended (controlled).
- Action:** Handle the message in the input buffer (for example, display it). Your program now has control of the session, and can issue another evoke function (to start another program), issue a release operation (to perform local processing or to start another session), or end. However, the recommended action is to complete all processing and end your program as soon as possible. The system eventually changes a job ended (controlled) to a job ended (immediate) and forces all processing to stop for your job.
- 0231**      **Description:** On a successful input operation, your program received a truncated remote system message. The message is in your program's input buffer, and was truncated because it was too long for the buffer. Also, your job is being ended (controlled). Your program should continue to receive input.
- Action:** Handle the message in the input buffer (for example, display it), and issue another input operation. If your program detects the equivalent of an end-of-transmission indication, it can issue an output operation. However, the recommended action is to complete all processing and end your program as soon as possible. The system eventually changes a job ended (controlled) to a job ended (immediate) and forces all processing to stop for your job.
- 0238**      **Description:** On a successful input operation, your program received a detach indication with a truncated remote system message. The communications transaction with the remote program has ended, but the session with the remote system is still active. The message is in your program's input buffer and describes the status of the transaction that has ended. The message was truncated because it was too long for the buffer. Also, your job is being ended (controlled).
- Action:** Handle the message in the input buffer (for example, display it). If your program started the session, it can issue another evoke function (to start another program), issue a release operation (to perform local processing or to start another session), or end. If a program start request from the remote program started the transaction, your program can either issue an end-of-session function or end. However, the recommended action is to complete all processing and end your program as soon as possible. The system eventually changes a job ended (controlled) to a job ended (immediate) and forces all processing to stop for your job.



---

## Major Code 03

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

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

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

Code	Description/Action
<b>0300</b>	<p><b>Description:</b> On a successful input operation, your program received an end-of-transmission indication without any data. The session is still active. If you specified GRPSEP(*DEV3740) for this session, 0300 indicates that the last file has been received. If you specified GRPSEP(*OFCSYS) for this session, 0300 indicates that the last document has been received. If you specified neither, 0300 indicates that the last record in the file has been received. If you specified the DDS keyword RCVTRNRND, the receive-turnaround response indicator is also set on.</p> <p><b>Action:</b> Issue an input or output operation, or end the transaction using a write operation with a detach function.</p>
<b>0301</b>	<p><b>Description:</b> On a successful input operation, your program received no data. If you specified GRPSEP(*DEV3740) for this session, 0301 indicates that the last record in a file (a null record) has been received. If you specified GRPSEP(*OFCSYS) for this session, 0301 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. Your program must continue to receive input until it receives an end-of-transmission or detach indication.</p> <p><b>Action:</b> Issue an input operation. If your program detects an end-of-transmission indication, it can issue an output operation.</p>
<b>0308</b>	<p><b>Description:</b> On a successful input operation, one of the following occurred:</p> <ol style="list-style-type: none"><li>1. When RMTBSCSEL(*YES) is specified for the session, your program received a detach indication without any data. The communications transaction with the remote program has ended, but the session with the remote system is still active. If you specified the DDS keyword RCVDETACH, the receive-detach indicator is also set on.</li><li>2. When RMTBSCSEL(*NO) was specified, BSCSEL received a disconnect indication (switched lines only). Communications has ended with the program on the remote system.</li></ol> <p><b>Action:</b> If your program started the session, and</p> <ol style="list-style-type: none"><li>1. You specified RMTBSCSEL(*YES), it can issue another evoke function (to start another program), issue a release operation (to perform local processing or to start another session), or end. If a program start request from the remote program started the trans-</li></ol>

action, your program can either issue an end-of-session function or end.

2. You specified RMTBSCEL(\*NO), use a release operation, an end-of-session function, or end your program.

**0309**      **Description:** On a read-from-invited-program-devices operation, your program did not receive any data. Also, your job is being ended (controlled).

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

**Messages:**

CPF4741 (Notify)

**0310**      **Description:** On a read-from-invited-program-devices operation, the time interval specified by a timer function in your program or by the WAITRCD value specified for the ICF file expired.

**Action:** Issue the intended operation after the specified time interval has ended. For example, if you were using the time interval to control the length of time to wait for data, you can issue another read-from-invited-program-devices operation to receive the data.

**Note:** Since no specific program device name is associated with the completion of this operation, the program device name in the common I/O feedback area is set to \*N. Therefore, your program should not make any checks based on the program device name after receiving the 0310 return code.

**Messages:**

CPF4742 (Status)

CPF4743 (Status)

---

## 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, or 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.

<b>Code</b>	<b>Description/Action</b>
<b>0411</b>	<p><b>Description:</b> 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 issue an output operation.</p> <p><b>Action:</b> Issue an input operation to receive the message.</p> <p><b>Note:</b> If your program issues another output operation before an input operation, your program receives a return code of 831C.</p> <p><b>Messages:</b></p> <p>CPF4705 (Notify)</p>
<b>0412</b>	<p><b>Description:</b> An output exception occurred because your program attempted to send data when it should be receiving data that was sent by the remote program, or because it attempted to cancel an invite function when it should be receiving data, a message, or a return code. The data from your output operation was not sent to the remote system. Your program can attempt to send the data later.</p> <p><b>Action:</b> Issue an input operation to receive the data, message, or return code.</p> <p><b>Note:</b> If your program issues another output operation before an input operation, your program receives a return code of 831C.</p> <p><b>Messages:</b></p> <p>CPF4702 (Notify) CPF4705 (Notify) CPF4750 (Notify) CPF4799 (Notify) CPF5076 (Notify)</p>

---

## Major Codes 08 and 11

**Major Codes 08 and 11** – Miscellaneous program errors occurred.

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

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

<b>Code</b>	<b>Description/Action</b>
<b>0800</b>	<p><b>Description:</b> The acquire operation just attempted by your program was not successful. Your program tried to acquire a program device that was already acquired and is still active.</p> <p><b>Action:</b> If the session associated with the original acquire operation is the one needed, your program can begin communicating in that session since it is already available. If you want a different session, issue another acquire operation for the new session by specifying a different program device name in the PGMDEV parameter of the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command that precedes the program.</p> <p><b>Messages:</b></p> <ul style="list-style-type: none"><li>CPD4077 (Diagnostic)</li><li>CPF5041 (Status)</li><li>CPF50A0 (Status)</li></ul>
<b>1100</b>	<p><b>Description:</b> The read-from-invited-program-devices operation just attempted by your program was not successful because your program tried this operation when no program devices were invited and no timer function was in effect.</p> <p><b>Action:</b> Issue an invite function (or a combined operation that includes an invite) followed by a read-from-invited-program-devices operation.</p> <p><b>Messages:</b></p> <ul style="list-style-type: none"><li>CPF4740 (Notify)</li></ul>

## Major Code 34

**Major Code 34** – Input exception occurred.

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

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

<b>Code</b>	<b>Description/Action</b>
<b>3401</b>	<p><b>Description:</b> The input operation issued by your program was not successful because the length of the data record sent by the remote system was longer than the length specified for your program's input buffer. The length of the data record received from the remote system, if available, is in the actual-record-length field in the I/O feedback area.</p> <p><b>Action:</b> Issue another input operation if your program can specify a record size large enough to receive the data, plus any indicators for a file without a separate indicator area. Otherwise, you should close the file, end your program, correct the record size, then run your program again.</p> <p><b>Messages:</b></p> <p>CPF4768 (Notify) CPF5319 (Notify)</p>
<b>3441</b>	<p><b>Description:</b> A valid record format name was specified with format selection type *RECID. However, although the data received matched one of the record formats in the ICF file, it did not match the format specified on the read operation.</p> <p><b>Action:</b> Correct your program to issue a read operation that does not specify a record format name, or specify the correct record format name to process the data based on the format selection option for the file.</p> <p><b>Messages:</b></p> <p>CPF5058 (Notify)</p>
<b>3451</b>	<p><b>Description:</b> Your program specified a file record size that was not large enough for the indicators to be included with the data sent by the remote program (for a file defined with a nonseparate indicator area). Your program did not receive any data. For a file using a nonseparate indicator area, the actual record length field in the device-dependent I/O feedback area contains the number of indicators specified by the record format.</p>

**Action:** End the session; close the file; correct the file record size; then open the file again.

**Messages:**

CPF4768 (Notify)

---

## Major Code 80

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

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

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

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

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

Code	Description/Action
8081	<p><b>Description:</b> The operation attempted by your program was not successful because a system error condition was detected.</p> <p><b>Action:</b> Your communications configurations may need to be varied off and then on again. Your program can do one of the following:</p> <ul style="list-style-type: none"><li>• Continue local processing.</li><li>• Close the ICF file, open the file again, and establish the session again.</li><li>• End.</li></ul> <p><b>Messages:</b></p> <p>CPF4170 (Escape) CPF4510 (Escape) CPF4562 (Escape) CPF4582 (Escape) CPF4601 (Escape) CPF4602 (Escape) CPF4603 (Escape) CPF4701 (Diagnostic) CPF4707 (Diagnostic) CPF4708 (Diagnostic) CPF4709 (Diagnostic) CPF4710 (Diagnostic) CPF4711 (Diagnostic)</p>

CPF4712 (Diagnostic)  
CPF4713 (Diagnostic)  
CPF4714 (Diagnostic)  
CPF4715 (Diagnostic)  
CPF4716 (Diagnostic)  
CPF4717 (Diagnostic)  
CPF4722 (Diagnostic)  
CPF4731 (Diagnostic)  
CPF4732 (Diagnostic)  
CPF5105 (Escape)  
CPF5257 (Escape)  
CPF5351 (Escape)  
CPF5352 (Escape)  
CPF5353 (Escape)  
CPF5355 (Escape)  
CPF5410 (Escape)  
CPF5411 (Escape)  
CPF5447 (Escape)  
CPF5451 (Escape)

**8082**

**Description:** The operation attempted by your program was not successful because the device associated with the remote location you are using is not usable. For example, this may have occurred because communications were stopped for the device by a Hold Communications Device (HLDCMNDEV) command, or because a cancel reply was issued in response to an error recovery message for the device. Your program should not issue any operations to the device.

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

**Messages:**

CPF4566 (Escape)  
CPF4744 (Escape)  
CPF5269 (Escape)  
CPF5274 (Escape)  
CPF5358 (Escape)

**80B3**

**Description:** The open operation issued by your program was not successful because the ICF file is in use by another process.

**Action:** Wait for the file to become available, then issue another open operation. Otherwise, your program may continue processing, or it can end.

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

**Messages:**

CPF4128 (Escape)

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

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

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

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

**Messages:**

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

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

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

**Messages:**

CPF4131 (Escape)

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

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



**Messages:**

CPF4104 (Escape)

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

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

**Action:**

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

**Messages:**

CPF4132 (Escape)

CPF5129 (Escape)

## Major Code 81

**Major Code 81** – Permanent session error (irrecoverable).

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

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

If your program initiated the session, you can:

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

If your session was initiated by a program start request from the 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.

<b>Code</b>	<b>Description/Action</b>
<b>810A</b>	<p><b>Description:</b> On an unsuccessful operation, BSCCL detected a combination of values that was not valid. ASCII code was specified in the line description, but transparency was specified in the device description or on the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command. The two values are not valid together. The session has ended.</p> <p><b>Action:</b> If you want to use transparency, change the value of the CODE parameter from ASCII to EBCDIC on the CHGLINBSC command. If you do not want to use transparency, change the value of the TRNSPY parameter from *YES to *NO on the CHGDEVBSC command or in the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command. After you correct the error, run your program again.</p> <p><b>Messages:</b></p> <p style="padding-left: 40px;">CPF5511 (Escape)</p>
<b>8140</b>	<p><b>Description:</b> A cancel reply was received from your program or from the operator in response to a notify message, or was the result of a system default, causing the session to be ended. The session is no longer active.</p> <p><b>Action:</b> If your program started the session, issue an acquire operation to start the session again. If your program was started by a program start request, it can continue local processing or end.</p> <p><b>Messages:</b></p> <p style="padding-left: 40px;">CPF5104 (Escape)</p>
<b>8187</b>	<p><b>Description:</b> On an unsuccessful input or output operation, BSCCL detected a combination of values that was not valid. Either the record length or the block length specified in the device description or on the ADDICFDEVE, CHGICFDEVE, 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.</p> <p><b>Action:</b> If the record length and block length are correct for your program, change the MAXBUFFER parameter on the CHGLINBSC command. If the record length or block length are not correct, change the RCDLEN or BLKLEN parameter on the CHGDEVBSC command or on the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command. After you correct the error, run your program again.</p> <p><b>Messages:</b></p> <p style="padding-left: 40px;">CPF5232 (Escape)</p>
<b>8191</b>	<p><b>Description:</b> A permanent line or controller error occurred on an output operation, and the system operator attempted recovery in response to the error message. You can learn what type of line error occurred by checking the system operator's message queue. The session has ended. Data may have been lost.</p> <p><b>Action:</b> If your program started the session, issue an acquire operation to start the session again. If your program was started by a</p>

program start request from the remote program, it can continue local processing or end.

**Messages:**

CPF4146 (Escape)  
CPF4707 (Diagnostic)  
CPF4708 (Diagnostic)  
CPF4709 (Diagnostic)  
CPF4710 (Diagnostic)  
CPF4711 (Diagnostic)  
CPF4713 (Diagnostic)  
CPF4714 (Diagnostic)  
CPF4715 (Diagnostic)  
CPF4716 (Diagnostic)  
CPF4717 (Diagnostic)  
CPF4718 (Diagnostic)  
CPF4722 (Diagnostic)  
CPF5128 (Escape)  
CPF5138 (Escape)  
CPF5351 (Escape)

**8192**      **Description:** A permanent line or controller error occurred on an input operation, and the system operator took a recovery option in response to the error message. You can learn what type of line error occurred by checking the system operator's message queue. The session has ended. Data may have been lost.

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

**Messages:**

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

**8193**      **Description:** On an output operation (for switched lines only), your program received a disconnect indication, or the switched connection failed. 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 data that was not valid. The session has ended.

**Action:** If the switched connection failed, contact the remote system to determine why the disconnect indication was sent. Otherwise, verify that your program did not cause a time-out and that it did not send data that was not valid. Also, verify that it did not try to send data after the transaction had ended. If your program started the session, issue an

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

**Messages:**

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

**8194**

**Description:** On an input operation (for switched lines only), your program received a disconnect indication, or the switched connection failed. 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, contact the remote system to determine why the disconnect indication was sent. Otherwise, verify that your program did not cause a time-out. Also, verify that it did not try to receive data after the transaction had ended. If your program started the session, issue an acquire operation to start the session again. If your program was started by a program start request from the remote program, it can continue local processing or end.

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

**Messages:**

CPF4712 (Diagnostic)  
CPF5351 (Escape)

**8198**

**Description:** On an input 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, issue an acquire operation to start the session again. If your program was started by a program start request from the remote program, it can continue local processing or end.

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

**Messages:**

CPF4717 (Diagnostic)  
CPF5351 (Escape)

**819A** **Description:** On an input 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 by, or received from, 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 input operations. If your program started the session, issue an acquire operation to start the session again. If your program was started by a program start request from the remote program, it can continue local processing or end.

**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 remote program is correct. If your program started the session, issue an acquire operation to start the session again. If your program was started by a program start request from the remote program, it can continue local processing or end.

**Messages:**

CPF5350 (Escape)

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

- BSCCL received unexpected data from the remote program after your program received a detach indicator or before your program used an evoke function.
- BSCCL received an unexpected program start request from the remote program while your session was still active.

- BSCCL received a command that was not valid from the remote system while your session was still active.
- BSCCL 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 remote program did not contain a BSCCL command. Also, check to see if the remote program sent a program start request while your session was still active. If your program started the session, issue an acquire operation to start the session again. If your program was started by a program start request from the remote program, it can continue local processing or end.

**Messages:**

CPF5306 (Escape)

**81E9**

**Description:** An input operation was issued and the format selection option for the ICF file was \*RECID, but the data received did not match any record formats in the file. There was no format in the file defined without a RECID keyword, so there was no default record format to use. The session has ended.

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

**Messages:**

CPF5291 (Escape)

## Major Code 82

**Major Code 82** – Open or acquire operation failed.

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

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

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

- Correct the problem and attempt to establish the session again. The next operation could be successful only if the error occurred because of some temporary condition 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.



<b>Code</b>	<b>Description/Action</b>
<b>8209</b>	<p><b>Description:</b> The open or acquire operation issued by your program was not successful because a prestart job is being canceled. One of the following may have occurred:</p> <ul style="list-style-type: none"> <li>• An End Job (ENDJOB), End Prestart Job (ENDPJ), End Subsystem (ENDSBS), End System (ENDSYS), or Power Down System (PWRDWNSYS) command was being issued.</li> <li>• The maximum number of prestart jobs (MAXJOBS parameter) was reduced by the Change Prestart Job Entry (CHGPJE) command.</li> <li>• The value for the maximum number of program start requests allowed (specified in the MAXUSE parameter on the ADDPJE or CHGPJE command) was exceeded.</li> <li>• Too many unused prestart jobs exist.</li> <li>• The prestart job had an initialization error.</li> </ul> <p><b>Action:</b> Complete all processing and end your program as soon as possible. Correct the system error before starting this job again.</p> <p><b>Messages:</b></p> <p style="padding-left: 40px;">CPF4292 (Escape) CPF5313 (Escape)</p>
<b>820A</b>	<p><b>Description:</b> On an unsuccessful open or acquire operation, BSCCEL detected one of the following incorrect combinations of values:</p> <ul style="list-style-type: none"> <li>• 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, CHGICFDEVE, or OVRICFDEVE command.</li> <li>• BLOCK(*USER) and RMTBSCCEL(*YES) were both specified.</li> </ul> <p>The session was not started.</p> <p><b>Action:</b></p> <ul style="list-style-type: none"> <li>• If you want to use transparency, change the value of the CODE parameter from ASCII to EBCDIC on the CHGLINBSC command. If you do not want to use transparency, change the value of the TRNSPY parameter from *YES to *NO on the CHGDEVBSC command, or on the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command.</li> <li>• If you want to use RMTBSCCEL(*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 RMTBSCCEL parameter from *YES to *NO.</li> </ul> <p>Following one of these changes, your program can try the open or acquire operation again.</p> <p><b>Messages:</b></p> <p style="padding-left: 40px;">CPF4303 (Escape) CPF5511 (Escape)</p>
<b>8233</b>	<p><b>Description:</b> A program device name that was not valid was detected. Either an ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command was not run, or the program device name in your program does not</p>

match the program device name specified in the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command for the session being acquired. The session was not started.

**Action:** If the error was in your program, change your program to specify the correct program device name. If an incorrect identifier was specified in the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command, specify the correct value in the PGMDEV parameter.

**Messages:**

CPF4288 (Escape)  
CPF5068 (Escape)

**8281**

**Description:** On an unsuccessful open or acquire operation, a system error condition was detected. For example, the file may previously have been in error, or the file could not be opened due to a system error.

**Action:** Your communications configurations may need to be varied off and then on again. Your program can do one of the following:

- Continue local processing.
- Close the ICF file, open the file again, and acquire the program device again. However, if this results in another 8281 return code, your program should close the file and end.
- Close the file and end.

**Messages:**

CPF4168 (Escape)  
CPF4182 (Escape)  
CPF4221 (Escape)  
CPF4304 (Escape)  
CPF4369 (Escape)  
CPF4370 (Escape)  
CPF4375 (Escape)  
CPF5105 (Escape)  
CPF5257 (Escape)  
CPF5274 (Escape)  
CPF5317 (Escape)  
CPF5318 (Escape)  
CPF5355 (Escape)

**8282**

**Description:** The open or acquire operation attempted by your program was not successful because the device associated with the remote location you are using is not usable. For example, this may have occurred because communications were stopped for the device by a Hold Communications Device (HLDCMNDEV) command, or because a cancel reply was issued in response to an error recovery message for the device. Your program should not issue any operations to the device. The session was not started.

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

**Messages:**

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

**8287**      **Description:** On an unsuccessful open or acquire operation, BSCEL detected a combination of values that was not valid. The value for either the record length or the block length parameter in the device description or in the ADDICFDEVE, CHGICFDEVE, 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, change the value of the MAXBUFFER parameter on the CHGLINBSC command. If the values are not correct, change the value of the RCDLEN or BLKLEN parameter on the CHGDEVBSC command or on the ADDICFDEVE, CHGICFDEVE, 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 a combination of values that was not valid. Both a record separator and transparency were specified in the device description or on the ADDICFDEVE, CHGICFDEVE, 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) or on the ADDICFDEVE, CHGICFDEVE, 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 a combination of values that was not valid. The maximum user record length was specified to be greater than the block length in the device description or on the ADDICFDEVE, CHGICFDEVE, 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 in the device description (CHGDEVBSC command) or on the ADDICFDEVE, CHGICFDEVE, 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 a combination of values that was not valid. Both 3740 multiple file format and intermediate text block (ITB) blocking were specified in the device description or on the ADDICFDEVE, CHGICFDEVE, 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) or on the ADDICFDEVE, CHGICFDEVE, 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, BSCEL detected a combination of values that was not valid. Both blank compression and intermediate text block (ITB) blocking were specified in the device description or on the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command. The session was not started.

**Action:** If you want blank compression, change the value of the BLOCK parameter in the device description (CHGDEVBSC command) or on the ADDICFDEVE, CHGICFDEVE, 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, BSCEL detected one of the following incorrect combinations of values:

- Both TRUNC(\*YES) and BLOCK(\*ITB) were specified in the device description or on the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command.
- Both TRUNC(\*YES) and BLOCK(\*NOSEP) were specified in the device description or on the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command.

The session was not started.

**Action:** If you want blank truncation, change the value of the BLOCK parameter in the device description (CHGDEVBSC command) or on the ADDICFDEVE, CHGICFDEVE, 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, BSCCL detected a combination of values that was not valid. Both blank compression and transparency were specified in the device description or on the ADDICFDEVE, CHGICFDEVE, 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 in the device description (CHGDEVBSC command) or on the ADDICFDEVE, CHGICFDEVE, 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 or controller error occurred on an unsuccessful open or acquire operation, and the system operator took a recovery option in response to the error message. The session was not started.

**Action:** If your program was attempting to start the session, it can try the acquire operation again. If your program was started by a program start request from the remote program, your program can continue local processing or end.

**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:** On an unsuccessful open or acquire operation (for switched lines only), your program received a disconnect indication, or the switched connection failed. Either the switched connection was not established, or the line was unexpectedly disconnected. The session was not started.

**Action:** Contact the remote system to determine why the disconnect indication was sent. Your program can try the open or acquire operation again, continue local processing, or end.

**Messages:**

CPF4701 (Diagnostic)  
CPF5260 (Escape)

CPF5351 (Escape)

**8297**     **Description:** On an unsuccessful open or acquire operation, the session was not started because the remote system has ended the line transmission.

**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 a record separator character that was not valid. A record separator character that was not valid was specified on the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command. The session was not started.

**Action:** Change the value of the record separator character in the BLOCK parameter on the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command. Try the open or acquire operation again.

**Messages:**

CPF4302 (Escape)  
CPF5510 (Escape)

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

**Action:** Your program can wait for the program device to become available, then try the open or acquire operation again. Otherwise, it can continue local processing or end.

**Messages:**

CPF4106 (Escape)  
CPF5507 (Escape)

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

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

**Messages:**

CPF4745 (Diagnostic)  
CPF5041 (Status)

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

- Your program has already acquired the \*REQUESTER device.
- The job was started by a program start request with the \*REQUESTER device detached.

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

**Action:**

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

**Messages:**

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

**82AA**

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

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

**Messages:**

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

**82AB**

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

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

**Messages:**

CPF4285 (Escape)  
CPF5333 (Escape)

**82AC**

**Description:** The open or acquire operation attempted by your program was not successful because the remote location name specified for this device was incorrect. The value \*REQUESTER must be specified as the remote location name when the target program is started by a program start request. The session was not started.

**Action:** Close the ICF file. Specify the value \*REQUESTER for the remote location name (RMTLOCNAME) parameter on the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command. After the error is corrected, your target program can be started by another program start request.

**Messages:**

CPF411C (Escape)  
CPF511A (Escape)

**82B3**

**Description:** The open or acquire operation attempted by your program was not successful because your program is trying to use a device description that is already in use by another job. The session was not started.

**Action:** Wait for the device description to become available, then issue the acquire operation again. You can use the Work with Configuration Status (WRKCFGSTS) command to determine which job is using the device description. Consider increasing the WAITFILE parameter of the CHGICFF or OVRICFF command to allow more time for the device to become available. Otherwise, your program can continue local processing or end.

**Messages:**

CPF4282 (Escape)  
CPF5332 (Escape)

**82EA**

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

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

**Messages:**

CPF4348 (Escape)  
CPF5521 (Escape)



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

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

**Messages:**

CPF4105 (Escape)  
CPF4223 (Escape)  
CPF4251 (Escape)  
CPF4760 (Escape)  
CPF5038 (Escape)  
CPF5550 (Escape)

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

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

**Messages:**

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

**82F5**      **Description:** The open or acquire operation was not successful because your program tried to use a format selection option of \*RMTFMT in the FMTSLT parameter on the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command. The session was not started.

**Action:** Change the value in the FMTSLT parameter on the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command, then issue 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:** A session error occurred, but the session may still be active. Recovery within your program might be possible.

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

- Correct the problem and continue processing with the session. If the error occurred because of a resource failure on the remote system 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	Description/Action
<b>830B</b>	<p><b>Description:</b> Your program attempted an operation that was not valid because the session was not yet acquired or has ended. The session may have ended because of a release operation, an end-of-session function, or a permanent error. Your program may have incorrectly handled a previous error.</p> <p><b>Action:</b> Verify that your program does not attempt any operations without an active session. Also verify that your program correctly handles the permanent error or session-not-acquired return codes (80xx, 81xx, 82xx) it received on previously issued input and output operations. To recover from an incorrectly handled error condition, your program may or may not be able to issue another acquire operation, depending on the return code.</p>

**Messages:**

CPD4079 (Diagnostic)  
CPF4739 (Status)  
CPF5067 (Escape)  
CPF5068 (Escape)  
CPF5070 (Escape)

**831A**     **Description:** The evoke function attempted by your program was not successful due to an error on the remote system. The remote system sent a message indicating the type of error that occurred.

**Action:** Issue an input operation to receive the message.

**Messages:**

CPF4796 (Notify)

**831C**     **Description:** Your program's previous output operation received a return code of 0411 or 0412, indicating that your program must receive information sent by the remote system or BSCEL; however, your program did not handle the return code correctly. The current output operation was not successful because your program should have issued an input operation to receive the information already sent by the remote system.

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

**Messages:**

CPF4934 (Notify)

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

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

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

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

**Messages:**

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

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

- On an output operation, your program tried to send a data record that was longer than the MAXRCDLEN value specified for the ICF file, or longer than the RCDLEN value specified on the CRTDEVBSC, CHGDEVBSC, ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command.
- If this was an evoke function, one of the following occurred:
  - The combined length of the specified target program name, blank separator(s), and user-defined parameters exceeded the maximum of 120 bytes available in positions 7 through 126 of the program start request.
  - The target program name was not specified or was longer than 8 characters.
  - 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 (VARLEN) function was used, then the length of the user buffer was not valid.

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

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

**Messages:**

CPF4762 (Notify)

CPF4765 (Notify)

CPF4767 (Notify)

CPF4797 (Notify)

**8322** **Description:** Your program tried to issue a request-to-write function. This function is only valid while your program is in receive state.

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

**Messages:**

CPF4703 (Notify)

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

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

**Messages:**

CPF5098 (Notify)

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

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

**Messages:**

CPF5099 (Notify)

**832B** **Description:** Your program issued an output operation that was not valid 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.

**Messages:**

CPF4798 (Notify)

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

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

**Messages:**

CPF4769 (Notify)

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

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

**Messages:**

CPF4924 (Notify)

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

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

**Messages:**

CPF4801 (Notify)

CPF5099 (Notify)

**8334** **Description:** The evoke function attempted by your program was not valid. Your program used an evoke function without a target program name, or the target program name was longer than 8 characters.

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

**Messages:**

CPF4797 (Notify)

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

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

**Messages:**

CPF5054 (Notify)

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

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

The session is still active.

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

**Messages:**

CPF4763 (Notify)

**83F6 Description:** On an unsuccessful output operation, your program attempted to send some user-defined data that was not valid. This was due to one of the following reasons:

- ASCII code was specified in the line description, but the data record contains a character that cannot be translated into ASCII.
- BLOCK(\*USER) was specified in the device description or on the ADDICFDEVE, CHGICFDEVE, or OVRICFDEVE command, but a starting or ending character that was not valid was found in a user-blocked data record.

The session is still active.

**Action:** Correct the data record sent by your program, then try the output operation again.

**Messages:**

CPF4706 (Notify)

**83F7 Description:** On an unsuccessful output operation, your program attempted to send a user-blocked data record that had a length of fewer than 2 characters. A length of at least 2 characters is necessary for the starting and ending character pair required for a BSC record.

**Action:** Correct the data record sent by your program, then try the output operation again.

**Messages:**

CPF4718 (Notify)

**83F8**

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

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

**Messages:**

CPF5293 (Escape)



## 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.

Figure 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
Interchange group 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

---

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

---

<b>Name</b>	<b>Control Char- acter</b>	<b>ASCII</b>	<b>EBCDIC</b>
Transparent intermediate block	XITB		DLE IUS '101F'X
Transparent end-of-text	XETX		DLE ETX '1003'X
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. EBCDIC and ASCII Character Sets

The following charts show the EBCDIC and ASCII character sets. The charts are provided to show the data link control characters that are used in data communications.

### EBCDIC Character Set

Figure D-1 shows a complete EBCDIC character set.

		Main Storage Bit Positions 0,1,2,3																
Main Storage Bit Positions 4,5,6,7	Hex	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0000	0	NUL	DLE	DS		SP	&	-						{	}	\	0	
0001	1	SOH	DC1	SOS		RSP		/		a	j	~		A	J	NSP	1	
0010	2	STX	DC2	FS	SYN					b	k	s		B	K	S	2	
0011	3	ETX	DC3	WUS	IR					c	l	t		C	L	T	3	
0100	4	SEL	ENP RES	INP BYP	PP					d	m	u		D	M	U	4	
0101	5	HT	NL	LF	RS					e	n	v		E	N	V	5	
0110	6		BS	ETB	NBS					f	o	w		F	O	W	6	
0111	7	DEL	POC	ESC	EOT					g	p	x		G	P	X	7	
1000	8	GE	CAN		SBS					h	q	y		H	Q	Y	8	
1001	9	SPS	EM		IT					'	i	r	z		I	R	Z	9
1010	A	RPT	UBS	SM SW	RFF	¢	!		:					SHY				
1011	B	VT	CU1	FMT	CU3	.	\$	,	#									
1100	C	FF	IFS			<	*	%	@					⌋		⌈		
1101	D	CR	IGS	ENQ	NAK	(	) DC4	—	'									
1110	E	SO	IRS	ACK		+	;	>	=					⌋				
1111	F	SI	ITB IUS	BEL	SUB		⌋	?	"									EO

RV2P802-0

Figure D-1. EBCDIC Character Set

# ASCII Character Set

Figure D-2 shows the ASCII character set.

		Main Storage Bit Positions 0,1,2,3															
Main Storage Bit Positions 4,5,6,7	Hex	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	NUL	DLE	SP	0	@	P	'	p								
0001	1	SOH	DC1	!	1	A	Q	a	q								
0010	2	STX	DC2	"	2	B	R	b	r								
0011	3	ETX	DC3	#	3	C	S	c	s								
0100	4	EOT	DC4	\$	4	D	T	d	t								
0101	5	ENQ	NAK	%	5	E	U	e	u								
0110	6	ACK	SYN	&	6	F	V	f	v								
0111	7	BEL	ETB	'	7	G	W	g	w								
1000	8	BS	CAN	(	8	H	X	h	x								
1001	9	HT	EM	)	9	I	Y	i	y								
1010	A	LF	SUB	*	:	J	Z	j	z								
1011	B	VT	ESC	+	;	K	[	k	{								
1100	C	FF	FS	,	<	L	\	l									
1101	D	CR	GS	-	=	M	]	m	}								
1110	E	SO	RS	.	>	N	^	n	~								
1111	F	SI	US	/	?	O	_	o	DEL								

RV2P803-0

Figure D-2. ASCII Character Set

# Appendix E. Binary Synchronous Communications Configuration Example

This appendix contains a binary synchronous communications (BSC) configuration example for an AS/400 system attached to a System/36, using BSCCEL over a nonswitched line.

The example provided in this appendix shows an AS/400 BSC configuration matched to a configuration created on the System/36. It is not the intention of this appendix to show all possible methods of configuring these systems for binary synchronous communications.

## Configuring for BSCCEL Communications

This example shows the prompt displays that can be used to create BSCCEL configurations for a System/36 communicating with an AS/400 system.

## Configuring the System/36 for BSCCEL Communications

Line and subsystem members must be created on the System/36 for BSCCEL communications. This example shows the complete sequence of displays shown by running the CNFIGICF procedure to create a nonswitched line member (LINBSCCEL) and a BSCCEL subsystem member (SUBBSCCEL).

Several of the values specified in the System/36 line and subsystem members must be matched by values specified for the AS/400 line, controller, and device descriptions. These values are discussed under "Configuring the AS/400 System for BSCCEL Communications" on page E-2.

### Creating the System/36 Line Member:

The following displays show the CNFIGICF procedure for creating a nonswitched point-to-point line member called LINBSCCEL.

```
1.0          SSP-ICF CONFIGURATION MEMBER DEFINITION          W1
1. Configuration member name . . . . . LINBSCCEL
2. Library name . . . . . CNFIGLIB
3. Select one of the following:
   1. Create new member
   2. Edit existing member
   3. Create new member from existing member
   4. Remove a member
   5. Review a member
Option . . . . . 1-5 1

Cmd7-End      Cmd19-Cancel
```

```
2.0          SSP-ICF CONFIGURATION MEMBER TYPE          LINBSCCEL  W1
Select one of the following options:
   1. Intra
   2. BSC
   3. SNA
   4. Async
   5. PC Support/36
Option: 2
```

```
3.0          BSC CONFIGURATION MEMBER TYPE          LINBSCCEL  W1
1. BSC member type . . . . . 1,2 2
   1. BSC subsystem member
   2. BSC line member
2. Is this for MSRJE ? . . . . . Y,N  N
```

```
10.0         BSC LINE MEMBER ATTRIBUTES          LINBSCCEL  W1
1. Line type . . . . . 2
   1-Multipoint      2-Nonswitched pt-pt      3-Switched pt-pt
2. Error retry count . . . . . 1-255 007
```

### Creating the System/36 Subsystem Member:

The following displays show the CNFIGICF procedure for creating a BSCCEL subsystem member called SUBBSCCEL.

```
1.0          SSP-ICF CONFIGURATION MEMBER DEFINITION          W1
1. Configuration member name . . . . . SUBBSCCEL
2. Library name . . . . . CNFIGLIB
3. Select one of the following:
   1. Create new member
   2. Edit existing member
   3. Create new member from existing member
   4. Remove a member
   5. Review a member
Option . . . . . 1-5 1
```

```

2.0          SSP-ICF CONFIGURATION MEMBER TYPE          SUBBSCELE  W1

Select one of the following options:
  1. Intra
  2. BSC
  3. SNA
  4. Async
  5. PC Support/36

Option: 2

```

```

3.0          BSC CONFIGURATION MEMBER TYPE          SUBBSCELE  W1

1. BSC member type . . . . . 1,2  1
  1. BSC subsystem member
  2. BSC line member

2. Is this for MSRJE ? . . . . . Y,N  N

```

```

20.0         BSC SUBSYSTEM MEMBER SELECTION          SUBBSCELE  W1

1. Line member name . . . . . LINBSCELE

2. Select subsystem type from the following options:
  1. BSC CCP
  2. BSC CICS
  3. BSCELE
  4. BSC IMS/IRSS
  5. BSC 3270
  6. BSC MSRJE
Option . . . . . 1-6  3

```

```

22.0         SUBSYSTEM MEMBER DEFINITION            SUBBSCELE  W1

1. Remote location name . . . . . SUBBSCELE

2. Maximum user record length . . . . . 1-4075 1024

```

```

50.0         BSC SUBSYSTEM MEMBER ATTRIBUTES        SUBBSCELE  W1

1. EBCDIC/ASCII . . . . . 1-EBCDIC,2-ASCII  1
2. Transparency? . . . . . Y,N  N
3. Wait time in seconds . . . . . 1-999  999

```

```

52.0         SUBSYSTEM MEMBER DEFINITION            SUBBSCELE  W1

1. Block length . . . . . 0-4075  2048
2. Record separator in hexadecimal . . . . . 00
3. ITB mode? . . . . . Y,N  Y
4. Blank . . . . . 0-No,1-Compression,2-Truncation  0
5. 3740 multiple files? . . . . . Y,N  N
6. Office systems mode? . . . . . Y,N  N
7. Partner . . . . . 1-NORM,2-ATTR  1

Cmd5-Restart CNFIGICF      Cmd7-End
Cmd19-Cancel

COPR IBM Corp. 1986

```

## Configuring the AS/400 System for BSCLE Communications

The following prompt displays show the line, controller, and device descriptions created on the AS/400 system for BSCLE communications with a System/36 configured as shown under “Configuring the System/36 for BSCLE Communications” on page E-1. AS/400 configuration values that must match values specified in the System/36 CNFIGICF procedure are discussed following each of the prompt displays.

The following CL commands are used to create the AS/400 configuration:

- CRTLINBSC
- CRTCTLBSC
- CRTDEVBSCE

This example uses the prompt displays shown by typing the command name (such as CRTLINBSC) on the command line, then pressing F4 (Prompt).

**Creating the Line Description for BSCLE Communications:** The following displays show the values specified for the CRTLINBSC command to create a line description called LINBSCELE.

```

Create Line Desc (BSC) (CRTLINBSC)

Type choices, press Enter.

Line description . . . . . > LINBSCELE Name
Resource name . . . . . > LINB41 Name
Online at IPL . . . . . > *NO *YES, *NO
Application type . . . . . > *PCSA 1 *PGM, *RJE, *EML
Physical interface . . . . . > *RS232V24 *RS232V24, *V35, ...
Connection type . . . . . > *NONSMTTP 2 *NONSMTTP, *SMTTP, *MPTRIB
Switched network backup . . . . . *NO *NO, *YES
Duplex . . . . . *HALF *HALF, *FULL
Line speed . . . . . > 9600 600, 1200, 2400, 4800...
Modem type supported . . . . . *NORMAL *NORMAL, *V54, *IBMWRAP
Maximum buffer size . . . . . > 2048 0-8192
Character code . . . . . > *EBCDIC 3 *EBCDIC, *ASCII
Receive timer . . . . . > 30 30-254 (0.1 seconds)
Continue timer . . . . . > 20 16-24 (0.1 seconds)
Contention state retry . . . . . > 7 0-21
Data state retry . . . . . > 7 0-255

F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys

```

```

Create Line Desc (BSC) (CRTLINBSC)

Type choices, press Enter.

Transmit TTD or WACK retry . . . 60 0-65534, *NOMAX
Receive TTD or WACK retry . . . 45 0-65534, *NOMAX
Text 'description' . . . . . > 'Line description for BSCELE to S/36'

```

**Considerations for specifying the CRTLINBSC command:**

- 1 The value specified for the *Application type* (APPTYPE parameter) must be \*PGM for BSCCEL communications.
- 2 The value specified for the *Connection type* (CNN parameter) must match the *Line type* specified on display 10.0 of the System/36 CNFIGICF procedure.
- 3 The value specified for the *Maximum buffer size* (MAXBUFFER parameter) must be equal to or greater than the *Block length* specified on display 52.0 of the CNFIGICF procedure.
- 4 The value specified for the *Character code* (CODE parameter) must match the value specified for the *EBCDIC/ASCII* prompt on display 50.0 of the CNFIGICF procedure.
- 5 The values specified for the *Receive timer* (RCVTMR parameter) and *Continue timer* (CONTTMR parameter) must be compatible with the System/36 values. Because the System/36 values cannot be configured, the AS/400 default values should be used.

**Creating the Controller Description for BSCCEL Communications:** The following display shows the values specified for the CRTCTLBSC command to create a controller description called CTLBSCCEL.

```

Create Ctl Desc (BSC) (CRTCTLBSC)

Type choices, press Enter.

Controller description . . . . . > CTLBSCCEL      Name
Online at IPL . . . . . > *NO                    *YES, *NO
Connection type . . . . . > *NONSWTTPP 1  *NONSWTTPP, *SWTTPP, *MPTRIB
Switched network backup . . . . > *NO          *NO, *YES
Attached nonswitched line . . . . > CTLBSCCEL   Name
Application type . . . . . > *PGM 2          *PGM, *RJE, *EML
Text 'description' . . . . . > 'Controller description for BSCCEL to S/36'

Bottom
F3=Exit  F4=Prompt  F5=Refresh  F10=Additional parameters  F12=Cancel
F13=How to use this display  F24=More keys

```

**Considerations for specifying the CRTCTLBSC command:**

- 1 The value specified for the *Connection type* (CNN parameter) must match the *Line type* specified on display 10.0 of the System/36 CNFIGICF procedure.
- 2 The value specified for the *Application type* (APPTYPE parameter) must be \*PGM for BSCCEL communications.

**Creating the Device Description for BSCCEL Communications:** The following displays show the values specified for the CRTDEVBSCE command to create a device description called DEVBSCECEL.

```

Create Device Desc (BSC) (CRTDEVBSCE)

Type choices, press Enter.

Device description . . . . . > DEVBSCECEL      Name
Local location address . . . . . > 00        00-FE
Remote location name . . . . . > BSCCEL36 11 Name
Online at IPL . . . . . > *NO                *YES, *NO
Attached controller . . . . . > CTLBSCCEL   Name
Connection type . . . . . > *PP          *PP, *MPTRIB
Application type . . . . . > *BSCCEL 2   *BSCCEL, *RJE, *EML, *BSC38...
Contention resolution winner . . . > *SEC        *SEC, *PRI
Blocking type . . . . . > *ITB 3         *NONE, *ITB, *IRS, *NOSEP...
Remote BSCCEL . . . . . > *YES 4        *YES, *NO
Record length . . . . . > 1024 5         1-8192
Block length . . . . . > 2048 6         1-8192
Transmit in transparent mode . . . > *NO 7        *NO, *YES
Compress and decompress data . . . > *NO 8        *NO, *YES
Truncate trailing blanks . . . . . > *NO 9        *NO, *YES
Group separator type . . . . . > *EOT        *EOT, *OFCSYS, *DEV3740
More...

F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys

```

```

Create Device Desc (BSC) (CRTDEVBSCE)

Type choices, press Enter.

Text 'description' . . . . . > 'Device description for BSCCEL to S/36'

```

**Considerations for specifying the CRTDEVBSCE command:**

- 1 The *Remote location name* (RMTLOCNAME parameter) specified here should match the remote location name specified on the ADDICFDEVE, OVRICFDEVE, or CHGICFDEVE command. The remote location name provides the link between the device description and the program device entry.
- 2 Specify \*BSCCEL for communications applications using ICF files.
- 3 The value specified for the *Blocking type* (BLOCK parameter) must be \*ITB if the *ITB mode* prompt specified on display 52.0 of the CNFIGICF procedure is Y.
- 4 The value specified for the *Remote BSCCEL* (RMTBSCCEL parameter) must be \*YES if the *Partner* prompt specified on display 52.0 of the CNFIGICF procedure is 1-NORM; if the System/36 specifies 2-ATTR, this value must be \*NO.

- 5 The value specified for the *Record length* (RCDLEN parameter) must match the value specified for the *Maximum user record length* prompt on display 22.0 of the CNFIGICF procedure.
- 6 The value specified for the *Block length* (BLKLEN parameter) must match the *Block length* specified on display 52.0 of the CNFIGICF procedure.
- 7 The value specified for *Transmit in transparent mode* (TRANSPY parameter) must match the value specified for the *Transparency* prompt on display 50.0 of the CNFIGICF procedure.
- 8 The values specified for *Data compression* (DTACPR parameter) and *Truncate trailing blanks* (TRUNC parameter) must match the

value specified for the *Blank* prompt on display 52.0 of the CNFIGICF procedure. Because *0-No* is specified on the System/36 prompt, \*NO is specified for both AS/400 parameters.

- 9 The value specified for *Group separator* (GRPSEP parameter) must be \*EOT because the values specified for the *3740 multiple files?* and *Office systems mode?* prompts on display 52.0 of the CNFIGICF procedure are both specified N.

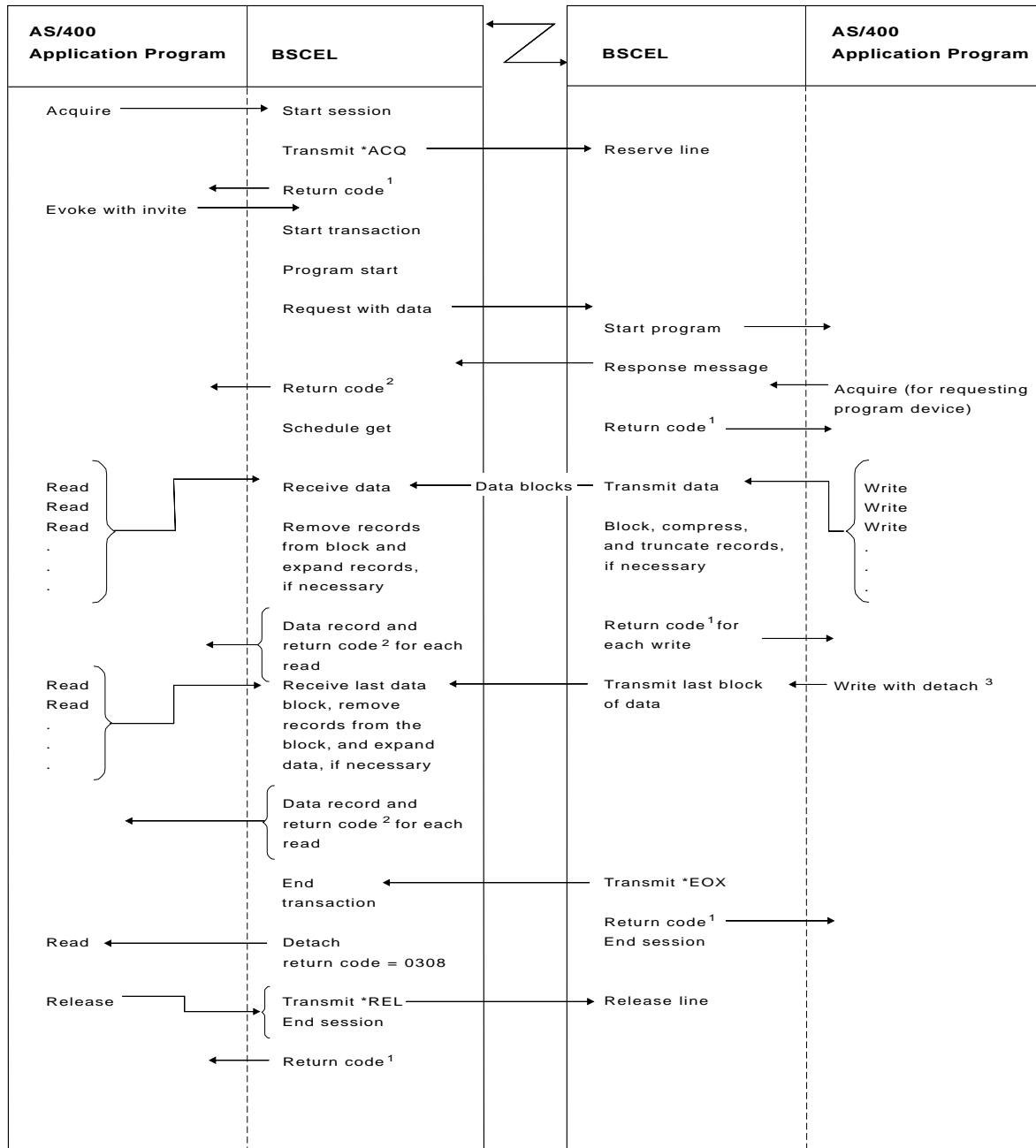
If *3740 multiple files?* is specified Y on the System/36, the AS/400 GRPSEP value must be \*DEV3740; if *Office systems mode?* is specified Y, the GRPSEP value must be \*OFCSYS.



# Appendix F. BSCEL Application Flow Examples

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

Figure F-1 is an example of communications between two AS/400 application programs. Each program is using BSCEL (RMTBSCEL(\*YES)) and data records are blocked.



<sup>1</sup> Normal return code is 0000, unless an error occurs.

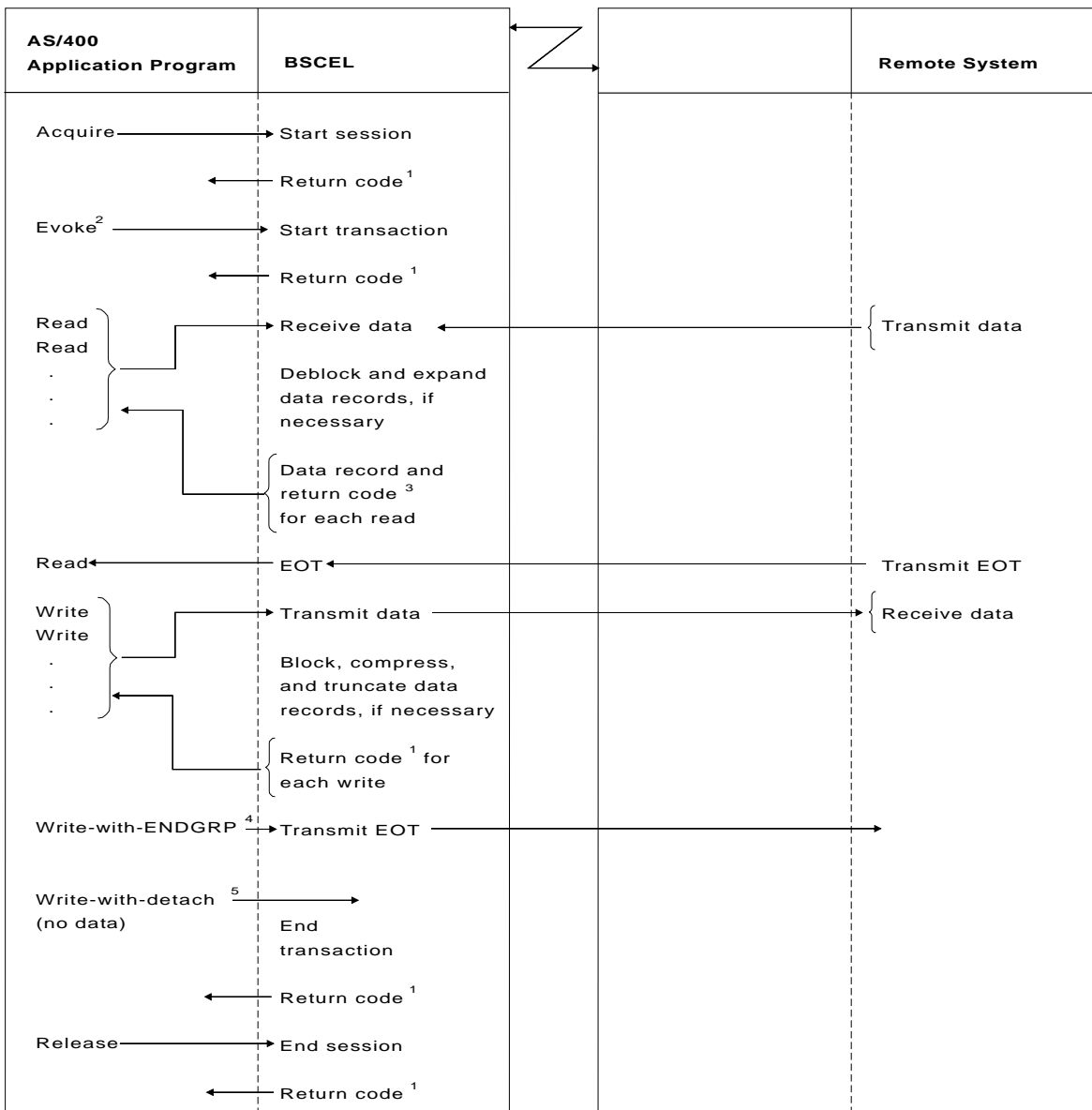
<sup>2</sup> Normal return code is 0001, unless an error occurs.

<sup>3</sup> The detach function ends both the transaction and the session.

RSLS355-5

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

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



<sup>1</sup>Normal return code is 0000, unless an error occurs.

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

<sup>3</sup>Normal return code is 0001, unless an error occurs.

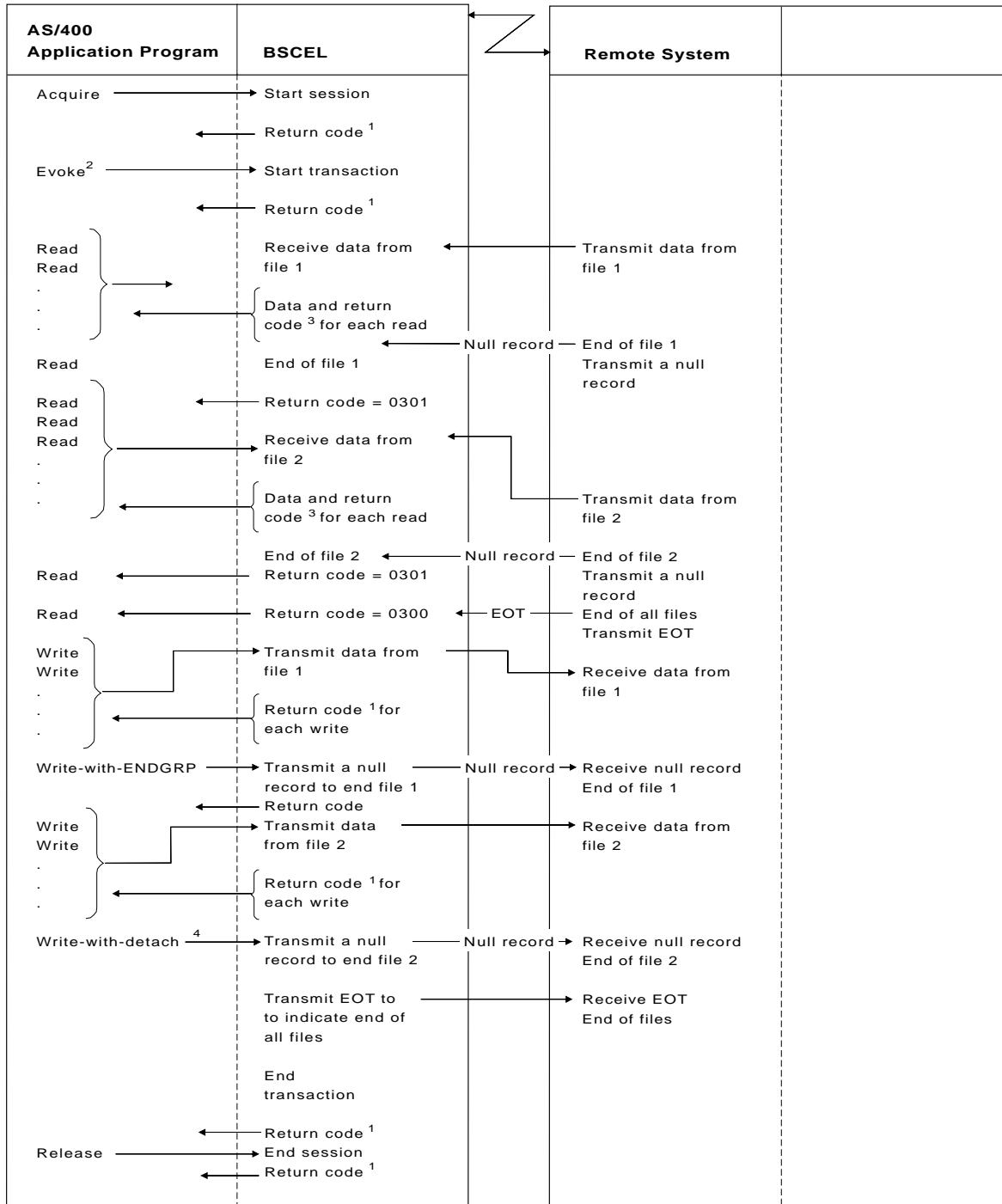
<sup>4</sup>The end-of-group function is optional in this example. However, either the allow-write function or the detach function must be used to force BSCSEL to transmit EOT if end-of-group is not used.

<sup>5</sup>The write operation with detach function is optional, since an evoke function is not required for RMTBSCSEL (\*NO).

RSL356-4

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

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



<sup>1</sup> Normal return code is 0000, unless an error occurs.

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

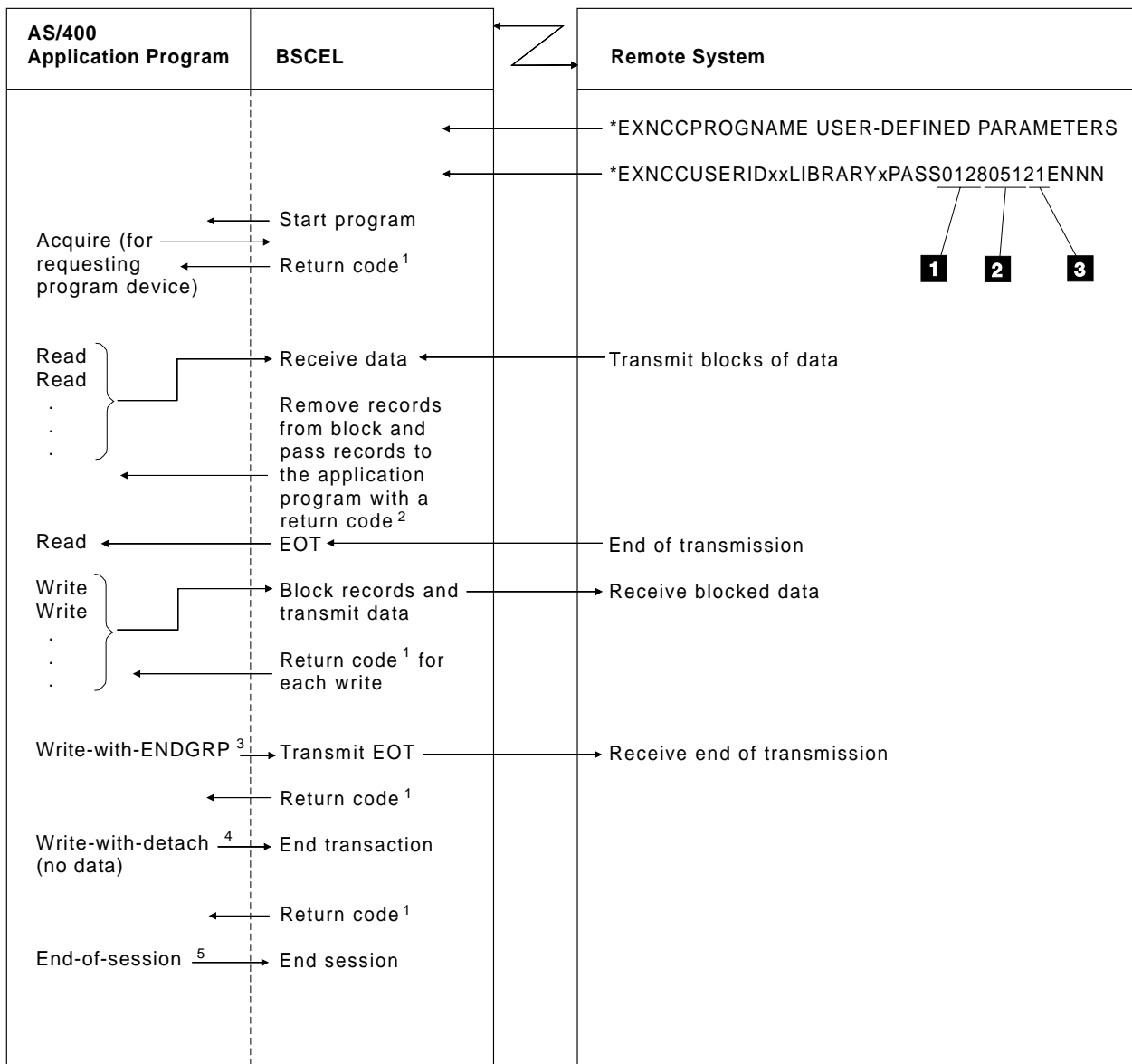
<sup>3</sup> Normal return code is 0001, unless an error occurs.

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

RSL3357-4

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

Figure F-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).



<sup>1</sup>Normal return code is 0000, unless an error occurs.

<sup>2</sup>Normal return code is 0001, unless an error occurs.

<sup>3</sup>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 BSCCL to transmit EOT if end-of-group is not used.

<sup>4</sup>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.

<sup>5</sup>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.

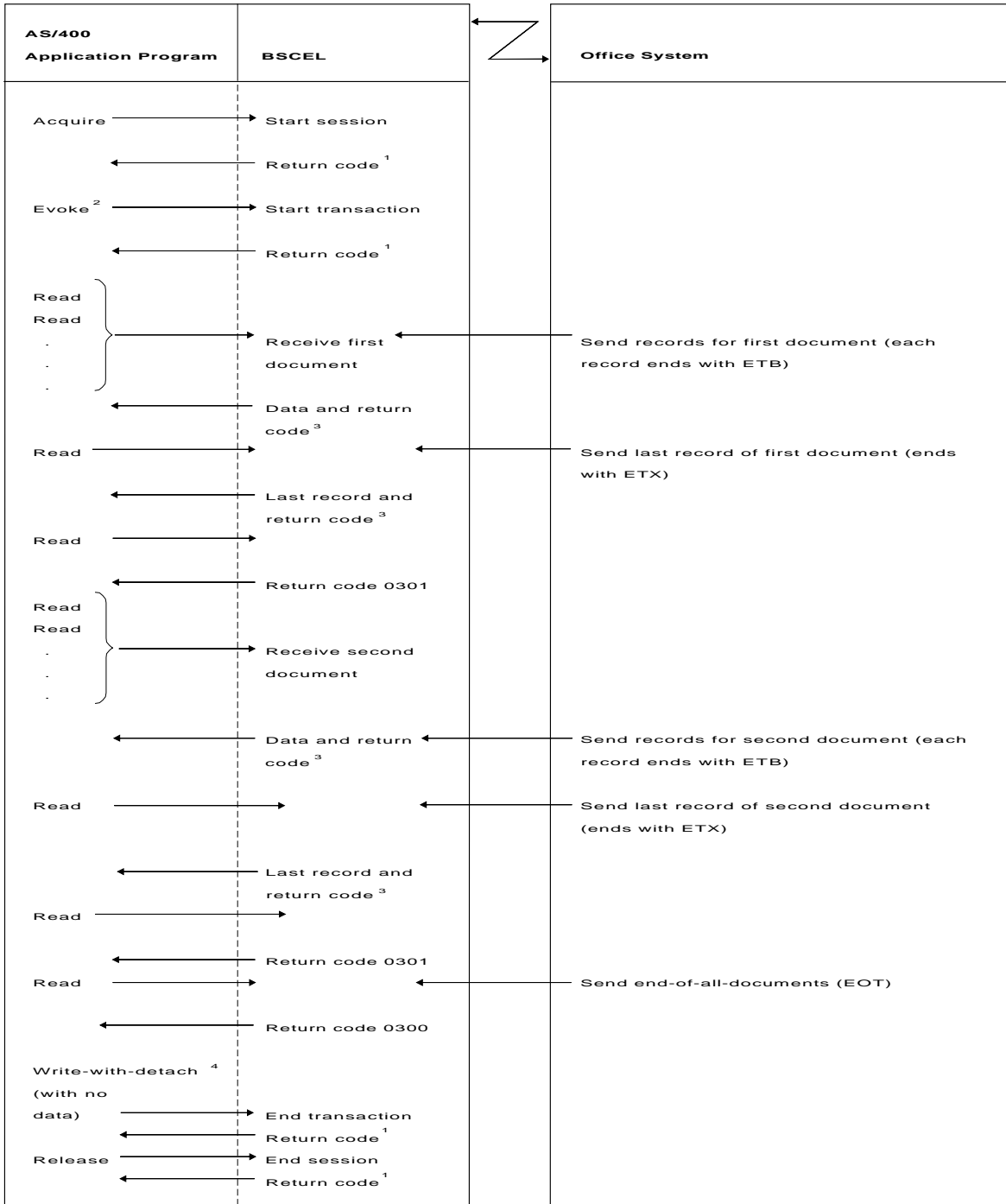
RSL358-6

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

The remote system in Figure F-4 on page F-4 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 F-5 on page F-6 is an example of an AS/400 system receiving data from an office system (RMTBSCEL(\*NO)). This example uses GRPSEP(\*OFCSYS).



<sup>1</sup> Normal return code is 0000, unless an error occurs.

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

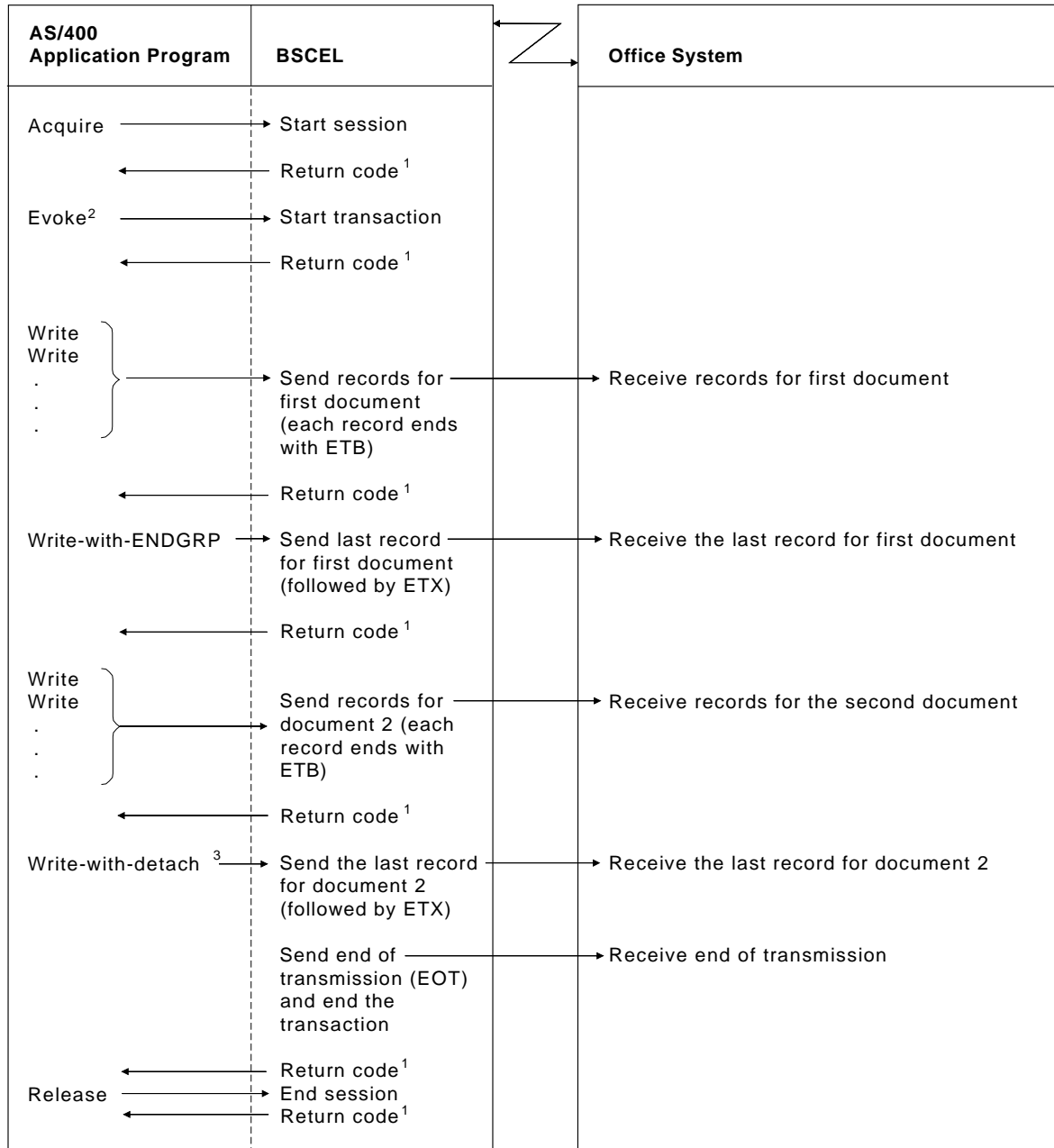
<sup>3</sup> Normal return code is 0001, unless an error occurs.

<sup>4</sup> The write operation with detach function is optional, since an evoke function is not required for RMTBSCCEL (\*NO).

RSL3359-7

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

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



<sup>1</sup> Normal return code is 0000, unless an error occurs.

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

<sup>3</sup> The 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.

RSL360-5

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

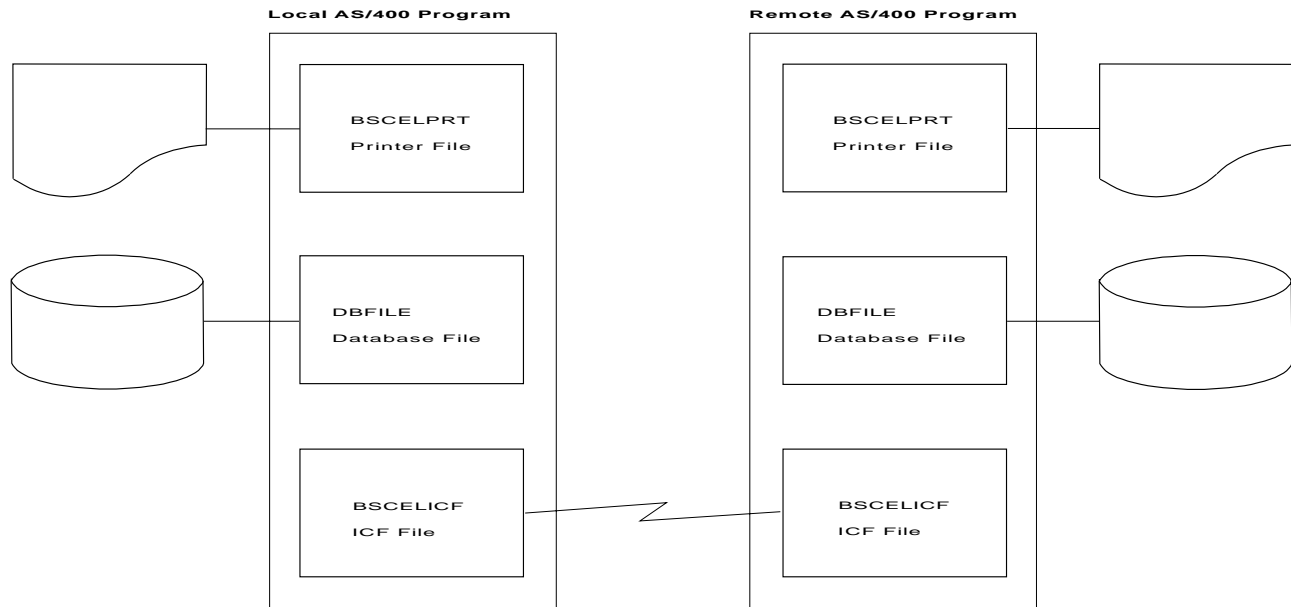




## Appendix G. Example Programs

This appendix provides sample programs in COBOL/400, RPG/400, and ILE C/400 to demonstrate how BSCCEL is used. The example programs included in this appendix are also available in the QUSRTOOL library (see file QATTINFO, member T9593INF in library QUSRTOOL).

Figure G-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.



RSL5362-1

Figure G-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 binary synchronous line, controller, and device descriptions used by the local system:

```
CRTLINBSC LIND(BSCELSRC) RSRRCNAME(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:

**BSCCELICF** 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/BSCCELICF)
  SRCFILE(BSCELIB/QDDSSRC)
  SRCMBR(BSCCELICF) TEXT('ICF FILE FOR BSCCEL')
```

The DDS used by this file is shown in Figure G-2.

```

                                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 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 G-2. DDS for the BSCCELICF File Used by the Local System (COBOL/400 Language)

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

**BSCSELPRT**

The printer file used to format output to a printer.

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

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

## Program Explanation

The following describes the COBOL/400 program on the local system, which is shown in Figure G-5 on page G-7.

- 1** The three files used in this program are specified in the input/output section (file-control) portion of the program.  

<b>DBFILE</b>	The name of the database file that contains the data which will be sent to the remote system.
<b>BSCSELICF</b>	The name of the ICF file used to send and receive data between the local and remote systems.
<b>BSCSELPRF</b>	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 BSCSELICF 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 (BSCSELICF) by the ADDICFDEVE command.
- 4** The next routine builds the program start request and issues the write operation to (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 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 database 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.

```

Program . . . . . : C85ELSRC
Library . . . . . : BSCELIB
Source file . . . . . : QCBLSRC
Library . . . . . : BSCELIB
Source member . . . . . : C85ELSRC 10/16/90 16:07:48
Generation severity level . . . . . : 29
Text 'description' . . . . . : Source Cobol program for BSCEL
Source listing options . . . . . : *NONE
Generation options . . . . . : *NONE
Message limit:
Number of messages . . . . . : *NOMAX
Message limit severity . . . . . : 29
Print file . . . . . : QSYSPRT
Library . . . . . : *LIBL
FIPS flagging . . . . . : *NOFIPS *NOSEG *NODEB *NOOBSOLETE
SAA flagging . . . . . : *NOFLAG
Flagging severity . . . . . : 0
Replace program . . . . . : *YES
Target release . . . . . : *CURRENT
User profile . . . . . : *USER
Authority . . . . . : *LIBCRTAUT
Compiler . . . . . : IBM AS/400 COBOL/400
STMT SEQNBR -A 1 B.+...2....+...3....+...4....+...5....+...6....+...7..IDENTFCN S COPYNAME CHG DATE
1 000010 IDENTIFICATION DIVISION.
2 000020 PROGRAM-ID. C85ELSRC.
3 000030 AUTHOR. JSP.
4 000040 DATE-WRITTEN. 11/87.
5 000050 DATE-COMPILED.
10/17/90 15:18:35 .
6 000060 ENVIRONMENT DIVISION.
7 000070 CONFIGURATION SECTION.
8 000080 SOURCE-COMPUTER. IBM-S3X.
9 000090 OBJECT-COMPUTER. IBM-S3X.
10 000100 SPECIAL-NAMES. I-O-FEEDBACK IS FEEDBACK-AREA.
11 000110 INPUT-OUTPUT SECTION.
12 000120 FILE-CONTROL.
13 000130 SELECT DBFILE ASSIGN TO DATABASE-DBFILE.
14 000140 SELECT BSCELICF ASSIGN TO WORKSTATION-BSCELICF-SI
15 000150 ORGANIZATION IS TRANSACTION
16 000160 CONTROL-AREA IS TRAN-CTL-AREA
17 000170 FILE STATUS IS STATUS-IND MAJ-MIN.
18 000180 SELECT BSCELPRT ASSIGN TO FORMATFILE-BSCELPRT
19 000190 ORGANIZATION IS SEQUENTIAL.
20 000200 DATA DIVISION.
21 000210 FILE SECTION.
22 000220 FD DBFILE
23 000230 LABEL RECORDS ARE STANDARD.
24 000240 01 DBREC. COPY DDS-ALL-FORMATS-I OF DBFILE.
25 +000001 05 DBFILE-RECORD PIC X(79). <-ALL-FMTS
+000002* I-O FORMAT:DBFMT FROM FILE DBFILE OF LIBRARY BSCELIB <-ALL-FMTS
+000003* <-ALL-FMTS
26 +000004 05 DBFMT REDEFINES DBFILE-RECORD. <-ALL-FMTS
27 +000005 06 CUSNUM PIC X(7). <-ALL-FMTS
+000006* Customer Number <-ALL-FMTS
28 +000007 06 CUSNAM PIC X(25). <-ALL-FMTS
+000008* Customer Name <-ALL-FMTS
29 +000009 06 ADDR PIC X(15). <-ALL-FMTS
+000010* Address <-ALL-FMTS
30 +000011 06 CITY PIC X(15). <-ALL-FMTS
+000012* City <-ALL-FMTS
31 +000013 06 STATE PIC X(2). <-ALL-FMTS
+000014* State <-ALL-FMTS
32 +000015 06 ZIP PIC S9(5). <-ALL-FMTS
+000016* Zip <-ALL-FMTS
33 +000017 06 CRDLMT PIC S9(5). <-ALL-FMTS
+000018* Credit Limit <-ALL-FMTS
34 +000019 06 CRDAMT PIC S9(5). <-ALL-FMTS
+000020* Credit Amount <-ALL-FMTS
35 000250 FD BSCELICF
36 000260 LABEL RECORDS ARE STANDARD.
37 000270 01 ICFREC. COPY DDS-ALL-FORMATS OF BSCELICF.
38 +000001 05 BSCELICF-RECORD PIC X(79). <-ALL-FMTS
+000002* INPUT FORMAT:RCVDATA FROM FILE BSCELICF OF LIBRARY BSCELIB <-ALL-FMTS
+000003* <-ALL-FMTS
39 +000004 05 RCVDATA-I REDEFINES BSCELICF-RECORD. <-ALL-FMTS
40 +000005 06 CUSNUM PIC X(7). <-ALL-FMTS
+000006* Customer Number <-ALL-FMTS
41 +000007 06 CUSNAM PIC X(25). <-ALL-FMTS

```

Figure G-5 (Part 1 of 6). COBOL/400 Source Program for the Local System

```

STMT SEQNBR -A 1 B.+. . . . 2. . . . +. . . . 3. . . . +. . . . 4. . . . +. . . . 5. . . . +. . . . 6. . . . +. . . . 7. . . . IDENTFCN S COPYNAME CHG DATE
+000008* Customer Name <-ALL-FMTS
42 +000009 06 ADDR PIC X(15). <-ALL-FMTS
+000010* Address <-ALL-FMTS
43 +000011 06 CITY PIC X(15). <-ALL-FMTS
+000012* City <-ALL-FMTS
44 +000013 06 STATE PIC X(2). <-ALL-FMTS
+000014* State <-ALL-FMTS
45 +000015 06 ZIP PIC S9(5). <-ALL-FMTS
+000016* Zip <-ALL-FMTS
46 +000017 06 CRDLMT PIC S9(5). <-ALL-FMTS
+000018* Credit Limit <-ALL-FMTS
47 +000019 06 CRDAMT PIC S9(5). <-ALL-FMTS
+000020* Credit Amount <-ALL-FMTS
+000021* OUTPUT FORMAT:RCVDATA FROM FILE BSCELICF OF LIBRARY BSCELIB <-ALL-FMTS
+000022* <-ALL-FMTS
48 +000023 05 RCVDATA-0 REDEFINES BSCELICF-RECORD. <-ALL-FMTS
49 +000024 06 CUSNUM PIC X(7). <-ALL-FMTS
+000025* Customer Number <-ALL-FMTS
50 +000026 06 CUSNAM PIC X(25). <-ALL-FMTS
+000027* Customer Name <-ALL-FMTS
51 +000028 06 ADDR PIC X(15). <-ALL-FMTS
+000029* Address <-ALL-FMTS
52 +000030 06 CITY PIC X(15). <-ALL-FMTS
+000031* City <-ALL-FMTS
53 +000032 06 STATE PIC X(2). <-ALL-FMTS
+000033* State <-ALL-FMTS
54 +000034 06 ZIP PIC S9(5). <-ALL-FMTS
+000035* Zip <-ALL-FMTS
55 +000036 06 CRDLMT PIC S9(5). <-ALL-FMTS
+000037* Credit Limit <-ALL-FMTS
56 +000038 06 CRDAMT PIC S9(5). <-ALL-FMTS
+000039* Credit Amount <-ALL-FMTS
+000040* INPUT FORMAT:SNDDATA FROM FILE BSCELICF OF LIBRARY BSCELIB <-ALL-FMTS
+000041* <-ALL-FMTS
57 +000042 05 SNDDATA-I REDEFINES BSCELICF-RECORD. <-ALL-FMTS
58 +000043 06 CUSNUM PIC X(7). <-ALL-FMTS
+000044* Customer Number <-ALL-FMTS
59 +000045 06 CUSNAM PIC X(25). <-ALL-FMTS
+000046* Customer Name <-ALL-FMTS
60 +000047 06 ADDR PIC X(15). <-ALL-FMTS
+000048* Address <-ALL-FMTS
61 +000049 06 CITY PIC X(15). <-ALL-FMTS
+000050* City <-ALL-FMTS
62 +000051 06 STATE PIC X(2). <-ALL-FMTS
+000052* State <-ALL-FMTS
63 +000053 06 ZIP PIC S9(5). <-ALL-FMTS
+000054* Zip <-ALL-FMTS
64 +000055 06 CRDLMT PIC S9(5). <-ALL-FMTS
+000056* Credit Limit <-ALL-FMTS
65 +000057 06 CRDAMT PIC S9(5). <-ALL-FMTS
+000058* Credit Amount <-ALL-FMTS
+000059* OUTPUT FORMAT:SNDDATA FROM FILE BSCELICF OF LIBRARY BSCELIB <-ALL-FMTS
+000060* <-ALL-FMTS
66 +000061 05 SNDDATA-0 REDEFINES BSCELICF-RECORD. <-ALL-FMTS
67 +000062 06 CUSNUM PIC X(7). <-ALL-FMTS
STMT SEQNBR -A 1 B.+. . . . 2. . . . +. . . . 3. . . . +. . . . 4. . . . +. . . . 5. . . . +. . . . 6. . . . +. . . . 7. . . . IDENTFCN S COPYNAME CHG DATE
+000063* Customer Number <-ALL-FMTS
68 +000064 06 CUSNAM PIC X(25). <-ALL-FMTS
+000065* Customer Name <-ALL-FMTS
69 +000066 06 ADDR PIC X(15). <-ALL-FMTS
+000067* Address <-ALL-FMTS
70 +000068 06 CITY PIC X(15). <-ALL-FMTS
+000069* City <-ALL-FMTS
71 +000070 06 STATE PIC X(2). <-ALL-FMTS
+000071* State <-ALL-FMTS
72 +000072 06 ZIP PIC S9(5). <-ALL-FMTS
+000073* Zip <-ALL-FMTS
73 +000074 06 CRDLMT PIC S9(5). <-ALL-FMTS
+000075* Credit Limit <-ALL-FMTS
74 +000076 06 CRDAMT PIC S9(5). <-ALL-FMTS
+000077* Credit Amount <-ALL-FMTS
+000078* INPUT FORMAT:EVOKPGM FROM FILE BSCELICF OF LIBRARY BSCELIB <-ALL-FMTS
+000079* <-ALL-FMTS
+000080* 05 EVOKPGM-I REDEFINES BSCELICF-RECORD. <-ALL-FMTS
+000081* OUTPUT FORMAT:EVOKPGM FROM FILE BSCELICF OF LIBRARY BSCELIB <-ALL-FMTS
+000082* <-ALL-FMTS

```

Figure G-5 (Part 2 of 6). COBOL/400 Source Program for the Local System



```

75 +000083      05 EVOKPGM-0   REDEFINES BSCELICF-RECORD.          <-ALL-FMTS
76 +000084      06 PASS          PIC X(4).          <-ALL-FMTS
77 +000085      06 USERID       PIC X(8).          <-ALL-FMTS
78 +000086      06 LIB          PIC X(8).          <-ALL-FMTS
79 +000087      06 PGMID        PIC X(8).          <-ALL-FMTS
+000088* INPUT FORMAT:DETACH   FROM FILE BSCELICF  OF LIBRARY BSCELIB  <-ALL-FMTS
+000089*                                     <-ALL-FMTS
+000090*      05 DETACH-I       REDEFINES BSCELICF-RECORD.          <-ALL-FMTS
+000091* OUTPUT FORMAT:DETACH  FROM FILE BSCELICF  OF LIBRARY BSCELIB  <-ALL-FMTS
+000092*                                     <-ALL-FMTS
+000093*      05 DETACH-0       REDEFINES BSCELICF-RECORD.          <-ALL-FMTS
80 000280 FD BSCELPRT
81 000290 LABEL RECORDS ARE STANDARD
82 000300 DATA RECORD IS PRINT-RECORD.
83 000310 01 PRINT-RECORD.
84 000320 COPY DDS-ALL-FORMATS-0 OF BSCELPRT.
85 +000001      05 BSCELPRT-RECORD PIC X(79).          <-ALL-FMTS
+000002* I-O FORMAT:HDG      FROM FILE BSCELPRT  OF LIBRARY BSCELIB  <-ALL-FMTS
+000003*                                     <-ALL-FMTS
+000004*      05 HDG          REDEFINES BSCELPRT-RECORD.          <-ALL-FMTS
+000005* OUTPUT FORMAT:DTL    FROM FILE BSCELPRT  OF LIBRARY BSCELIB  <-ALL-FMTS
+000006*                                     <-ALL-FMTS
86 +000007      05 DTL-0        REDEFINES BSCELPRT-RECORD.          <-ALL-FMTS
87 +000008      06 CUSNUM       PIC X(7).          <-ALL-FMTS
+000009* Customer Number          <-ALL-FMTS
88 +000010      06 CUSNAM       PIC X(25).         <-ALL-FMTS
+000011* Customer Name          <-ALL-FMTS
89 +000012      06 ADDR         PIC X(15).         <-ALL-FMTS
+000013* Address              <-ALL-FMTS
90 +000014      06 CITY         PIC X(15).         <-ALL-FMTS
+000015* City                  <-ALL-FMTS
91 +000016      06 STATE        PIC X(2).          <-ALL-FMTS
+000017* State                  <-ALL-FMTS
92 +000018      06 ZIP          PIC S9(5).         <-ALL-FMTS
+000019* Zip                    <-ALL-FMTS
STMT SEQNBR -A 1 B.+. . . . 2. . . . +. . . . 3. . . . +. . . . 4. . . . +. . . . 5. . . . +. . . . 6. . . . +. . . . 7. . . . IDENTFCN S COPYNAME CHG DATE
93 +000020      06 CRDLMT       PIC S9(5).         <-ALL-FMTS
+000021* Credit Limit          <-ALL-FMTS
94 +000022      06 CRDAMT       PIC S9(5).         <-ALL-FMTS
+000023* Credit Amount        <-ALL-FMTS
+000024* OUTPUT FORMAT:ERROR  FROM FILE BSCELPRT  OF LIBRARY BSCELIB  <-ALL-FMTS
+000025*                                     <-ALL-FMTS
95 +000026      05 ERROR-0      REDEFINES BSCELPRT-RECORD.          <-ALL-FMTS
96 +000027      06 PGMDEV       PIC X(10).         <-ALL-FMTS
97 +000028      06 FMTNM        PIC X(8).          <-ALL-FMTS
98 +000029      06 MAJCOD       PIC X(2).          <-ALL-FMTS
99 +000030      06 MINCOD       PIC X(2).          <-ALL-FMTS
100 000330 WORKING-STORAGE SECTION.
101 000340 77 STATUS-IND       PIC XX.
102 000350 77 INDON           PIC 1 VALUE B"1".
103 000360 77 INDOFF          PIC 1 VALUE B"0".
000370
104 000380 01 PGM-INDIC-AREA.
105 000390 05 PGM-INDIC       PIC 1 OCCURS 99 TIMES
106 000400 INDICATOR 1.
107 000410 01 TRAN-CTL-AREA.
108 000420 05 FILLER          PIC X(2).
109 000430 05 PGM-DEV-NAME    PIC X(10).
110 000440 05 REC-FOR-NAME    PIC X(10).
2 000450
111 000460 01 FEEDBACK-DATA.
112 000470 05 FILLER          PIC X(20).
113 000480 05 FMTNM          PIC X(10).
114 000490 05 FILLER          PIC X(2).
115 000500 05 PGMDEV         PIC X(10).
116 000510 05 FILLER         PIC X(102).
117 000520 05 FILLER         PIC X(34).
118 000530 05 MAJCOD         PIC X(2).
119 000540 05 MINCOD         PIC X(2).
000550
120 000560 01 MAJ-MIN.
121 000570 05 MAJ            PIC X(2).
122 000580 05 MIN            PIC X(2).
000590
123 000600 PROCEDURE DIVISION.
000610

```

Figure G-5 (Part 3 of 6). COBOL/400 Source Program for the Local System

```

000620*****
000630* THE FOLLOWING DECLARATIVES SECTION IS AN ERROR ROUTINE THAT *
000640* IS RUN WHEN AN ERROR OCCURS ON THE READ OR THE WRITE OF THE *
000650* OS/400-ICF FILE "BSCCELICF". THE ROUTINE MOVES DATA FROM THE *
000660* I-O FEEDBACK AREA TO THE "ERROR" FORMAT OF THE PRINT FILE *
000670* "BSCCLPRT". ALL OF THE FILES ARE THEN CLOSED AND THE PROGRAM *
000680* IS TERMINATED. *
000690*****
000700 DECLARATIVES.
000710 ERROR-SECTION SECTION.
000720     USE AFTER STANDARD ERROR PROCEDURE ON OUTPUT.
10 000730 ERROR-PARAGRAPH.
124 000740     ACCEPT FEEDBACK-DATA FROM FEEDBACK-AREA.
125 000750     MOVE CORRESPONDING FEEDBACK-DATA TO ERROR-0.
    *     ** CORRESPONDING items for statement 125:
STMT SEQNBR -A 1 B...2...3...4...5...6...7..IDENTFCN S COPYNAME CHG DATE
    *     ** FMTNM
    *     ** PGMDEV
    *     ** MAJCOD
    *     ** MINCOD
    *     ** End of CORRESPONDING items for statement 125
126 000760     WRITE PRINT-RECORD FORMAT IS "ERROR".
127 000770     CLOSE BSCCELICF
    000780         DBFILE
    000790         BSCCLPRT.
128 000800     STOP RUN.
    000810     END DECLARATIVES.
    000820
    000830     MAIN-PROGRAM SECTION.
    000840
    000850     000-PROCESS-GROUP.
    000860
3 000870     OPEN I-O BSCCELICF.
130 000880     OPEN INPUT DBFILE.
131 000890     OPEN OUTPUT BSCCLPRT.
    000900
132 000910     ACQUIRE "SOURCE " FOR BSCCELICF.
133 000920     MOVE "SOURCE " TO PGM-DEV-NAME.
    000930
134 000940     MOVE ZEROES TO PGM-INDIC-AREA.
    000950
000960*****
000970*
4 000980* THE FOLLOWING ROUTINE EVOKES THE PROGRAM 'C85ELTGT' ON THE
000990* TARGET SYSTEM.
001000*
001010*****
001020
135 001030     MOVE "BSCCELIB " TO LIB OF EVOKPGM-0.
136 001040     MOVE "C85ELTGT" TO PGMID OF EVOKPGM-0.
137 001050     MOVE "BSCE" TO PASS OF EVOKPGM-0.
138 001060     MOVE "BSCCEL " TO USERID OF EVOKPGM-0.
139 001070     WRITE ICFREC     FORMAT IS "EVOKPGM"
    001080         TERMINAL IS PGM-DEV-NAME.
    001090
001100*****
5 001110* SEND DATA BASE FILE TO THE TARGET SYSTEM.
001120*****
001130
140 001140     PERFORM 100-SEND-DATA UNTIL PGM-INDIC(99) = INDON.
    001150
141 001160     PERFORM 110-PAGE-HEADING.
    001170
001180*****
6 001190* READ DATA FROM TARGET SYSTEM AND PRINT IT.
001200*****
001210
142 001220     PERFORM 120-READ-DATA UNTIL PGM-INDIC(38) = INDON.
    001230
001240*****
7 001250* DETACH FROM TARGET SYSTEM AND CLOSE FILES.

```

Figure G-5 (Part 4 of 6). COBOL/400 Source Program for the Local System

```

STMT SEQNBR -A 1 B.+. . . . 2. . . . +. . . . 3. . . . +. . . . 4. . . . +. . . . 5. . . . +. . . . 6. . . . +. . . . 7. . . . IDENTFCN S COPYNAME CHG DATE
001260*****
001270
143 001280 WRITE ICFREC FORMAT IS "DETACH"
001290 TERMINAL IS PGM-DEV-NAME.
001300
001310 050-CLOSE-END.
001320
144 001330 CLOSE BSCELICF
001340 DBFILE
001350 BSCELPRT.
001360
145 001370 STOP RUN.
001380
001390
001400* SUBROUTINES.
001410*****
001420* *
001430* THE FOLLOWING ROUTINE READS RECORDS FROM THE DATA BASE FILE *
001440* 'DBFILE'. WHEN 'DBFILE' REACHES 'END OF FILE' INDICATOR 99 *
001450* IS TURNED ON. *
001460* *
8 001470*****
001480 100-SEND-DATA.
146 001490 READ DBFILE
147 001500 AT END MOVE INDON TO PGM-INDIC(99).
148 001510 IF PGM-INDIC(99) EQUAL INDOFF THEN
149 001520 MOVE CORRESPONDING DBFMT TO SNDDATA-0
* ** CORRESPONDING items for statement 149:
* ** CUSNUM
* ** CUSNAM
* ** ADDR
* ** CITY
* ** STATE
* ** ZIP
* ** CRDLMT
* ** CRDAMT
* ** End of CORRESPONDING items for statement 149
150 001530 WRITE ICFREC FORMAT IS "SNDDATA"
001540 TERMINAL IS PGM-DEV-NAME.
001550
001560 110-PAGE-HEADING.
151 001570 WRITE PRINT-RECORD FORMAT IS "HDG".
001580
001590*****
001600* *
001610* THE FOLLOWING ROUTINE RECEIVES RECORDS FROM THE TARGET SYSTEM *
001620* UNTIL INDICATOR 40 IS TURNED ON. INDICATOR 40 SIGNALS THAT A *
001630* 'RCVENDGRP' HAS BEEN RECEIVED, AND THE TARGET PROGRAM HAS *
001640* STOPPED SENDING RECORDS. AS RECORDS ARE RECEIVED THEY ARE *
001650* PRINTED IN THE PRINT FILE 'BSCELPRT'. *
001660* *
001670*****
9 001680
001690 120-READ-DATA.
152 001700 READ BSCELICF FORMAT IS "RCVDATA"
STMT SEQNBR -A 1 B.+. . . . 2. . . . +. . . . 3. . . . +. . . . 4. . . . +. . . . 5. . . . +. . . . 6. . . . +. . . . 7. . . . IDENTFCN S COPYNAME CHG DATE
001710 TERMINAL IS PGM-DEV-NAME
001720 INDICATORS ARE PGM-INDIC-AREA.
153 001730 IF PGM-INDIC(40) EQUAL INDOFF THEN
154 001740 MOVE CORRESPONDING RCVDATA-I TO DTL-0
* ** CORRESPONDING items for statement 154:
* ** CUSNUM
* ** CUSNAM
* ** ADDR
* ** CITY
* ** STATE
* ** ZIP
* ** CRDLMT
* ** CRDAMT
* ** End of CORRESPONDING items for statement 154
155 001750 WRITE PRINT-RECORD FORMAT IS "DTL"
156 001760 AT EOP PERFORM 110-PAGE-HEADING.
* * * * * E N D O F S O U R C E * * * * *

```

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

```

STMT
* 22 MSGID: LBL0650 SEVERITY: 00 SEQNBR: 000220
    Message . . . . : Blocking/Deblocking for file 'DBFILE' will
    be performed by compiler-generated code.
* 37 MSGID: LBL0600 SEVERITY: 10 SEQNBR: 000270
    Message . . . . : No INPUT fields found for format EVOKPGM.
* 37 MSGID: LBL0600 SEVERITY: 10 SEQNBR: 000270
    Message . . . . : No INPUT fields found for format DETACH.
* 37 MSGID: LBL0600 SEVERITY: 10 SEQNBR: 000270
    Message . . . . : No OUTPUT fields found for format DETACH.
* 84 MSGID: LBL0600 SEVERITY: 10 SEQNBR: 000320
    Message . . . . : No INPUT fields found for format HDG.
    * * * * * E N D   O F   M E S S A G E S   * * * * *
                                Message Summary
Total      Info(0-4)   Warning(5-19)   Error(20-29)   Severe(30-39)   Terminal(40-99)
   5         1         4             0             0             0
Source records read . . . . . : 176
Copy records read . . . . . : 143
Copy members processed . . . . . : 3
Sequence errors . . . . . : 0
Highest severity message issued . . : 10
LBL0901 00 Program C85ELSRC created in library BSCSELIB.
    * * * * * E N D   O F   C O M P I L A T I O N   * * * * *

```

Figure G-5 (Part 6 of 6). 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 BSCSEL target program.

### Configuration

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

```

CRTLINBSC LIND(BSCSELTGT) RSRNAME(LIN021)
    ONLINE(*NO)

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

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

```

## Program Files

The following files are used by the remote system:

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

```

CRITICFF FILE(BSCSELIB/BSCSELICF)
    SRCFILE(BSCSELIB/QDDSSRC)
    SRCMBR(BSCSELICF) TEXT('ICF FILE FOR BSCSEL')

```

The DDS used by this file is shown in Figure G-6 on page G-13.

```

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

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 G-7 on page G-14.

```

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 BSCEL 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 G-7. DDS for the Database File Used in BSCEL Data Transfer Program on the Remote System (COBOL/400 Language)

**BSCELPRT** The printer file used to format output to a printer. The DDS for this file is shown in Figure G-8.

```

Data Description Source
SEQNBR *...+...1...+...2...+...3...+...4...+...5...+...6...+...7.
100  A          REF(DBFILE)
200  A          R HDG          SKIPB(5)
300  A          SDATE 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 G-8. DDS for the Printer File on the Remote System (COBOL/400 Language)

## Program Explanation

The following describes the COBOL/400 program on the remote system, which is shown in Figure G-9 on page G-16.

- 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.

**BSCSELICF** The name of the ICF file used to send and receive data between the local and remote systems.

**BSCLEPRT** 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 BSCSELICF 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 (BSCSELICF) 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.

```

Program . . . . . : C85ELTGT
Library . . . . . : BSCELIB
Source file . . . . . : QCBLSRC
Library . . . . . : BSCELIB
Source member . . . . . : C85ELTGT 10/16/90 16:07:48
Generation severity level . . . . . : 29
Text 'description' . . . . . : Target Cobol program for BSCEL
Source listing options . . . . . : *NONE
Generation options . . . . . : *NONE
Message limit:
  Number of messages . . . . . : *NOMAX
  Message limit severity . . . . . : 29
Print file . . . . . : QSYSVRT
Library . . . . . : *LIBL
FIPS flagging . . . . . : *NOFIPS *NOSEG *NODEB *NOOBSOLETE
SAA flagging . . . . . : *NOFLAG
Flagging severity . . . . . : 0
Replace program . . . . . : *YES
Target release . . . . . : *CURRENT
User profile . . . . . : *USER
Authority . . . . . : *LIBCRTAUT
Compiler . . . . . : IBM AS/400 COBOL/400
STMT SEQNBR -A 1 B.+.2....3....4....5....6....7..IDENTFCN S COPYNAME CHG DATE
1 000010 IDENTIFICATION DIVISION.
2 000020 PROGRAM-ID. C85ELTGT.
3 000030 AUTHOR. JSP.
4 000040 DATE-WRITTEN. 11/87.
5 000050 DATE-COMPILED. 10/17/90 15:19:11 .
6 000060 ENVIRONMENT DIVISION.
7 000070 CONFIGURATION SECTION.
8 000080 SOURCE-COMPUTER. IBM-S3X.
9 000090 OBJECT-COMPUTER. IBM-S3X.
10 000100 SPECIAL-NAMES. I-O-FEEDBACK IS FEEDBACK-AREA.
11 000110 INPUT-OUTPUT SECTION.
12 000120 FILE-CONTROL.
13 000130 SELECT DBFILE ASSIGN TO DATABASE-DBFILE.
14 000140 SELECT BSCELICF ASSIGN TO WORKSTATION-BSCELICF-SI
15 000150 ORGANIZATION IS TRANSACTION
16 000160 CONTROL-AREA IS TRAN-CTL-AREA
17 000170 FILE STATUS IS STATUS-IND MAJ-MIN.
18 000180 SELECT BSCELPRT ASSIGN TO FORMATFILE-BSCELPRT
19 000190 ORGANIZATION IS SEQUENTIAL.
20 000200 DATA DIVISION.
21 000210 FILE SECTION.
22 000220 FD DBFILE
23 000230 LABEL RECORDS ARE STANDARD.
24 000240 01 DBREC. COPY DDS-ALL-FORMATS-I OF DBFILE.
25 +000001 05 DBFILE-RECORD PIC X(79). <-ALL-FMTS
+000002* I-O FORMAT:DBFMT FROM FILE DBFILE OF LIBRARY BSCELIB <-ALL-FMTS
+000003* <-ALL-FMTS
26 +000004 05 DBFMT REDEFINES DBFILE-RECORD. <-ALL-FMTS
27 +000005 06 CUSNUM PIC X(7). <-ALL-FMTS
+000006* Customer Number <-ALL-FMTS
28 +000007 06 CUSNAM PIC X(25). <-ALL-FMTS
+000008* Customer Name <-ALL-FMTS
29 +000009 06 ADDR PIC X(15). <-ALL-FMTS
+000010* Address <-ALL-FMTS
30 +000011 06 CITY PIC X(15). <-ALL-FMTS
+000012* City <-ALL-FMTS
31 +000013 06 STATE PIC X(2). <-ALL-FMTS
+000014* State <-ALL-FMTS
32 +000015 06 ZIP PIC S9(5). <-ALL-FMTS
+000016* Zip <-ALL-FMTS
33 +000017 06 CRDLMT PIC S9(5). <-ALL-FMTS
+000018* Credit Limit <-ALL-FMTS
34 +000019 06 CRDAMT PIC S9(5). <-ALL-FMTS
+000020* Credit Amount <-ALL-FMTS
35 000250 FD BSCELICF
36 000260 LABEL RECORDS ARE STANDARD.
37 000270 01 ICFREC. COPY DDS-ALL-FORMATS OF BSCELICF.
38 +000001 05 BSCELICF-RECORD PIC X(79). <-ALL-FMTS
+000002* INPUT FORMAT:RCVDATA FROM FILE BSCELICF OF LIBRARY BSCELIB <-ALL-FMTS
+000003* <-ALL-FMTS
39 +000004 05 RCVDATA-I REDEFINES BSCELICF-RECORD. <-ALL-FMTS
40 +000005 06 CUSNUM PIC X(7). <-ALL-FMTS
+000006* Customer Number <-ALL-FMTS
41 +000007 06 CUSNAM PIC X(25). <-ALL-FMTS
+000008* Customer Name <-ALL-FMTS

```

Figure G-9 (Part 1 of 6). COBOL/400 Source Program for the Remote System



```

STMT SEQNBR -A 1 B.+. . . . 2. . . . +. . . . 3. . . . +. . . . 4. . . . +. . . . 5. . . . +. . . . 6. . . . +. . . . 7. . . . IDENTFCN S COPYNAME CHG DATE
42 +000009      06 ADDR      PIC X(15).      <-ALL-FMTS
+000010*      Address      <-ALL-FMTS
43 +000011      06 CITY      PIC X(15).      <-ALL-FMTS
+000012*      City      <-ALL-FMTS
44 +000013      06 STATE     PIC X(2).      <-ALL-FMTS
+000014*      State      <-ALL-FMTS
45 +000015      06 ZIP       PIC S9(5).     <-ALL-FMTS
+000016*      Zip      <-ALL-FMTS
46 +000017      06 CRDLMT    PIC S9(5).     <-ALL-FMTS
+000018*      Credit Limit <-ALL-FMTS
47 +000019      06 CRDAMT    PIC S9(5).     <-ALL-FMTS
+000020*      Credit Amount <-ALL-FMTS
+000021* OUTPUT FORMAT:RCVDATA FROM FILE BSCELCIF OF LIBRARY BSCELIB <-ALL-FMTS
+000022* <-ALL-FMTS
48 +000023      05 RCVDATA-0 REDEFINES BSCELCIF-RECORD. <-ALL-FMTS
49 +000024      06 CUSNUM    PIC X(7).      <-ALL-FMTS
+000025*      Customer Number <-ALL-FMTS
50 +000026      06 CUSNAM    PIC X(25).     <-ALL-FMTS
+000027*      Customer Name <-ALL-FMTS
51 +000028      06 ADDR      PIC X(15).     <-ALL-FMTS
+000029*      Address      <-ALL-FMTS
52 +000030      06 CITY      PIC X(15).     <-ALL-FMTS
+000031*      City      <-ALL-FMTS
53 +000032      06 STATE     PIC X(2).      <-ALL-FMTS
+000033*      State      <-ALL-FMTS
54 +000034      06 ZIP       PIC S9(5).     <-ALL-FMTS
+000035*      Zip      <-ALL-FMTS
55 +000036      06 CRDLMT    PIC S9(5).     <-ALL-FMTS
+000037*      Credit Limit <-ALL-FMTS
56 +000038      06 CRDAMT    PIC S9(5).     <-ALL-FMTS
+000039*      Credit Amount <-ALL-FMTS
+000040* INPUT FORMAT:SNDDATA FROM FILE BSCELCIF OF LIBRARY BSCELIB <-ALL-FMTS
+000041* <-ALL-FMTS
57 +000042      05 SNDDATA-I REDEFINES BSCELCIF-RECORD. <-ALL-FMTS
58 +000043      06 CUSNUM    PIC X(7).      <-ALL-FMTS
+000044*      Customer Number <-ALL-FMTS
59 +000045      06 CUSNAM    PIC X(25).     <-ALL-FMTS
+000046*      Customer Name <-ALL-FMTS
60 +000047      06 ADDR      PIC X(15).     <-ALL-FMTS
+000048*      Address      <-ALL-FMTS
61 +000049      06 CITY      PIC X(15).     <-ALL-FMTS
+000050*      City      <-ALL-FMTS
62 +000051      06 STATE     PIC X(2).      <-ALL-FMTS
+000052*      State      <-ALL-FMTS
63 +000053      06 ZIP       PIC S9(5).     <-ALL-FMTS
+000054*      Zip      <-ALL-FMTS
64 +000055      06 CRDLMT    PIC S9(5).     <-ALL-FMTS
+000056*      Credit Limit <-ALL-FMTS
65 +000057      06 CRDAMT    PIC S9(5).     <-ALL-FMTS
+000058*      Credit Amount <-ALL-FMTS
+000059* OUTPUT FORMAT:SNDDATA FROM FILE BSCELCIF OF LIBRARY BSCELIB <-ALL-FMTS
+000060* <-ALL-FMTS
66 +000061      05 SNDDATA-0 REDEFINES BSCELCIF-RECORD. <-ALL-FMTS
67 +000062      06 CUSNUM    PIC X(7).      <-ALL-FMTS
+000063*      Customer Number <-ALL-FMTS
STMT SEQNBR -A 1 B.+. . . . 2. . . . +. . . . 3. . . . +. . . . 4. . . . +. . . . 5. . . . +. . . . 6. . . . +. . . . 7. . . . IDENTFCN S COPYNAME CHG DATE
68 +000064      06 CUSNAM    PIC X(25).     <-ALL-FMTS
+000065*      Customer Name <-ALL-FMTS
69 +000066      06 ADDR      PIC X(15).     <-ALL-FMTS
+000067*      Address      <-ALL-FMTS
70 +000068      06 CITY      PIC X(15).     <-ALL-FMTS
+000069*      City      <-ALL-FMTS
71 +000070      06 STATE     PIC X(2).      <-ALL-FMTS
+000071*      State      <-ALL-FMTS
72 +000072      06 ZIP       PIC S9(5).     <-ALL-FMTS
+000073*      Zip      <-ALL-FMTS
73 +000074      06 CRDLMT    PIC S9(5).     <-ALL-FMTS
+000075*      Credit Limit <-ALL-FMTS
74 +000076      06 CRDAMT    PIC S9(5).     <-ALL-FMTS
+000077*      Credit Amount <-ALL-FMTS
+000078* INPUT FORMAT:EVOKPGM FROM FILE BSCELCIF OF LIBRARY BSCELIB <-ALL-FMTS
+000079* <-ALL-FMTS
+000080*      05 EVOKPGM-I REDEFINES BSCELCIF-RECORD. <-ALL-FMTS

```

Figure G-9 (Part 2 of 6). COBOL/400 Source Program for the Remote System

```

+000081* OUTPUT FORMAT:EVOKPGM FROM FILE BSCELCIF OF LIBRARYBSCELIB <-ALL-FMTS
+000082* <-ALL-FMTS
75 +000083 05 EVOKPGM-0 REDEFINES BSCELCIF-RECORD. <-ALL-FMTS
76 +000084 06 PASS PIC X(4). <-ALL-FMTS
77 +000085 06 USERID PIC X(8). <-ALL-FMTS
78 +000086 06 LIB PIC X(8). <-ALL-FMTS
79 +000087 06 PGMID PIC X(8). <-ALL-FMTS
+000088* INPUT FORMAT:DETACH FROM FILE BSCELCIF OF LIBRARY BSCELIB <-ALL-FMTS
+000089* <-ALL-FMTS
+000090* 05 DETACH-I REDEFINES BSCELCIF-RECORD. <-ALL-FMTS
+000091* OUTPUT FORMAT:DETACH FROM FILE BSCELCIF OF LIBRARY BSCELIB <-ALL-FMTS
+000092* <-ALL-FMTS
+000093* 05 DETACH-O REDEFINES BSCELCIF-RECORD. <-ALL-FMTS
80 000280 FD BSCELPRT
81 000290 LABEL RECORDS ARE STANDARD
82 000300 DATA RECORD IS PRINT-RECORD.
83 000310 01 PRINT-RECORD.
84 000320 COPY DDS-ALL-FORMATS-0 OF BSCELPRT.
85 +000001 05 BSCELPRT-RECORD PIC X(79). <-ALL-FMTS
+000002* I-O FORMAT:HDG FROM FILE BSCELPRT OF LIBRARY BSCELIB <-ALL-FMTS
+000003* <-ALL-FMTS
+000004* 05 HDG REDEFINES BSCELPRT-RECORD. <-ALL-FMTS
+000005* OUTPUT FORMAT:DTL FROM FILE BSCELPRT OF LIBRARY BSCELIB <-ALL-FMTS
+000006* <-ALL-FMTS
86 +000007 05 DTL-0 REDEFINES BSCELPRT-RECORD. <-ALL-FMTS
87 +000008 06 CUSNUM PIC X(7). <-ALL-FMTS
+000009* Customer Number <-ALL-FMTS
88 +000010 06 CUSNAM PIC X(25). <-ALL-FMTS
+000011* Customer Name <-ALL-FMTS
89 +000012 06 ADDR PIC X(15). <-ALL-FMTS
+000013* Address <-ALL-FMTS
90 +000014 06 CITY PIC X(15). <-ALL-FMTS
+000015* City <-ALL-FMTS
91 +000016 06 STATE PIC X(2). <-ALL-FMTS
+000017* State <-ALL-FMTS
92 +000018 06 ZIP PIC S9(5). <-ALL-FMTS
+000019* Zip <-ALL-FMTS
93 +000020 06 CRDLMT PIC S9(5). <-ALL-FMTS
SMT SEQNBR -A 1 B.+. . . . 2. . . . +. . . . 3. . . . +. . . . 4. . . . +. . . . 5. . . . +. . . . 6. . . . +. . . . 7. . . . IDENTFCN S COPYNAME CHG DATE
+000021* Credit Limit <-ALL-FMTS
94 +000022 06 CRDAMT PIC S9(5). <-ALL-FMTS
+000023* Credit Amount <-ALL-FMTS
+000024* OUTPUT FORMAT:ERROR FROM FILE BSCELPRT OF LIBRARY BSCELIB <-ALL-FMTS
+000025* <-ALL-FMTS
95 +000026 05 ERROR-0 REDEFINES BSCELPRT-RECORD. <-ALL-FMTS
96 +000027 06 PGMDEV PIC X(10). <-ALL-FMTS
97 +000028 06 FMTNM PIC X(8). <-ALL-FMTS
98 +000029 06 MAJCOD PIC X(2). <-ALL-FMTS
99 +000030 06 MINCOD PIC X(2). <-ALL-FMTS
100 000330 WORKING-STORAGE SECTION.
101 000340 77 STATUS-IND PIC XX.
102 000350 77 INDON PIC 1 VALUE B"1".
103 000360 77 INDOFF PIC 1 VALUE B"0".
104 000370 01 PGM-INDIC-AREA.
105 000380 05 PGM-INDIC PIC 1 OCCURS 99 TIMES
106 000390 INDICATOR 1.
107 000400 01 TRAN-CTL-AREA.
108 000410 05 FILLER PIC X(2).
109 000420 05 PGM-DEV-NAME PIC X(10).
110 000430 05 RCD-FMT-NAME PIC X(10).
2 000440
111 000450 01 FEEDBACK-DATA.
112 000460 05 FILLER PIC X(20).
113 000470 05 FMTNM PIC X(10).
114 000480 05 FILLER PIC X(2).
115 000490 05 PGMDEV PIC X(10).
116 000500 05 FILLER PIC X(102).
117 000510 05 FILLER PIC X(34).
118 000520 05 MAJCOD PIC X(2).
119 000530 05 MINCOD PIC X(2).
000540
120 000550 01 MAJ-MIN.
121 000560 05 MAJ PIC X(2).
122 000570 05 MIN PIC X(2).
000580
123 000590 PROCEDURE DIVISION.
000600

```

Figure G-9 (Part 3 of 6). COBOL/400 Source Program for the Remote System

```

000610*****
000620* THE FOLLOWING DECLARATIVES SECTION IS AN ERROR ROUTINE THAT *
000630* IS RUN WHEN AN ERROR OCCURS ON THE READ OR THE WRITE OF THE *
000640* OS/400-ICF FILE "BSCSELICF". THE ROUTINE MOVES DATA FROM THE *
000650* I-O FEEDBACK AREA TO THE "ERROR" FORMAT OF THE PRINT FILE *
000660* "BSCSELPRT". ALL OF THE FILES ARE THEN CLOSED AND THE PROGRAM *
000670* IS TERMINATED. *
000680*****
000690 DECLARATIVES.
000700 ERROR-SECTION SECTION.
000710 USE AFTER STANDARD ERROR PROCEDURE ON OUTPUT.
9 000720 ERROR-PARAGRAPH.
124 000730 ACCEPT FEEDBACK-DATA FROM FEEDBACK-AREA.
125 000740 MOVE CORRESPONDING FEEDBACK-DATA TO ERROR-0.
* ** CORRESPONDING items for statement 125:
* ** FMTNM
* ** PGMDEV
STMT SEQNBR -A 1 B.+. . . . 2. . . . +. . . . 3. . . . +. . . . 4. . . . +. . . . 5. . . . +. . . . 6. . . . +. . . . 7. . . . IDENTFCN S COPYNAME CHG DATE
* ** MAJCOD
* ** MINCOD
* ** End of CORRESPONDING items for statement 125
126 000750 WRITE PRINT-RECORD FORMAT IS "ERROR".
127 000760 CLOSE BSCSELICF
000770 DBFILE
000780 BSCSELPRT.
128 000790 STOP RUN.
000800 END DECLARATIVES.
000810
000820 MAIN-PROGRAM SECTION.
000830
000840 000-PROCESS-GROUP.
3 000850
129 000860 OPEN I-O BSCSELICF.
130 000870 OPEN INPUT DBFILE.
131 000880 OPEN OUTPUT BSCSELPRT.
000890
132 000900 ACQUIRE "TARGET " FOR BSCSELICF.
133 000910 MOVE "TARGET " TO PGM-DEV-NAME.
000920
134 000930 MOVE ZEROES TO PGM-INDIC-AREA.
000940
135 000950 PERFORM 100-PAGE-HEADING.
000960
000970*****
4 000980* READ DATA FROM SOURCE SYSTEM.
000990*****
001000
136 001010 PERFORM 110-READ-DATA UNTIL PGM-INDIC(40) = INDON.
001020
001030*****
5 001040* READ DATA BASE FILE AND SEND IT TO SOURCE SYSTEM.
001050*****
001060
137 001070 PERFORM 120-SEND-DATA UNTIL PGM-INDIC(99) = INDON.
001080
001090*****
001100*
001110* THE FOLLOWING ROUTINE RECEIVES THE "DETACH" KEYWORD FROM THE *
001120* SOURCE PROGRAM CLOSES ALL OF THE FILES AND TERMINATES THE *
001130* PROGRAM. *
001140* *
6 001150*****
138 001160 READ BSCSELICF FORMAT IS "RCVDATA".
139 001170 CLOSE BSCSELICF
001180 BSCSELPRT
001190 DBFILE.
001200
140 001210 STOP RUN.
001220
001230* SUBROUTINES.
001240
001250 100-PAGE-HEADING.
141 001260 WRITE PRINT-RECORD FORMAT IS "HDG".

```

Figure G-9 (Part 4 of 6). COBOL/400 Source Program for the Remote System

```

STMT SEQNBR -A 1 B..+...2....+...3....+...4....+...5....+...6....+...7..IDENTFCN S COPYNAME CHG DATE
001270
001280*****
001290*
001300* THE FOLLOWING ROUTINE RECEIVES RECORDS FROM THE SOURCE SYSTEM *
001310* UNTIL INDICATOR 40 IS TURNED ON. INDICATOR 40 SIGNALS THAT A *
001320* 'RCVENDGRP' HAS BEEN RECEIVED, AND THE SOURCE PROGRAM HAS *
001330* STOPPED SENDING RECORDS. AS RECORDS ARE RECEIVED THEY ARE *
001340* PRINTED IN THE PRINT FILE 'BSCPLPRT'. *
001350*
001360*****
7 001370 110-READ-DATA.
142 001380 READ BSCCLICF FORMAT IS "RCVDATA"
001390 TERMINAL IS PGM-DEV-NAME
001400 INDICATORS ARE PGM-INDIC-AREA.
143 001410 IF PGM-INDIC(40) EQUAL INDOFF THEN
144 001420 MOVE CORRESPONDING RCVDATA-I TO DTL-0
* ** CORRESPONDING items for statement 144:
* ** CUSNUM
* ** CUSNAM
* ** ADDR
* ** CITY
* ** STATE
* ** ZIP
* ** CRDLMT
* ** CRDAMT
* ** End of CORRESPONDING items for statement 144
145 001430 WRITE PRINT-RECORD FORMAT IS "DTL"
146 001440 AT EOP PERFORM 100-PAGE-HEADING.
001450
001460*****
001470*
001480* THE FOLLOWING ROUTINE READS RECORDS FROM THE DATA BASE FILE *
001490* 'DBFILE'. WHEN 'DBFILE' REACHES 'END OF FILE' INDICATOR 99 *
001500* IS TURNED ON. *
001510*
001520*****
8 001530 120-SEND-DATA.
147 001540 READ DBFILE
148 001550 AT END MOVE INDON TO PGM-INDIC(99).
149 001560 MOVE CORRESPONDING DBFMT TO SNDDATA-0.
* ** CORRESPONDING items for statement 149:
* ** CUSNUM
* ** CUSNAM
* ** ADDR
* ** CITY
* ** STATE
* ** ZIP
* ** CRDLMT
* ** CRDAMT
* ** End of CORRESPONDING items for statement 149
150 001570 IF PGM-INDIC(99) EQUAL INDOFF THEN
151 001580 WRITE ICFREC FORMAT IS "SNDDATA"
001590 TERMINAL IS PGM-DEV-NAME.
001600
STMT SEQNBR -A 1 B..+...2....+...3....+...4....+...5....+...6....+...7..IDENTFCN S COPYNAME CHG DATE
* * * * * E N D O F S O U R C E * * * * *

```

Figure G-9 (Part 5 of 6). COBOL/400 Source Program for the Remote System

```

STMT
* 22 MSGID: LBL0650 SEVERITY: 00 SEQNBR: 000220
    Message . . . . : Blocking/Deblocking for file 'DBFILE' will
    be performed by compiler-generated code.
* 37 MSGID: LBL0600 SEVERITY: 10 SEQNBR: 000270
    Message . . . . : No INPUT fields found for format EVOKPGM.
* 37 MSGID: LBL0600 SEVERITY: 10 SEQNBR: 000270
    Message . . . . : No INPUT fields found for format DETACH.
* 37 MSGID: LBL0600 SEVERITY: 10 SEQNBR: 000270
    Message . . . . : No OUTPUT fields found for format DETACH.
* 84 MSGID: LBL0600 SEVERITY: 10 SEQNBR: 000320
    Message . . . . : No INPUT fields found for format HDG.
    * * * * * E N D   O F   M E S S A G E S   * * * * *
                                Message Summary
Total      Info(0-4)   Warning(5-19)   Error(20-29)   Severe(30-39)   Terminal(40-99)
   5         1         4             0             0             0
Source records read . . . . . : 160
Copy records read . . . . . : 143
Copy members processed . . . . . : 3
Sequence errors . . . . . : 0
Highest severity message issued . . : 10
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 G-9 (Part 6 of 6). 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 binary synchronous 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/QDDSSRC)
SRCMBR(BSCELICF) TEXT('ICF FILE FOR BSCEL')

```

The DDS used by this file is shown in Figure G-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 G-10. DDS for the BSCSELICF File Used by the Local System (RPG/400 Language)

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 G-11 on page G-23.

```

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 G-11. DDS for the Database File Used in BSCCEL Data Transfer on the Local System (RPG/400 Language)

**BSCCLPRT** The printer file used to format The DDS for this file is shown in Figure G-12.  
output to a printer.

```

Data Description Source
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
          * * * * * E N D O F S O U R C E * * * * *

```

Figure G-12. DDS for the Printer File on the Local System (RPG/400 Language)

## Program Explanation

The following describes the RPG/400 program on the local system, which is shown in Figure G-13 on page G-25.

- 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 program start request and issues the write operation to 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.



```

Compiler . . . . . : IBM AS/400 RPG/400
Command Options:
  Program . . . . . : BSCELIB/RPGELSRC
  Source file . . . . . : BSCELIB/QRPGSRC
  Source member . . . . . : *PGM
Text not available for message RXT0073 file QRPGMSG.
Generation options . . . . . : *NOLIST *NOXREF *NOATR *NODUMP *NOOPTIMIZE
Source listing indentation . . . . . : *NONE
SAA flagging . . . . . : *NOFLAG
Generation severity level . . . . . : 9
Print file . . . . . : *LIBL/QSYSVRT
Replace program . . . . . : *YES
Target release . . . . . : *CURRENT
User profile . . . . . : *USER
Authority . . . . . : *LIBCRTAUT
Text . . . . . : *SRCMBRTXT
Phase trace . . . . . : *NO
Intermediate text dump . . . . . : *NONE
Snap dump . . . . . : *NONE
Codelist . . . . . : *NONE
Ignore decimal data error . . . . . : *NO
Actual Program Source:
Member . . . . . : RPGELSRC
File . . . . . : QRPGSRC
Library . . . . . : BSCELIB
Last Change . . . . . : 10/17/90 13:29:45
Description . . . . . : Source RPG program for BSCEL
SEQUENCE
NUMBER *...1....+....2....+....3....+....4....+....5....+....6....+....7...*  IND DO LAST PAGE PROGRAM
                                         USE NUM UPDATE LINE ID
                                         S o u r c e L i s t i n g
H
1 10 FBSCELICFCF E WORKSTN *****
20 F KINFDS FEEDBK
30 F KINFSR FAIL
40 F KNUM 1
50 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
60 FDBFILE IF E K DISK
RECORD FORMAT(S): LIBRARY BSCELIB FILE DBFILE.
EXTERNAL FORMAT DBFMT RPG NAME DBFMT
70 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 Customer Number
A000002 8 32 CUSNAM Customer Name
A000003 33 47 ADDR Address
A000004 48 62 CITY City
A000005 63 64 STATE State
A000006 65 690ZIP Zip
A000007 70 740CRDLMT Credit Limit
A000008 75 790CRDAMT Credit Amount
B000000 INPUT FIELDS FOR RECORD SNDDATA FILE BSCELICF FORMAT SNDDATA.
B000001 1 7 CUSNUM Customer Number
B000002 8 32 CUSNAM Customer Name
B000003 33 47 ADDR Address
B000004 48 62 CITY City
B000005 63 64 STATE State
B000006 65 690ZIP Zip
B000007 70 740CRDLMT Credit Limit
B000008 75 790CRDAMT Credit Amount
C000000 INPUT FIELDS FOR RECORD EVOKPGM FILE BSCELICF FORMAT EVOKPGM.
D000000 INPUT FIELDS FOR RECORD DETACH FILE BSCELICF FORMAT DETACH.
E000000 INPUT FIELDS FOR RECORD DBFMT FILE DBFILE FORMAT DBFMT.

```

Figure G-13 (Part 1 of 4). RPG/400 Source Program for the Local System

```

E000001          1  7 CUSNUM          Customer Number
E000002          8 32 CUSNAM          Customer Name
E000003         33 47 ADDR            Address
E000004         48 62 CITY            City
E000005         63 64 STATE           State
E000006         65 690ZIP            Zip
E000007         70 740CRDLMT         Credit Limit
SEQUENCE          IND  DO  LAST  PAGE  PROGRAM
NUMBER          *...1...+...2...+...3...+...4...+...5...+...6...+...7...*  USE  NUM  UPDATE  LINE  ID
E000008          75 790CRDAMT         Credit Amount
2  80  IFEEBKB      DS
  90  I              38 45 FMTNM
 100  I              273 282 PGMDEV
 110  I              401 402 MAJCOD
 120  I              403 404 MINCOD
 130  C*****
 140  C* START PROGRAM *
 150  C* *
 160  C* FILES ARE IMPLICITLY OPENED, AND THE BSCCL PROGRAM DEVICE *
 170  C* IS ACQUIRED. *
 180  C* *
 190  C*****
3  200  C          'SOURCE' ACQ BSCCLICF
 210  C              MOVEL'SOURCE' PGMDEV 10
 220  C*****
 230  C* *
 240  C* THE FOLLOWING ROUTINE STARTS THE PROGRAM 'RPGELTGT' ON THE *
 250  C* TARGET SYSTEM. *
 260  C* *
 270  C*****
4  280  C              MOVEL'BSCEL' PASS
 290  C              MOVEL'BSCEL' USERID
 300  C              MOVEL'BSCELIB' LIB
 310  C              MOVEL'RPGL' PGMID
 320  C              MOVE 'TGT' PGMID
 330  C              WRITEEVOKPGM
 340  C*****
 350  C* *
 360  C* THE FOLLOWING ROUTINE READS RECORDS FROM THE DATA BASE FILE *
 370  C* 'DBFILE'. WHEN 'DBFILE' REACHES END OF FILE, INDICATOR 99 *
 380  C* IS TURNED ON. CONDITION N99 IS USED ON THE WRITE TO AVOID *
 390  C* SENDING A BLANK RECORD. **
 400  C* *
 410  C*****
5  420  C          *IN99  DOWEQ'1'          B001          B001
 430  C              READ DBFMT          99          3 001
 440  C N99          WRITESNDATA
 450  C              END          E001          E001
 460  C*****
 470  C* *
 480  C* THE READ RCVDATA IN THE NEXT SECTION OF THE PROGRAM WILL *
 490  C* CAUSE THE END-OF-TRANSMISSION TO BE SENT BEFORE RECEIVING *
 500  C* THE FIRST DATA RECORD. RECORDS WILL BE RECEIVED FROM THE *
 510  C* TARGET SYSTEM UNTIL AN END-OF-TRANSMISSION IS RECEIVED. THIS *
 520  C* IS DEFINED AS A RCVENDGRP, SO INDICATOR 40 WILL BE SET ON. *
 530  C* AS RECORDS ARE RECEIVED THEY ARE PRINTED IN THE PRINT FILE *
 540  C* 'BSCCLPRT'. INDICATOR 66 SIGNALS PAGE OVERFLOW. *
 550  C* *
 560  C*****
6  570  C              WRITEHDG
 580  C              READ RCVDATA          98          3
SEQUENCE          IND  DO  LAST  PAGE  PROGRAM
NUMBER          *...1...+...2...+...3...+...4...+...5...+...6...+...7...*  USE  NUM  UPDATE  LINE  ID
590  C          *IN40  DOWEQ'0'          B001          B001
600  C              WRITEDTL          001
610  C          *IN66  IFEQ '1'          B002          B002
620  C              WRITEHDG          002
630  C              MOVE '0'          *IN66          002
640  C              END          E002          E002
650  C              READ RCVDATA          98          3 001
660  C              END          E001          E001

```

Figure G-13 (Part 2 of 4). RPG/400 Source Program for the Local System

```

670 C*****
680 C*
690 C* THE FOLLOWING ROUTINE WRITES A DETACH TO THE TARGET SYSTEM. *
700 C* WHEN THE TARGET SYSTEM RECEIVES THE DETACH, THE CONVERSATION *
710 C* BETWEEN THE SOURCE AND TARGET SYSTEMS WILL BE TERMINATED. *
720 C* SINCE NO ADDITIONAL PROCESSING IS NEEDED IN THIS PROGRAM, THE *
730 C* LR INDICATOR IS SET ON AND THE PROGRAM IS ENDED. *
740 C*
750 C*****
760 C          WRITEDETACH
770 C          MOVE '1'          *INLR
780 C*****
790 C*
800 C*FAIL SUBROUTINE
810 C*
820 C*****
830 C          FAIL          BEGSR
840 C          WRITEERROR
850 C          ENDSR'*CANCL'
F000000 OUTPUT FIELDS FOR RECORD SNDDATA FILE BSCELICF FORMAT SNDDATA.
F000001          CUSNUM          7 CHAR          7          Customer Number
F000002          CUSNAM          32 CHAR         25          Customer Name
F000003          ADDR           47 CHAR         15          Address
F000004          CITY           62 CHAR         15          City
F000005          STATE          64 CHAR          2          State
F000006          ZIP            69 ZONE         5,0         Zip
F000007          CRDLMT         74 ZONE         5,0         Credit Limit
F000008          CRDAMT         79 ZONE         5,0         Credit Amount
G000000 OUTPUT FIELDS FOR RECORD EVOKPGM FILE BSCELICF 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 BSCELICF FORMAT DETACH.
I000000 OUTPUT FIELDS FOR RECORD HDG FILE BSCELPRF FORMAT HDG.
J000000 OUTPUT FIELDS FOR RECORD DTL FILE BSCELPRF FORMAT DTL.
J000001          CUSNUM          7 CHAR          7          Customer Number
J000002          CUSNAM          32 CHAR         25          Customer Name
J000003          ADDR           47 CHAR         15          Address
J000004          CITY           62 CHAR         15          City
J000005          STATE          64 CHAR          2          State
J000006          ZIP            69 ZONE         5,0         Zip
J000007          CRDLMT         74 ZONE         5,0         Credit Limit
J000008          CRDAMT         79 ZONE         5,0         Credit Amount
K000000 OUTPUT FIELDS FOR RECORD ERROR FILE BSCELPRF FORMAT ERROR.
K000001          PGMDEV         10 CHAR          10
SEQUENCE
NUMBER *...1....+...2....+...3....+...4....+...5....+...6....+...7...*  IND  DO  LAST  PAGE  PROGRAM
                                         USE  NUM UPDATE  LINE  ID
K000002          FMTNM          18 CHAR          8
K000003          MAJCOD         20 CHAR          2
K000004          MINCOD         22 CHAR          2
***** END OF SOURCE *****
Additional Diagnostic Messages
* 7089 10 RPG PROVIDES SEPARATE INDICATOR AREA FOR FILE BSCELICF.
* 7086 60 RPG PROVIDES BLOCK OR UNBLOCK SUPPORT FOR FILE DBFILE.
Key Field Information
      PHYSICAL LOGICAL
FILE/RCD FIELD FIELD ATTRIBUTES
02 DBFILE
   DBFMT
       CUSNUM          CHAR 7
Cross Reference
File and Record References:
FILE/RCD DEV/RCD REFERENCES (D=DEFINED)
01 BSCELICF WORKSTN          100 200
   DETACH          100 D000000 760 H000000
   EVOKPGM         100 C000000 330 G000000
   RCVDATA         100 A000000 580 650
   SNDDATA         100 B000000 440 F000000
03 BSCELPRF PRINTER          700
   DTL             700 600 J000000
   ERROR          700 840 K000000
   HDG            700 570 620 I000000
02 DBFILE DISK              600
   DBFMT          600 E000000 430

```

Figure G-13 (Part 3 of 4). RPG/400 Source Program for the Local System

```

Field References:
FIELD  ATTR  REFERENCES (M=MODIFIED D=DEFINED)
*INLR  A(1)   770M
*IN40  A(1)   590
*IN66  A(1)   610    630M
*IN99  A(1)   420
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  10     830D
FEEDBK DS(404) 10     80D
FMTNM  A(8)   90D K000002D
LIB    A(8)   300M G000003D
MAJCOD A(2)   110D K000003D
MINCOD A(2)   120D K000004D
PASS   A(4)   280M G000001D
PGMDEV A(10)  100D 210D K000001D
PGMID  A(8)   310M 320M G000004D
STATE  A(2)   A000005D B000005D E000005D F000005D J000005D
USERID A(8)   290M G000002D
ZIP    P(5,0) A000006D B000006D E000006D F000006D J000006D
'*CANCL' LITERAL 850
'BSCE'  LITERAL 280
'BSCEL' LITERAL 290
'BSCELIB' LITERAL 300
'RPSEL' LITERAL 310
'SOURCE' LITERAL 200    210
'TGT'   LITERAL 320
'0'     LITERAL 590    630
'1'     LITERAL 420    610    770

```

```

Indicator References:
INDICATOR REFERENCES (M=MODIFIED D=DEFINED)
*IN      420    590    610    630M
LR       770M
* 7031  35
* 7031  38
      40    590
      66    70D    610    630M
* 7031  98    580M    650M
      99    420    430M    440

```

```

***** END OF C R O S S R E F E R E N C E *****
                M e s s a g e S u m m a r y

```

```

* 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 M E S S A G E S U M M A R Y *****
                F i n a l S u m m a r y

```

```

Message Count: (by Severity Number)
TOTAL  00  10  20  30  40  50
       5   5   0   0   0   0

```

```

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 C O M P I L A T I O N *****

```

Figure G-13 (Part 4 of 4). 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 BSCCEL target program.

### Configuration

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

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

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

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

### Program Files

The following files are used by the remote system:

**BSCCELICF** 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(BSCCELIB/BSCCELICF)
  SRCFILE(BSCCELIB/QDDSSRC)
  SRCMBR(BSCCELICF) TEXT('ICF FILE FOR BSCCEL')
```

The DDS used by this file is shown in Figure G-14.

```

                                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*
```

Figure G-14 (Part 1 of 2). DDS for the BSCCELICF Program File Used by the Remote System (RPG/400 Language)

```

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
4100 A          SECURITY(2 &PASS 3 &USERID)
4200 A          EVOKE(&LIB/&PGMID)
4300 A          PASS        4A
4400 A          USERID     8A
4500 A          LIB         8A
4600 A          PGMID       8A
          R DETACH          DETACH
          * * * * * E N D   O F   S O U R C E   * * * * *

```

Figure G-14 (Part 2 of 2). DDS for the BSCELICF Program File Used by the Remote System (RPG/400 Language)

The command needed to define the program device entry is:

```

ADDICFDEVE FILE(BSCCELIB/BSCELICF)
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 G-15.

```

Data Description Source
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 .
100 A*****
200 A*          * * * DBFILE * * *          *
300 A*          DDS FOR DATABASE FILE USED IN BSCCEL DATA TRANSFER PROGRAMS. *
400 A*          * * * * *          *
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 G-15. DDS for Database File Used in BSCEL Data Transfer Programs on the Remote System (RPG/400 Language)

**BSCCLPRT** The printer file used to format output to a printer.

The DDS for this file is shown in Figure G-16.

```

                                Data Description Source
SEQNBR *...+...1...+...2...+...3...+...4...+...5...+...6...+...7.
 100    A                                REF(DBFILE)
 200    A                                SKIPB(5)
 300    A      R HDG                      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 G-16. DDS for the Printer File on the Remote System (RPG/400 Language)

## Program Explanation

The following describes the RPG/400 program on the remote system, which is shown in Figure G-17 on page G-33.

- 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 processing. 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.



```

Compiler . . . . . : IBM AS/400 RPG/400
Command Options:
  Program . . . . . : BSCSELIB/RPGELTGT
  Source file . . . . . : BSCSELIB/QRPGSRC
  Source member . . . . . : *PGM
Text not available for message RXT0073 file QRPGMSG.
Generation options . . . . . : *NOLIST *NOXREF *NOATR *NODUMP *NOOPTIMIZE
Source listing indentation . . . . . : *NONE
SAA flagging . . . . . : *NOFLAG
Generation severity level . . . . . : 9
Print file . . . . . : *LIBL/QSYSVRT
Replace program . . . . . : *YES
Target release . . . . . : *CURRENT
User profile . . . . . : *USER
Authority . . . . . : *LIBCRTAUT
Text . . . . . : *SRCMBRTXT
Phase trace . . . . . : *NO
Intermediate text dump . . . . . : *NONE
Snap dump . . . . . : *NONE
Codelist . . . . . : *NONE
Ignore decimal data error . . . . . : *NO
Actual Program Source:
Member . . . . . : RPGELTGT
File . . . . . : QRPGSRC
Library . . . . . : BSCSELIB
Last Change . . . . . : 10/16/90 16:08:04
Description . . . . . : Target RPG program for BSCSEL
SEQUENCE
NUMBER *...1....+....2....+....3....+....4....+....5....+....6....+....7...* IND DO LAST PAGE PROGRAM
      USE NUM UPDATE LINE ID
      S o u r c e L i s t i n g
      *****
1 10 H FBSCSELICFCF E WORKSTN
20 F KINFDS FEEDBK
30 F KINFSR FAIL
40 F KNUM 1
RECORD FORMAT(S): LIBRARY BSCSELIB FILE BSCSELICF.
EXTERNAL FORMAT RCVDATA RPG NAME RCVDATA
EXTERNAL FORMAT SNDDATA RPG NAME SNDDATA
EXTERNAL FORMAT EVOKPGM RPG NAME EVOKPGM
EXTERNAL FORMAT DETACH RPG NAME DETACH
50 FDBFILE IF E K DISK
RECORD FORMAT(S): LIBRARY BSCSELIB FILE DBFILE.
EXTERNAL FORMAT DBFMT RPG NAME DBFMT
60 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 BSCSELICF FORMAT RCVDATA.
A000001 1 7 CUSNUM Customer Number
A000002 8 32 CUSNAM Customer Name
A000003 33 47 ADDR Address
A000004 48 62 CITY City
A000005 63 64 STATE State
A000006 65 690ZIP Zip
A000007 70 740CRDLMT Credit Limit
A000008 75 790CRDAMT Credit Amount
B000000 INPUT FIELDS FOR RECORD SNDDATA FILE BSCSELICF FORMAT SNDDATA.
B000001 1 7 CUSNUM Customer Number
B000002 8 32 CUSNAM Customer Name
B000003 33 47 ADDR Address
B000004 48 62 CITY City
B000005 63 64 STATE State
B000006 65 690ZIP Zip
B000007 70 740CRDLMT Credit Limit
B000008 75 790CRDAMT Credit Amount
C000000 INPUT FIELDS FOR RECORD EVOKPGM FILE BSCSELICF FORMAT EVOKPGM.
D000000 INPUT FIELDS FOR RECORD DETACH FILE BSCSELICF FORMAT DETACH.
E000000 INPUT FIELDS FOR RECORD DBFMT FILE DBFILE FORMAT DBFMT.
E000001 1 7 CUSNUM Customer Number
E000002 8 32 CUSNAM Customer Name
E000003 33 47 ADDR Address
E000004 48 62 CITY City
E000005 63 64 STATE State
E000006 65 690ZIP Zip
E000007 70 740CRDLMT Credit Limit
E000008 75 790CRDAMT Credit Amount

```

Figure G-17 (Part 1 of 4). RPG/400 Source Program for the Remote System

```

SEQUENCE          IND  DO  LAST  PAGE  PROGRAM
NUMBER  *...1...+...2...+...3...+...4...+...5...+...6...+...7...*  USE  NUM  UPDATE  LINE  ID
2  70  IFEEDBK      DS
80  I              38 45 FMTNM
90  I              273 282 PGMDEV
100 I              401 402 MAJCOD
110 I              403 404 MINCOD
120 C*****
130 C* START PROGRAM. *
140 C* ACQUIRE PROGRAM DEVICE. *
3  150 C          'TARGET' ACQ BSCCELICF
160 C*****
170 C*
180 C* THE FOLLOWING ROUTINE RECEIVES RECORDS FROM THE SOURCE SYSTEM *
190 C* UNTIL INDICATOR 40 IS TURNED ON. INDICATOR 40 SIGNALS THAT A *
200 C* 'RCVENDGRP' HAS BEEN RECEIVED, AND THE SOURCE PROGRAM HAS *
210 C* STOPPED SENDING RECORDS. AS RECORDS ARE RECEIVED THEY ARE *
220 C* PRINTED IN THE PRINT FILE 'BSCCLPRT'. INDICATOR 66 SIGNALS *
230 C* PAGE OVERFLOW. *
240 C* *
250 C*****
4  260 C          WRITEHDG
270 C          READ RCVDATA          99          3
280 C          *IN40  DOWEQ'0'          B001          B001
290 C          WRITEDTL          001
300 C          *IN66  IFEQ '1'          B002          B002
310 C          WRITEHDG          002
320 C          MOVE '0'          *IN66          002
330 C          END          E002          E002
340 C          READ RCVDATA          99          3 001
350 C          END          E001          E001
360 C*****
370 C*
380 C* THE FOLLOWING ROUTINE READS RECORDS FROM THE DATA BASE FILE *
390 C* 'DBFILE'. WHEN 'DBFILE' REACHES 'END OF FILE' INDICATOR 99 *
400 C* IS TURNED ON. CONDITION N99 IS USED ON THE WRITE TO AVOID *
410 C* SENDING A BLANK RECORD. *
420 C* *
430 C*****
5  440 C          *IN99  DOUEQ'1'          B001          B001
450 C          READ DBFMT          99          3 001
460 C N99          WRITESNDDATA
470 C          END          E001          E001
480 C*****
490 C*
500 C* THE FOLLOWING READ CAUSES THE SENDING OF END-OF-TRANSMISSION *
510 C* TO THE SOURCE SYSTEM, AND RECEIVES THE 'DETACH' FROM THE *
520 C* SOURCE SYSTEM. AFTER DETACH IS RECEIVED, NO OTHER PROCESSING *
530 C* IS NEEDED SO THE LR INDICATOR IS SET ON AND THE PROGRAM ENDS. *
540 C* *
550 C*****
6  560 C          READ RCVDATA          99          3
570 C          MOVE '1'          *INLR
580 C*****
SEQUENCE          IND  DO  LAST  PAGE  PROGRAM
NUMBER  *...1...+...2...+...3...+...4...+...5...+...6...+...7...*  USE  NUM  UPDATE  LINE  ID
590 C*
600 C* FAIL SUBROUTINE
610 C*
620 C*****
7  630 C          FAIL  BEGSR
640 C          WRITEERROR
650 C          ENDSR'*CANCL'
F000000 OUTPUT FIELDS FOR RECORD SNDDATA FILE BSCCELICF FORMAT SNDDATA.
F000001          CUSNUM  7  CHAR  7          Customer Number
F000002          CUSNAM 32  CHAR 25          Customer Name
F000003          ADDR   47  CHAR 15          Address
F000004          CITY   62  CHAR 15          City
F000005          STATE  64  CHAR  2          State
F000006          ZIP    69  ZONE 5,0        Zip
F000007          CRDLMT 74  ZONE 5,0        Credit Limit
F000008          CRDAMT 79  ZONE 5,0        Credit Amount

```

Figure G-17 (Part 2 of 4). RPG/400 Source Program for the Remote System

```

G000000 OUTPUT FIELDS FOR RECORD HDG FILE BSCELPRT FORMAT HDG.
H000000 OUTPUT FIELDS FOR RECORD DTL FILE BSCELPRT FORMAT DTL.
H000001 CUSNUM 7 CHAR 7 Customer Number
H000002 CUSNAM 32 CHAR 25 Customer Name
H000003 ADDR 47 CHAR 15 Address
H000004 CITY 62 CHAR 15 City
H000005 STATE 64 CHAR 2 State
H000006 ZIP 69 ZONE 5,0 Zip
H000007 CRDLMT 74 ZONE 5,0 Credit Limit
H000008 CRDAMT 79 ZONE 5,0 Credit Amount
I000000 OUTPUT FIELDS FOR RECORD ERROR FILE BSCELPRT 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 10 RPG PROVIDES SEPARATE INDICATOR AREA FOR FILE BSCELCIF.
* 7086 50 RPG PROVIDES BLOCK OR UNBLOCK SUPPORT FOR FILE DBFILE.
Key Field Information
PHYSICAL LOGICAL
FILE/RCD FIELD FIELD ATTRIBUTES
02 DBFILE
DBFMT
CUSNUM CHAR 7
Cross Reference
File and Record References:
FILE/RCD DEV/RCD REFERENCES (D=DEFINED)
01 BSCELCIF WORKSTN 10D 150
DETACH 10D D000000
EVOKPGM 10D C000000
RCVDATA 10D A000000 270 340 560
SNDDATA 10D B000000 460 F000000
03 BSCELPRT PRINTER 60D
DTL 60D 290 H000000
ERROR 60D 640 I000000
HDG 60D 260 310 G000000
02 DBFILE DISK 50D
DBFMT 50D E000000 450
Field References:
FIELD ATTR REFERENCES (M=MODIFIED D=DEFINED)
*INLR A(1) 570M
*IN40 A(1) 280
*IN66 A(1) 300 320M
*IN99 A(1) 440
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 10 630D
FEEDBK DS(404) 10 70D
FMTNM A(8) 80D I000002D
MAJCOD A(2) 100D I000003D
MINCOD A(2) 110D I000004D
PGMDEV A(10) 90D I000001D
STATE A(2) A000005D B000005D E000005D F000005D H000005D
ZIP P(5,0) A000006D B000006D E000006D F000006D H000006D
'*CANCL' LITERAL 650
'TARGET' LITERAL 150
'0' LITERAL 280 320
'1' LITERAL 300 440 570
Indicator References:
INDICATOR REFERENCES (M=MODIFIED D=DEFINED)
*IN 280 300 320M 440
LR 570M
* 7031 35
* 7031 38
40 280
66 60D 300 320M
99 270M 340M 440 450M 460 560M
***** END OF CROSS REFERENCE *****

```

Figure G-17 (Part 3 of 4). RPG/400 Source Program for the Remote System

```

                M e s s a g e   S u m m a r y
* QRG7031 Severity: 00   Number: 2
      Message . . . . : The Name or indicator is not referenced.
* QRG7086 Severity: 00   Number: 1
      Message . . . . : The RPG handles blocking function for file.
                        INFDS contents updated only when blocks of data transferred.
* QRG7089 Severity: 00   Number: 1
      Message . . . . : The RPG provides Separate-Indicator area for
                        file.
      * * * * *   E N D   O F   M E S S A G E   S U M M A R Y   * * * * *
                F i n a l   S u m m a r y
Message Count: (by Severity Number)
      TOTAL  00   10   20   30   40   50
             4    4    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 BSCSELIB. 00 highest Error-Severity-Code.
      * * * * *   E N D   O F   C O M P I L A T I O N   * * * * *

```

Figure G-17 (Part 4 of 4). RPG/400 Source Program for the Remote System

## ILE C/400 Source Program for Local System

The following describes the objects on the local system needed to run the ILE C/400 BSCSEL source program.

### Configuration

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

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

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

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

### Program Files

The following files are used by the local system:

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

```
CRDICFF FILE(BSCSELIB/SRCICFF)
      SRCFILE(BSCSELIB/QDDSSRC)
      SRCMBR(SRCICFF) TEXT('ICF FILE FOR BSCSEL')
```

The DDS used by this file is shown in Figure G-18 on page G-37.

```

                                Data Description Source
SEQNBR *...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8 Date
100    A*****
200    A*
300    A*          ICF FILE: SRCICFF          *
400    A*          USED IN C EXAMPLE 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 *USER 3 *USER)
4100   A          EVOKE(BSCSELIB/CTGTPGM)
4200   A          R DETACH          DETACH
                                * * * * * E N D   O F   S O U R C E   * * * * *

```

Figure G-18. DDS for the SRCICFF File Used by the Local System (ILE C/400 Language)

The command needed to define the program device entry is:

```

ADDICFDEVE FILE(BSCSELIB/SRCICFF)
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 G-19 on page G-38.

```

                                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 G-19. DDS for the Database File Used in BSCCEL Data Transfer on the Local System (ILE C/400 Language)

**BSCCLPRT** The printer file used to format output to a printer.

The DDS for this file is shown in Figure G-20.

```

                                Data Description Source
SEQNBR *...+...1...+...2...+...3...+...4...+...5...+...6...+...7.
100      A                                REF(DBFILE)
200      A                                SKIPB(5)
300      A      R HDG                      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 G-20. DDS for the Printer File on the Local System (ILE C/400 Language)

## Program Explanation

The following describes the ILE C/400 program on the local system, which is shown in Figure G-21 on page G-41.

- 1** The externally described formats are included in the program.
- 2** The internal routines are prototyped so the compiler knows the type of value returned by the routine, and the type of parameters passed to the routine, if any.
- 3** The database, ICF, and printer files are opened. The files are opened for record I/O with the separate indicator area option specified for the ICF file.
- 4** The separate indicator area array `s` is initialized to 0, and the separate indicator area is established with the `_Rindara` function. The program device `SOURCE` is acquired to establish an ICF session.
- 5** Record format `EVOKPGM` defined in `SRCICFF` is made the default record format for the following write operation. The remote program is started with a write using the evoke record format, `EVOKPGM`.
- 6** Function `send_data()` is called to send customer records found in the database file, `DBFILE`, to the remote program.
- 7** Procedure `print_heading()` is called to write a heading to the spooled file, and then `read_data()` is called to receive customer records from the remote program until the `RCVENDGRP` response indicator is set.
- 8** A detach is sent to the remote program to notify it that this program is ending.
- 9** Function `send_data()` sends customer records which are read from `DBFILE`, until end of file is reached.
- 10** The `while` loop ends after the last record in `DBFILE` is read, or, if an error occurs in the transaction, control returns to `main()`.
- 11** Procedure `print_heading()` issues a write to printer file `BSCELPRT` to print the heading in record format `HDG` to a spooled file.
- 12** Function `read_data()` reads records from the remote system until the `RCVENDGRP` response indicator is set in the separate indicator area.
- 13** The first read to the ICF file is performed with record format `RCVDATA`. The separate indicator area array is reset to 0 before the read is done.
- 14** The `while` loop ends when the `RCVENDGRP` indicator is set after a read is issued to the ICF file. `RCVENDGRP` is defined as indicator 40 in `SRCICFF`, but since the first position in the separate indicator area array starts at array index 0, the index for `RCVENDGRP` is at 39.
- 15** Function `check_error()` checks if the last input or output operation issued to the ICF file was successful or not. If the major return code is greater than or equal to 04, the program ends.
- 16** Procedure `end_error()` gets the program device and record format names from the display/ICF I/O feedback area. Data is written to a spooled file using record format `ERROR` defined in `BSCELPRT`, and the files are closed with a call to `close_files()`.
- 17** Procedure `close_files()` closes files `DBFILE`, `SRCICFF`, and `BSCELPRT`.
- 18** Pointers are set to the common and the display/ICF I/O feedback areas.



**1**

```
#pragma mapinc("icff/snd", "bscelib/srcicff(snddata)", "output", "p z")
#pragma mapinc("icff/rcv", "bscelib/srcicff(rcvdata)", "input", "p z")
#include "icff/snd"
#include "icff/rcv"
#pragma mapinc("dbf/fmt", "bscelib/dbfile(dbfmt)", "input", "p z")
#include "dbf/fmt"
#pragma mapinc("prtf/dtl", "bscelib/bscelprt(dtl)", "output", "p z")
#pragma mapinc("prtf/err", "bscelib/bscelprt(error)", "output", "p z")
#include "prtf/dtl"
#include "prtf/err"

/*-----*/
/* SOURCE PROGRAM FOR BSCCEL */
/*-----*/

#define NOERROR 0          /* No error occurred */
#define ERROR 1           /* An error occurred */
#define OFF '0'           /* Indicator off */
#define ON '1'            /* Indicator on */
#include <stdio.h>         /* Standard I/O header */
#include <recio.h>         /* Record I/O header */
#include <stddef.h>        /* Standard definitions */
#include <stdlib.h>        /* General utilities */
#include <string.h>        /* String handling utilities */
#include <xxfdbk.h>        /* Feedback area structures */

BSCCELIB_SRCICFF_SNDDATA_o_t  snddata_icf_o;
BSCCELIB_SRCICFF_RCVDATA_i_t rcvdata_icf_i;
BSCCELIB_DBFILE_DBFMT_i_t   dbfmt_db_i;
BSCCELIB_BSCCELPRT_DTL_o_t  dtl_prt_o;
BSCCELIB_BSCCELPRT_ERROR_o_t error_prt_o;

_RFILE *dbfptr;           /* Pointer to the database file */
_RFILE *icffptr;          /* Pointer to the ICF file */
_RFILE *prtfptr;          /* Pointer to the printer file */
_XXIOFB_T *comm_fdbk;     /* Pointer to common I/O feedback */
_XXIOFB_DSP_ICF_T *dsp_icf_fdbk; /* Pointer to dsp/ICF I/O fdbk */
char s??(99??);
```

**2**

```
int send_data(void);
void print_heading(void);
int read_data(void);
int check_error(void);
void end_error(void);
void close_files(void);
void get_access_to_fb(void);
```

Figure G-21 (Part 1 of 4). ILE C/400 Source Program for the Local System

```

main()
{
  3 if ((dbfptr = _Ropen("DBFILE", "rr riofb=y")) == NULL) {
    printf("Database file failed to open.\n");
    exit(ERROR);
  }
  if ((icffptr = _Ropen("SRCICFF", "ar+ indicators=y")) == NULL) {
    printf("ICF file failed to open.\n");
    _Rclose(dbfptr);
    exit(ERROR);
  }
  if ((prtfptr = _Ropen("BSCPLPRT", "wr")) == NULL) {
    printf("Printer file failed to open.\n");
    _Rclose(dbfptr);
    _Rclose(icffptr);
    exit(ERROR);
  }
  4 memset(s, '0', 99);
  _Rindara(icffptr, s);
  _Racquire(icffptr, "SOURCE");          /* Acquire a session */
  if (check_error() == ERROR) {
    end_error();
    exit(ERROR);
  }
  5 _Rformat(icffptr, "EVOKPGM");
  _Rwrite(icffptr, NULL, 0);
  if (check_error() == ERROR) {
    end_error();
    exit(ERROR);
  }
  6 if (send_data() == ERROR) {
    end_error();
    exit(ERROR);
  }
  7 print_heading();
  if (read_data() == ERROR) {
    end_error();
    exit(ERROR);
  }
  8 _Rformat(icffptr, "DETACH");
  _Rwrite(icffptr, NULL, 0);
  close_files();
}

```

Figure G-21 (Part 2 of 4). ILE C/400 Source Program for the Local System

```

/*-----*/
/* This routine reads records from the database file 'DBFILE' until the */
/* end of file is reached. The records read in are sent to the target */
/* program. */
/*-----*/
9
send_data()
{
    _RIOFB_T *fb;

    fb = _Readf(dbfptr, &dbfmt_db_i, sizeof(dbfmt_db_i), __DFT);
10
    while (fb->num_bytes > 0) {
        strncpy(snddata_icf_o.CUSNUM, dbfmt_db_i.CUSNUM, 7);
        strncpy(snddata_icf_o.CUSNAM, dbfmt_db_i.CUSNAM, 25);
        strncpy(snddata_icf_o.ADDR, dbfmt_db_i.ADDR, 15);
        strncpy(snddata_icf_o.CITY, dbfmt_db_i.CITY, 15);
        strncpy(snddata_icf_o.STATE, dbfmt_db_i.STATE, 2);
        strncpy(snddata_icf_o.ZIP, dbfmt_db_i.ZIP, 5);
        strncpy(snddata_icf_o.CRDLMT, dbfmt_db_i.CRDLMT, 5);
        strncpy(snddata_icf_o.CRDAMT, dbfmt_db_i.CRDAMT, 5);
        _Rformat(icffptr, "SNDDATA");
        _Rwrite(icffptr, &snddata_icf_o, sizeof(snddata_icf_o));
        if (check_error() == ERROR)
            return(ERROR);
        fb = _Readn(dbfptr, &dbfmt_db_i, sizeof(dbfmt_db_i), __DFT);
    }
    return(NOERROR);
}

/*-----*/
/* Print heading to the print file. */
/*-----*/
11
void print_heading()
{
    _Rformat(prtfptr, "HDG");
    _Rwrite(prtfptr, NULL, 0);
}

/*-----*/
/* This routine receives records from the target system until indicator */
/* 40 is turned on. Indicator 40 signals that a 'RCVENDGRP' has been */
/* received , and the target program has stopped sending records. As */
/* records are received they are printed in the print file 'BSCELPRT'. */
/*-----*/
12
read_data()
{
13
    _Rformat(icffptr, "RCVDATA");
    memset(s, '0', 99);
    _Readn(icffptr, &rcvdata_icf_i, sizeof(rcvdata_icf_i), __DFT);
    if (check_error() == ERROR)
        return(ERROR);
14
    while (s??(39??) == OFF) {
        strncpy(dtlprt_o.CUSNUM, rcvdata_icf_i.CUSNUM, 7);
        strncpy(dtlprt_o.CUSNAM, rcvdata_icf_i.CUSNAM, 25);
        strncpy(dtlprt_o.ADDR, rcvdata_icf_i.ADDR, 15);
        strncpy(dtlprt_o.CITY, rcvdata_icf_i.CITY, 15);
        strncpy(dtlprt_o.STATE, rcvdata_icf_i.STATE, 2);
        strncpy(dtlprt_o.ZIP, rcvdata_icf_i.ZIP, 5);
        strncpy(dtlprt_o.CRDLMT, rcvdata_icf_i.CRDLMT, 5);
        strncpy(dtlprt_o.CRDAMT, rcvdata_icf_i.CRDAMT, 5);
        _Rformat(prtfptr, "DTL");
        _Rwrite(prtfptr, &dtlprt_o, sizeof(dtlprt_o));
        memset(s, '0', 99);
        _Readn(icffptr, &rcvdata_icf_i, sizeof(rcvdata_icf_i), __DFT);
        if (check_error() == ERROR)
            return(ERROR);
    }
    return(NOERROR);
}

```

Figure G-21 (Part 3 of 4). ILE C/400 Source Program for the Local System

```

/*-----*/
/* Check for terminating error. If the major return code is greater */
/* than or equal to 04, then the program ends. */
/*-----*/
15
check_error()
{
    get_access_to_fb();
    if (strcmp(dsp_icf_fdbk->major_ret_code, "04", 2) == -1)
        return(NOERROR);
    else
        return(ERROR);
}

/*-----*/
/* Print error information to the print file, and close the files. */
/*-----*/
16
void end_error()
{
    get_access_to_fb();
    strncpy(error_prt_o.PGMDEV, comm_fdbk->dev_name, 10);
    strncpy(error_prt_o.FMTNM, comm_fdbk->rec_format, 8);
    strncpy(error_prt_o.MAJCOD, dsp_icf_fdbk->major_ret_code, 2);
    strncpy(error_prt_o.MINCOD, dsp_icf_fdbk->minor_ret_code, 2);
    _Rformat(prtfptr, "ERROR");
    _Rwrite(icffptr, &error_prt_o, sizeof(error_prt_o));
    close_files();
}

/*-----*/
/* Close the database, ICF, and printer files. */
/*-----*/
17
void close_files()
{
    _Rclose(dbfptr);
    _Rclose(icffptr);
    _Rclose(prtfptr);
}

/*-----*/
/* Get access to the common I/O feedabck and the display/ICF I/O */
/* feedback areas so the major/minor return code can be checked. */
/*-----*/
18
void get_access_to_fb()
{
    comm_fdbk = _Riobuf(icffptr);
    dsp_icf_fdbk = (_XXIOFB_DSP_ICF_T *)((char *)comm_fdbk +
        comm_fdbk->file_dep_fb_offset);
}

```

Figure G-21 (Part 4 of 4). ILE C/400 Source Program for the Local System

## ILE C/400 Source Program for Remote System

The following describes the objects on the remote system needed to run the ILE C/400 BSCCL target program.

## Configuration

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

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

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

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

## Program Files

The following files are used by the system:

**TGTICFF** The ICF file used to send records to and receive records from the local system. This file was

created by using the following command:

```
CRticFF FILE(BSCelIB/TGTicFF)
SRCFILE(BSCelIB/QDDSSRC)
SRCMBR(TGTicFF) TEXT('ICF FILE FOR BSCel')
```

The DDS used by this file is shown in Figure G-22 on page G-46.

```

                                Data Description Source
SEQNBR *...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8 Date
100    A*****
200    A*
300    A*          ICF FILE: TGTICFF          *
400    A*          USED IN C EXAMPLE 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 DETACH          DETACH
                                *****
                                E N D   O F   S O U R C E   * * * * *

```

Figure G-22. DDS for the TGTICFF File Used by the Remote System (ILE C/400 Language)

The command needed to define the program device entry is:

```

ADDICFDEVE FILE(BSCCELIB/TGTICFF)
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 G-23.

```

                                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 G-23. DDS for Database File Used in BSCCEL Data Transfer Programs on the Remote System (ILE C/400 Language)

**BSCCLPRT** The printer file used to format output to a printer.

The DDS for this file is shown in Figure G-24.

```

                                Data Description Source
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
                                * * * * * E N D   O F   S O U R C E   * * * * *

```

Figure G-24. DDS for the Printer File on the Remote System (ILE C/400 Language)

## Program Explanation

The following describes the ILE C/400 program on the remote system, which is shown in Figure G-25 on page G-49.

- 1** The externally described formats are included in the program.
- 2** The internal routines are prototyped so the compiler knows the type of value returned by the routine, and the type of parameters passed to the routine, if any.
- 3** The database, ICF, and printer files are opened. The files are opened for record I/O with the separate indicator area option specified for the ICF file.
- 4** The separate indicator area array `s` is initialized to 0, and the separate indicator area is established with the `_Rindara` function. The program device `TARGET` is acquired to establish an ICF session.
- 5** Procedure `print_heading()` is called to write a heading to the spooled file, and then `read_data()` is called to receive customer records from the local system until the `RCVENDGRP` response indicator is set.
- 6** Function `send_data()` is called to send customer records found in the database file, `DBFILE`, to the local system.
- 7** A detach is expected from the local program, and so another read is issued to the ICF file and the program ends.
- 8** Procedure `print_heading()` issues a write to the printer file `BSCLPRT` to print the heading in record format `HDG` to a spooled file.
- 9** Function `read_data()` reads records from the local system until the `RCVENDGRP` response indicator is set in the separate indicator area.
- 10** The first read to the ICF file is performed with record format `RCVDATA`. The separate indicator area array is reset to 0 before the read is done.
- 11** The `while` loop ends when the `RCVENDGRP` indicator is set after a read is issued to the ICF file. `RCVENDGRP` is defined as indicator 40 in `TGTICFF`, but since the first position in the separate indicator area array starts at array index 0, the index for `RCVENDGRP` is at 39.
- 12** Function `send_data()` sends customer records read from `DBFILE` until end of file is reached.
- 13** The `while` loop ends after the last record in `DBFILE` is read, or if an error occurs in the transaction, control returns to `main()`.
- 14** Function `check_error()` checks if the last input or output operation issued to the ICF file was successful or not. If the major return code is greater than or equal to 04, the program ends.
- 15** Procedure `end_error()` gets the program device and record format names from the common I/O feedback area, and gets the major/minor return codes from the display/ICF I/O feedback area. Data is written to a spooled file using record format `ERROR` defined in `BSCLPRT`, and the files are closed with a call to `close_files()`.
- 16** Procedure `close_files()` closes files `DBFILE`, `TGTICFF`, and `BSCLPRT`.
- 17** Pointers are set to the common and the display/ICF I/O feedback areas.



```

/*          * * * * * P R O L O G * * * * *
1
#pragma mapinc("icff/rcv", "bscelib/tgticff(rcvdata)", "input", "p z")
#pragma mapinc("icff/snd", "bscelib/tgticff(snddata)", "output", "p z")
#include "icff/rcv"
#include "icff/snd"
#pragma mapinc("dbf/fmt", "bscelib/dbfile(dbfmt)", "input", "p z")
#include "dbf/fmt"
#pragma mapinc("prtfdtl", "bscelib/bscelprt(dtl)", "output", "p z")
#pragma mapinc("prtfdtl", "bscelib/bscelprt(error)", "output", "p z")
#include "prtfdtl"
#include "prtfdtl"

/*-----*/
/* TARGET PROGRAM FOR BSCEL                                */
/*-----*/

#define NOERROR 0          /* No error occurred */
#define ERROR 1          /* An error occurred */
#define OFF '0'          /* Indicator off */
#define ON '1'          /* Indicator on */
#include <stdio.h>        /* Standard I/O header */
#include <recio.h>       /* Record I/O header */
#include <stdlib.h>      /* General utilities */
#include <stddef.h>      /* Standard definitions */
#include <string.h>      /* String handling utilities */
#include <xxfdbk.h>      /* Feedback area structures */

BSCELIB_TGTICFF_RCVDATA_i_t  rcvdata_icf_i;
BSCELIB_TGTICFF_SNDDATA_o_t  snddata_icf_o;
BSCELIB_DBFILE_DBFMT_i_t    dbfmt_db_i;
BSCELIB_BSCELPRT_DTL_o_t    dtl_prt_o;
BSCELIB_BSCELPRT_ERROR_o_t  error_prt_o;

_RFILE *dbfptr;          /* Pointer to the database file */
_RFILE *icffptr;        /* Pointer to the ICF file */
_RFILE *prtfdtl;        /* Pointer to the printer file */
_XXIOFB_T *comm_fdbk;   /* Pointer to common I/O feedback */
_XXIOFB_DSP_ICF_T *dsp_icf_fdbk; /* Pointer to dsp/ICF feedback */

char s??(99??);         /* Separate indic area */

2
int send_data(void);
void print_heading(void);
int read_data(void);
int check_error(void);
void end_error(void);
void close_files(void);
void get_access_to_fb(void);

```

Figure G-25 (Part 1 of 4). ILE C/400 Source Program for the Remote System

```

main()
{
  3 if ((dbfptr = _Ropen("DBFILE", "rr arrseq=y")) == NULL)
    exit(ERROR);
  if ((icffptr = _Ropen("TGTICFF", "ar+ indicators=y")) == NULL) {
    _Rclose(dbfptr);
    exit(ERROR);
  }
  if ((prtfptr = _Ropen("BSCLEPRT", "wr")) == NULL) {
    _Rclose(dbfptr);
    _Rclose(icffptr);
    exit(ERROR);
  }
  4 memset(s, '0', 99);
  _Rindara(icffptr, s); /* Set up separate indic area */
  _Racquire(icffptr, "TARGET"); /* Acquire a session */
  if (check_error() == ERROR) {
    end_error();
    exit(ERROR);
  }
  5 print_heading();
  if (read_data() == ERROR) {
    end_error();
    exit(ERROR);
  }
  6 if (send_data() == ERROR) {
    end_error();
    exit(ERROR);
  }
  7 _Rformat(icffptr, "RCVDATA"); /* Receive detach from source */
  _Rreadn(icffptr, NULL, 0, __DFT);
  close_files();
}

/*-----*/
/* Print heading to the print file. */
/*-----*/
  8 void print_heading()
{
  _Rformat(prtfptr, "HDG");
  _Rwrite(prtfptr, NULL, 0);
}

```

Figure G-25 (Part 2 of 4). ILE C/400 Source Program for the Remote System

```

/*-----*/
/* This routine receives records from the source system until indicator */
/* 40 is turned on. Indicator 40 signals that a 'RCVENDGRP' has been */
/* received, and the source program has stopped sending records. As */
/* records are received they are printed in the print file 'BSCPLPRT'. */
/*-----*/
9
read_data()
{
10
  _Rformat(icffptr, "RCVDATA");
  memset(s, '0', 99);
  _Rreadn(icffptr, &rcvdata_icf_i, sizeof(rcvdata_icf_i), __DFT);
  if (check_error() == ERROR)
    return(ERROR);

11
  while (s??(39??) == OFF) {
    strncpy(dtlprt_o.CUSNUM, rcvdata_icf_i.CUSNUM, 7);
    strncpy(dtlprt_o.CUSNAM, rcvdata_icf_i.CUSNAM, 25);
    strncpy(dtlprt_o.ADDR, rcvdata_icf_i.ADDR, 15);
    strncpy(dtlprt_o.CITY, rcvdata_icf_i.CITY, 15);
    strncpy(dtlprt_o.STATE, rcvdata_icf_i.STATE, 2);
    strncpy(dtlprt_o.ZIP, rcvdata_icf_i.ZIP, 5);
    strncpy(dtlprt_o.CRDLMT, rcvdata_icf_i.CRDLMT, 5);
    strncpy(dtlprt_o.CRDAMT, rcvdata_icf_i.CRDAMT, 5);
    _Rformat(prtfptr, "DTL");
    _Rwrite(prtfptr, &dtlprt_o, sizeof(dtlprt_o));
    memset(s, '0', 99);
    _Rreadn(icffptr, &rcvdata_icf_i, sizeof(rcvdata_icf_i), __DFT);
    if (check_error() == ERROR)
      return(ERROR);
  }
  return(NOERROR);
}

/*-----*/
/* This routine reads records from the database file 'DBFILE' until the */
/* end of file is reached. The records read in are sent to the target */
/* program. */
/*-----*/
12
send_data()
{
  _RIOFB_T *fb;

  fb = _Rreadf(dbfptr, &dbfmt_db_i, sizeof(dbfmt_db_i), __DFT);

13
  while (fb->num_bytes > 0) {
    strncpy(snddata_icf_o.CUSNUM, dbfmt_db_i.CUSNUM, 7);
    strncpy(snddata_icf_o.CUSNAM, dbfmt_db_i.CUSNAM, 25);
    strncpy(snddata_icf_o.ADDR, dbfmt_db_i.ADDR, 15);
    strncpy(snddata_icf_o.CITY, dbfmt_db_i.CITY, 15);
    strncpy(snddata_icf_o.STATE, dbfmt_db_i.STATE, 2);
    strncpy(snddata_icf_o.ZIP, dbfmt_db_i.ZIP, 5);
    strncpy(snddata_icf_o.CRDLMT, dbfmt_db_i.CRDLMT, 5);
    strncpy(snddata_icf_o.CRDAMT, dbfmt_db_i.CRDAMT, 5);
    _Rformat(icffptr, "SNDDATA");
    _Rwrite(icffptr, &snddata_icf_o, sizeof(snddata_icf_o));
    if (check_error() == ERROR)
      return(ERROR);
    fb = _Rreadn(dbfptr, &dbfmt_db_i, sizeof(dbfmt_db_i), __DFT);
  }
  return(NOERROR);
}

```

Figure G-25 (Part 3 of 4). ILE C/400 Source Program for the Remote System

```

/*-----*/
/* Check for terminating error.  If the major return code is greater */
/* than or equal to 04, then the program ends.                        */
/*-----*/
14
check_error()
{
    get_access_to_fb();
    if (strncmp(dsp_icf_fdbk->major_ret_code, "04", 2) == -1)
        return(NOERROR);
    else
        return(ERROR);
}

/*-----*/
/* Print error information to the print file, and close the files.    */
/*-----*/
15
void end_error()
{
    get_access_to_fb();
    strncpy(error_prt_o.PGMDEV, comm_fdbk->dev_name, 10);
    strncpy(error_prt_o.FMTNM, comm_fdbk->rec_format, 8);
    strncpy(error_prt_o.MAJCOD, dsp_icf_fdbk->major_ret_code, 2);
    strncpy(error_prt_o.MINCOD, dsp_icf_fdbk->minor_ret_code, 2);
    _Rformat(prtfptr, "ERROR");
    _Rwrite(prtfptr, &error_prt_o, sizeof(error_prt_o));
    close_files();
}

/*-----*/
/* Close the database, ICF, and printer files.                        */
/*-----*/
16
void close_files()
{
    _Rclose(dbfptr);
    _Rclose(icffptr);
    _Rclose(prtfptr);
}

/*-----*/
/* Get access to the common I/O feedback area and to the display/ICF */
/* I/O feedback area to check tht major/minor return code.          */
/*-----*/
17
void get_access_to_fb()
{
    comm_fdbk = _Riofbk(icffptr);
    dsp_icf_fdbk = (_XXIOFB_DSP_ICF_T *)((char *)comm_fdbk +
        comm_fdbk->file_dep_fb_offset);
}

```

Figure G-25 (Part 4 of 4). ILE C/400 Source Program for the Remote System

---

## Bibliography

This section lists publications that contain information about topics described or referred to in this guide. The books in this section are listed with their full title and order number.

### AS/400 Books

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

- *ICF Programming*, SC41-5442. Supplies the application programmer with information needed to write communications programs on the AS/400 system.
- *Communications Management*, SC41-5406. Supplies management information relating to communications, specific work management, communications error handling, and performance.
- *Communications Configuration*, SC41-5401. Contains general configuration information, including detailed descriptions of network interface, line, controller, device, mode, and class-of-service descriptions, configuration lists, and connection lists.
- *DDS Reference*, SC41-5712. Contains information about coding data description specifications for files.
- *Languages: Systems Application Architecture\* C/400\* User's Guide*, SC09-1347. Provides information regarding the ILE C/400 programming language.
- *Languages: Systems Application Architecture\* AD/Cycle\* COBOL/400\* User's Guide*, SC09-1383. Provides information regarding the ILE COBOL/400 programming language.

- *Languages: Systems Application Architecture\* AD/Cycle\* RPG/400\* User's Guide*, SC09-1348. Provides information regarding the ILE RPG/400 programming language.
- *System/36 Environment Programming*, SC41-4730. Contains information identifying the differences in the application process in the System/36 environment on the AS/400 system.
- *CL Reference*, SC41-5722. Contains the commands, command parameters and syntax for the commands used in this guide.
- *System/38 Environment Programming*, SC41-3735. Contains information identifying the differences in the application process in the System/36 environment on the AS/400 system.

### Other Related Books

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

For more information about binary synchronous communications, see the book, *General Information—Binary Synchronous Communications*, GA27-3004.

For more information about the features of the 3780 terminal, see the book, *Component Information for the IBM 3780 Data Communication Terminal*, GA27-3063.



---

# Index

## Special Characters

- \*ACQ (Acquire) command 5-9
- \*EOX (End-of-Transaction) command 5-9
- \*EXEC format example 5-4
- \*EXNX format example 5-4
- \*REL (Release) command 5-9

## Numerics

- 3740 Data Entry System 5-10
- 3780 data communication terminal 1-2
- 3780 emulation 1-2
- 6580 Displaywriter 5-11

## A

- ACQPGMDEV (program device to acquire) parameter 4-10**
- Acquire (\*ACQ) command 5-9**
- acquire operation**
  - description 4-10
- acquiring**
  - program device 4-10
  - source program 4-10
- Add Intersystem Communications Function Program Device Entry (ADDICFDEVE) command**
  - compared to configuration parameters 4-8
  - description 4-1
  - specifying parameters 4-2
- Add Prestart Job Entry (ADDPJE) command 5-8**
- ADDICFDEVE (Add Intersystem Communications Function Program Device Entry) command**
  - compared to configuration parameters 4-8
  - description 4-1
  - specifying parameters 4-2
- adding**
  - intersystem communications function (ICF) program device entry 4-1
  - prestart job entry 5-8
- ADDPJE (Add Prestart Job Entry) command 5-8**
- allow-write function 4-16**
- American National Standard Code for Information Interchange (ASCII) character set D-2**
- application**
  - flow examples F-1
  - programs supported 1-1
  - writing programs 4-1
- ASCII (American National Standard Code for Information Interchange) character set D-2**
- ASCVRYOFF (asynchronous vary off) parameter 3-1**

- asynchronous vary off (ASCVRYOFF) parameter 3-1**

## B

- bibliography H-1**
- binary synchronous communications (BSC)**
  - communications lines supported 1-2
  - configuration examples E-1
  - definition 1-1
  - handling of EOT control character 5-13
  - programming considerations 5-13
  - return code 4-19
- binary synchronous communications equivalence link (BSCSEL)**
  - See also* Communications Configuration
  - Acquire (\*ACQ) command 5-9
  - application flow examples F-1
  - command
    - \*EOX (End-of-Transaction) 5-9
    - \*REL (Release) 5-9
    - End-of-Transaction (\*EOX) 5-9
    - Release (\*REL) 5-9
  - configuration
    - deleting 2-1
    - configuring 2-1
  - control character 4-5
  - data formats supported 1-3
  - definition 1-1
  - entering commands 2-1
  - example configuration E-2
  - example of System/36 configuration E-1
  - example programs G-1
  - ICF operations A-1
  - multipoint tributary lines 1-2
  - network configuration example 1-3
  - online messages 5-11
  - point-to-point lines 1-2
  - program start request considerations 5-1
  - programming considerations 5-11
  - remote systems supported 1-1
  - return code B-1
  - running 3-1
  - specifying program device entry 4-2
  - writing application programs 4-1
- BLKLEN (block length) parameter 4-5**
- BLOCK (blocking type) parameter 4-3**
- block length (BLKLEN) parameter 4-5**
- blocking**
  - End-of-Transaction (\*EOX) command 5-9
  - write operation 4-13

**blocking type (BLOCK) parameter** 4-3  
**BSC (binary synchronous communications)**  
communications lines supported 1-2  
configuration examples E-1  
definition 1-1  
handling of EOT control character 5-13  
programming considerations 5-13  
return code 4-19

**BSCSEL (binary synchronous communications equivalence link)**

*See also* Communications Configuration

Acquire (\*ACQ) command 5-9  
application flow examples F-1  
command  
\*EOX (End-of-Transaction) 5-9  
\*REL (Release) 5-9  
End-of-Transaction (\*EOX) 5-9  
Release (\*REL) 5-9  
configuration  
deleting 2-1  
configuring 2-1  
control character 4-5  
data formats supported 1-3  
definition 1-1  
entering commands 2-1  
example configuration E-2  
example of System/36 configuration E-1  
example programs G-1  
ICF operations A-1  
multipoint tributary lines 1-2  
network configuration example 1-3  
online messages 5-11  
point-to-point lines 1-2  
program start request considerations 5-1  
programming considerations 5-11  
remote systems supported 1-1  
return code B-1  
running 3-1  
specifying program device entry 4-2  
writing application programs 4-1

**BSCSELICF file** G-1, G-29

**BSCSELPRT file** G-3, G-23

## C

**CALL (Call Program) command** 5-2

**Call Program (CALL) command** 5-2

**cancel-invite function** 4-16

**canceling**

cancel-invite function 4-16

**CFGOBJ (configuration object) parameter** 3-1

**CFGTYPE (configuration type) parameter** 3-1

**Change Display File (CHGDSPF) command** 4-15

**Change Intersystem Communications Function Device Entry (CHGICFDEVE) command** 4-1

**Change Intersystem Communications Function File (CHGICFF) command** 4-1, 4-15

**Change Prestart Job Entry (CHGPJE) command** 5-8

**changing**

BSCSEL configuration description 2-1  
controller description 2-1  
device description 2-1  
display file 4-15  
ICF (intersystem communications function) program device entry 4-1  
intersystem communications function file 4-1, 4-15  
line description 2-1  
prestart job entry 5-8

**character set**

ASCII (American National Standard Code for Information Interchange) character set D-2  
EBCDIC (extended binary-coded decimal interchange code) D-1

**CHGDSPF (Change Display File) command** 4-15

**CHGICFDEVE (Change Intersystem Communications Function Device Entry) command** 4-1

**CHGICFF (Change Intersystem Communications Function File) command** 4-1, 4-15

**CHGPJE (Change Prestart Job Entry) command** 5-8

**CMNTYPE (communications type) parameter** 4-3

**COBOL/400 program example** G-1, G-12

**coding positions** 5-2

**combining**

write operation with other operations 4-12

**command, binary synchronous communications equivalence link (BSCSEL)**

\*ACQ (Acquire) 5-9  
\*EOX (End-of-Transaction) 5-9  
\*REL (Release) 5-9  
Acquire (\*ACQ) 5-9  
End-of-Transaction (\*EOX) 5-9  
Release (\*REL) 5-9

**command, CL**

Add Intersystem Communications Function Device Entry (ADDICFDEVE) 4-1  
Add Prestart Job Entry (ADDPJE) 5-8  
ADDICFDEVE (Add Intersystem Communications Function Device Entry) 4-1  
ADDPJE (Add Prestart Job Entry) 5-8  
CALL (Call Program) 5-2  
Call Program (CALL) 5-2  
Change Display File (CHGDSPF) 4-15  
Change Intersystem Communications Function Device Entry (CHGICFDEVE) 4-1  
Change Intersystem Communications Function File (CHGICFF) 4-1, 4-15  
Change Prestart Job Entry (CHGPJE) 5-8  
CHGDSPF (Change Display File) 4-15  
CHGICFDEVE (Change Intersystem Communications Function Device Entry) 4-1



## **command, CL** *(continued)*

CHGICFF (Change Intersystem Communications Function File) 4-1, 4-15  
CHGPJE (Change Prestart Job Entry) 5-8  
Create Controller Description (CRTCTLBSC) 2-1  
Create Device Description (CRTDEVBSC) 2-1  
Create Display File (CRTDSPF) 4-15  
Create Intersystem Communications Function File (CRTICFF) 4-1, 4-15  
Create Line Description (CRTLINBSC) 2-1  
CRTCTLBSC (Create Controller Description) 2-1  
CRTDEVBSC (Create Device Description) 2-1  
CRTDSPF (Create Display File) 4-15  
CRTICFF (Create Intersystem Communications Function File) 4-1, 4-15  
CRTLINBSC (Create Line Description) 2-1  
Delete File (DLTF) 4-1  
Display File Description (DSPFD) 4-1  
Display File Field Description (DSPFFD) 4-1  
DLTF (Delete File) 4-1  
DSPFD (Display File Description) 4-1  
DSPFFD (Display File Field Description) 4-1  
HLDCMNDEV (Hold Communications Device) 5-7  
Hold Communications Device (HLDCMNDEV) 5-7  
Override with Display File (OVRDSPF) 4-15  
Override with Intersystem Communications Function File (OVRICFF) 4-1, 4-15  
Override with Intersystem Communications Function Program Device Entry (OVRICFDEVE) 4-1  
OVRDSPF (Override with Display File) 4-15  
OVRICFDEVE (Override with Intersystem Communications Function Program Device Entry) 4-1  
OVRICFF (Override with Intersystem Communications Function File) 4-1, 4-15  
Remove Intersystem Communications Function Program Device Entry (RMVICFDEVE) 4-2  
Rename Object (RNMOBJ) 2-1  
Retrieve Configuration Source (RTVCFGSRC) 2-1  
RMVICFDEVE (Remove Intersystem Communications Function Program Device Entry) 4-2  
RNMOBJ (Rename Object) 2-1  
RTVCFGSRC (Retrieve Configuration Source) 2-1  
Vary Configuration (VRYCFG) 3-1  
VRYCFG (Vary Configuration) 3-1  
Work Configuration Status (WRKCFGSTS) 2-1  
WRKCFGSTS (Work Configuration Status) 2-1

## **communications**

descriptions 2-1  
ending a session 4-17  
ending a transaction 4-17  
I/O feedback area 4-19  
notifying program of errors 4-15  
receiving data 4-14  
return code processing 4-19  
sending data 4-12  
starting a session 4-9

## **communications** *(continued)*

starting a transaction 4-10  
using response indicators 4-18

## **communications device**

holding 5-7

## **communications line**

contention state 4-16  
supported by BSCCEL 1-2  
using point-to-point with acquire operation 4-10

## **communications type (CMNTYPE) parameter 4-3**

## **compress and decompress data (DTACPR) parameter 4-6**

## **conducting**

consecutive sessions 1-2

## **configuration**

BSCCEL (binary synchronous communications equivalence link) 2-1

*See also* Communications Configuration

changing 2-1

deleting 2-1

example for BSCCEL E-1, E-2

retrieving status 2-1

varying 3-1

## **configuration description**

creating 2-1

## **configuration object (CFGOBJ) parameter 3-1**

## **configuration source**

retrieving 2-1

## **configuration status**

retrieving 2-1

working with 2-1

## **configuration type (CFGTYPE) parameter 3-1**

## **configuring**

AS/400 for BSCCEL example E-2

System/36 for BSCCEL example E-1

## **contention state**

at end-of-transmission 4-16

definition 4-13

for lines 4-16

## **continuation format**

description 5-4

entering 5-4

example 5-4

using with 3740 Data Entry Systems 5-10

## **control character 5-9, C-1**

## **controller description (BSC)**

changing 2-1

creating 2-1

deleting 2-1

retrieving status 2-1

## **Create Controller Description (CRTCTLBSC)**

### **command 2-1**

## **Create Device Description (CRTDEVBSC)**

### **command 2-1**

## **Create Display File (CRTDSPF) command 4-15**

**Create Intersystem Communications Function File (CRTICFF) command** 4-1, 4-15

**Create Line Description (CRTLINBSC) command** 2-1

**creating**

- configuration description 2-1
- controller description (BSC) 2-1
- device description (BSC) 2-1
- display file 4-15
- intersystem communications function file 4-1, 4-15
- line description (BSC) 2-1

**CRTCTLBSC (Create Controller Description) command** 2-1

**CRTDEVBSC (Create Device Description) command** 2-1

**CRTDSPF (Create Display File) command** 4-15

**CRTICFF (Create Intersystem Communications Function File) command** 4-1, 4-15

**CRTLINBSC (Create Line Description) command** 2-1

**current status** 4-17

## D

**data**

- receiving 4-14
- sending 4-12

**data description specifications (DDS) keyword** A-3  
*See also* ICF Programming

**data file** 5-10

**data format**

- for target program 5-6
- parameters 5-6
- supported by BSC 1-3

**data-link-escape (DLE) character**  
definition 4-4

**DBFILE file** G-13, G-22

**DDS (data description specifications) keyword** A-3  
*See also* ICF Programming

**Delete File (DLTF) command** 4-1

**deleting**

- BSC configuration description 2-1
- controller description 2-1
- device description 2-1
- file 4-1
- line description 2-1

**detach function** 4-17

**device description (BSC)**

- changing 2-1
- creating 2-1
- deleting 2-1
- retrieving status 2-1

**display file**

- changing 4-15
- creating 4-15
- overriding with 4-15

**Display File Description (DSPFD) command** 4-1

**Display File Field Description (DSPFFD)**

**command** 4-1

**displaying**

- file description 4-1
- file field description 4-1

**distributed data processing support** 1-1

**DLE (data-link-escape) character**

- definition 4-4

**DLTF (Delete File) command** 4-1

**DSPFD (Display File Description) command** 4-1

**DSPFFD (Display File Field Description)**

**command** 4-1

**DTACPR (compress and decompress data) parameter** 4-6

## E

**EBCDIC (extended binary-coded decimal interchange code) character set** D-1

**end-of-group function** 4-13

**end-of-session function** 4-18

**end-of-text (ETX) character**

- definition 4-3

**End-of-Transaction (\*EOX) command** 5-9

**end-of-transmission (EOT) character**

- binary synchronous handling 5-13
- definition 4-6

**end-of-transmission-block (ETB) character**

- definition 4-3

**ending**

- office system session 5-11
- session 4-17
- transaction 4-17

**entering**

- continuation format 5-4
- user-defined parameter 5-2

**EOT (end-of-transmission) character**

- binary synchronous handling 5-13
- definition 4-6

**error recovery** 4-20

**ETB (end-of-transmission-block) character**

- definition 4-3

**ETX (end-of-text) character**

- definition 4-3

**evoke function**

- See also* DDS Reference
- See also* ICF Programming
- definition 4-11
- parameters specified 4-11
- sending program start request with 4-11, 5-4
- specifying security function 4-12
- specifying target program 4-11, 5-6
- starting a transaction 4-11
- using more than one 4-11
- using with requesting device 4-11

**evoke function** (*continued*)  
 with RMTBSCEL (remote BSCEL) 4-11

**example**  
 configuring the AS/400 for BSCEL E-2  
 configuring the System/36 for BSCEL E-1  
 network configuration 1-3

**example program**  
 ILE C/400 program  
   local system G-36  
   remote system G-44  
 ILE COBOL/400 program  
   local system G-1  
   remote system G-12  
 ILE RPG/400 program  
   local system G-21  
   remote system G-29

**extended binary-coded decimal interchange code (EBCDIC) character set** D-1

## F

**fail function** 4-15

**failed program start request** 5-6

**file**  
 BSCELICF G-1, G-29  
 BSCELPRT G-3, G-23  
 DBFILE G-13, G-22  
 deleting 4-1

**file attribute**  
 managing 4-1

**file description**  
 displaying 4-1

**file field description**  
 displaying 4-1

**file format**  
 receiving multiple 5-10  
 sending multiple 5-10

**FILE parameter** 4-2

**FMTSLT (format select) parameter**  
 valid value 4-2  
 with acquire operation 4-10  
 with record-identification function 4-14

**format**  
 program start request 5-3

**format select (FMTSLT) parameter**  
 valid value 4-2  
 with acquire operation 4-10  
 with record-identification function 4-14

**function**  
 allow-write 4-16  
 cancel-invite 4-16  
 DDS keyword A-3  
 detach 4-17  
 end-of-session 4-18  
 evoke 4-11, 5-4  
 fail 4-15

**function** (*continued*)  
 invite 4-14  
 record-identification 4-14  
 request-to-write 4-15  
 security 4-12  
 subdevice selection 4-13  
 timer 4-16  
 variable-length-data 4-8

## G

**get-attribute**  
 See ICF Programming

**get-attributes operation** 4-17

**group separator type (GRPSEP) parameter** 4-6

**GRPSEP (group separator type) parameter** 4-6

## H

**handling**  
 office system documents 1-2

**high-level language (HLL)**  
 operations A-1  
 supported 1-1

**HLDCMNDEV (Hold Communications Device) command** 5-7

**HLL (high-level language)**  
 operations A-1  
 supported 1-1

**Hold Communications Device (HLDCMNDEV) command** 5-7

**holding**  
 communications device 5-7

## I

**I/O feedback area**  
 See *also* ICF Programming  
 general description 4-19

**ICF (intersystem communications function)**  
 definition 4-1

**ICF (intersystem communications function) file**  
 changing 4-1, 4-15  
 creating 4-1, 4-15  
 definition 4-1  
 overriding with 4-1, 4-15  
 parameter 4-2  
 using 4-1

**ICF (intersystem communications function) program device entry**  
 adding 4-1  
 changing 4-1  
 overriding with 4-1  
 removing 4-2

**idle disconnect timer parameter** 4-16

**ILE C/400 source program (local system)**  
 configuration G-36  
 program file  
   BSCelpRT G-39  
   DBFILE G-37  
   SCRICFF G-36

**ILE C/400 source program (remote system)**  
 configuration G-44  
 program file  
   BSCelpRT G-47  
   DBFILE G-46  
   TGTICFF G-45

**ILE COBOL/400 source program (local system)**  
 configuration G-1  
 program file  
   BSCelICF G-1  
   BSCelpRT G-3  
   DBFILE G-3

**ILE COBOL/400 source program (remote system)**  
 configuration G-12  
 program file  
   BSCelICF G-12  
   BSCelpRT G-14  
   DBFILE G-13

**ILE RPG/400 source program (local system)**  
 configuration G-21  
 program file  
   BSCelICF G-21  
   BSCelpRT G-23  
   DBFILE G-22

**ILE RPG/400 source program (remote system)**  
 configuration G-29  
 program file  
   BSCelICF G-29  
   BSCelpRT G-31  
   DBFILE G-30

**INACTTMR (inactivity timer) parameter 4-16**  
**indicating**  
 end of documents 5-11

**initial connection (INLCNN) parameter 4-7**  
**INLCNN (initial connection) parameter 4-7**

**intermediate-text-block (ITB) character**  
 definition 4-3

**interrecord-separator (IRS) character**  
 definition 4-3

**intersystem communications function (ICF)**  
 definition 4-1

**intersystem communications function (ICF) file**  
 changing 4-1, 4-15  
 creating 4-1, 4-15  
 definition 4-1  
 overriding with 4-1, 4-15  
 parameter 4-2  
 using 4-1

**intersystem communications function (ICF) program device entry**

**intersystem communications function (ICF) program device entry (continued)**  
 adding 4-1  
 changing 4-1  
 overriding with 4-1  
 removing 4-2

**invite function**  
 canceling with cancel-invite function 4-16  
 description 4-14  
 with read-from-invited-program-devices function 4-14

**IRS (interrecord-separator) character**  
 definition 4-3

**ITB (intermediate-text-block) character**  
 definition 4-3

**J**  
**journal (QHST) 5-12**

**K**  
**keyword**  
 DDS (data description specifications) A-3  
 variable length (VARLEN) 4-8  
 VARLEN (variable length) 4-8

**L**  
**language operation**  
 for programming languages A-2  
 supported by ICF A-1

**line**  
 number of sessions 1-2

**line description (BSC)**  
 changing 2-1  
 creating 2-1  
 deleting 2-1  
 retrieving status 2-1  
 specifying idle disconnect timer parameter 4-16

**line protocol C-1**

**local system**  
 COBOL/400 program G-1  
 ILE C/400 program G-36  
 RPG/400 program G-21

**M**  
**managing**  
 file attribute 4-1

**MAXBUFFER (maximum buffer size) parameter 4-9**  
**maximum buffer size (MAXBUFFER) parameter 4-9**  
**maximum record wait time (WAITRCD) parameter 4-12**

**message**  
 BSCel online 5-11

multiple file formats 5-10  
multipoint line 4-10  
multipoint tributary line 1-2

## N

### notifying

program of errors 4-15

### null record

control characters 5-9  
for multiple file formats 5-10  
receiving for remote system 5-9  
receiving for response indicator 4-18

**number of transactions active at one time 4-11**

## O

### object

renaming 2-1

### office system

ending session 5-11  
format 5-10  
handling documents 1-2  
indicating end of documents 5-11  
programming considerations 5-10  
receiving documents 5-11

**online message 5-12**

**open operation 4-10**

*See also* ICF Programming

### operation

acquire 4-10  
DDS keyword A-3  
get-attributes 4-17  
open 4-10  
read 4-14  
read-from-invited-program-devices 4-12, 4-14  
release 4-17  
using A-1  
write 4-12

**Override with Display File (OVRDSPF)**

**command 4-15**

**Override with Intersystem Communications Function File (OVRICFF) command 4-1, 4-15**

**Override with Intersystem Communications Function Program Device Entry (OVRICFDEVE) command 4-1**

### overriding with

display file 4-15  
intersystem communications function file 4-1, 4-15  
intersystem communications function program device entry 4-1

**OVRDSPF (Override with Display File)**

**command 4-15**

**OVRICFDEVE (Override with Intersystem Communications Function Device Entry) command 4-1**

**OVRICFDEVE (Override with Intersystem Communications Function Program Device Entry) command 4-1**

**OVRICFF (Override with Intersystem Communications Function File) command 4-1, 4-15**

## P

### parameter

ACQPGMDEV (program device to acquire) 4-10  
Add Intersystem Communications Function Device Entry (ADDICFDEVE) command 4-2

ADDICFDEVE (Add Intersystem Communications Function Device Entry) command 4-2

ASCVRYOFF (asynchronous vary off) 3-1

asynchronous vary off (ASCVRYOFF) 3-1

BLKLEN (block length) 4-5

BLOCK (blocking type) 4-3

block length (BLKLEN) 4-5

blocking type (BLOCK) 4-3

CFGOBJ (configuration object) 3-1

CFGTYPE (configuration type) 3-1

Change Intersystem Communications Function Device Entry (CHGICFDEVE) command 4-2

CHGICFDEVE (Change Intersystem Communications Function Device Entry) command 4-2

CMNTYPE (communications type) 4-3

communications type (CMNTYPE) 4-3

compress and decompress data (DTACPR) 4-6

configuration object (CFGOBJ) 3-1

configuration type (CFGTYPE) 3-1

data format 5-6

DTACPR (compress and decompress data) 4-6

FILE 4-2

FMTSLT (format select)

valid value 4-2

with acquire operation 4-10

with record-identification function 4-14

for evoke function 4-11

format select (FMSLT)

valid value 4-2

with acquire operation 4-10

with record-identification function 4-14

group separator type (GRPSEP) 4-6

GRPSEP (group separator type) 4-6

INACTTMR (activity timer) 4-16

initial connection (INLCNN) 4-7

INLCNN (initial connection) 4-7

MAXBUFFER (maximum buffer size) 4-9

maximum buffer size (MAXBUFFER) 4-9

maximum record wait time (WAITRCD) 4-12

Override with Intersystem Communications Function Device Entry (OVRICFDEVE) command 4-2

OVRICFDEVE (Override with Intersystem Communications Function Device Entry) command 4-2

PGMDEV (program device) 4-2

**parameter** (continued)

- positional 5-2
- program device (PGMDEV) 4-2
- program device to acquire (ACQPGMDEV) 4-10
- RANGE 3-1
- RCDLEN (record length) 4-5
- record length (RCDLEN) 4-5
- remote BSCCEL (RMTBSCCEL) 4-7
- remote location name (RMTLOCNAME) 4-2
- RESET 3-1
- RMTBSCCEL (remote BSCCEL) 4-7
- RMTLOCNAME (remote location name) 4-2
- SECURE 4-7
- security 4-12
- STATUS 3-1
- transmit in transparent mode (TRNSPY) 4-6
- TRNSPY (transmit in transparent mode) 4-6
- TRUNC (truncate trailing blanks) 4-6
- truncate trailing blanks (TRUNC) 4-6
- user-defined 5-2
- Vary Configuration (VRYCFG) command 3-1
- vary on wait (VRYWAIT) 3-1
- VRYCFG (Vary Configuration) command 3-1
- VRYWAIT (vary on wait) 3-1
- WAITRCD (maximum record wait time) 4-12
- with record-identification function 4-14

**password parameter** 4-12

**PGMDEV (program device) parameter** 4-2

**point-to-point line** 1-2, 4-10

**positional parameter** 5-2

**prestart job**

- See also ICF Programming
- definition 5-8

**prestart job entry**

- adding 5-8
- changing 5-8

**profile ID parameter** 4-12

**program**

- writing 4-1

**program device**

- acquiring 4-10

**program device (PGMDEV) parameter** 4-2

**program device entry**

- See also ICF Programming
- command 4-2
- parameter 4-2

**program device to acquire (ACQPGMDEV) parameter** 4-10

**program start request**

- \*EXEC format example 5-4
- \*EXNX format example 5-4
- coding positions 5-2
- continuation format 5-4
- continuations
  - using with 3740 Data Entry Systems 5-10
- failed 5-6

**program start request** (continued)

- files for 3740 Data Entry Systems 5-10
- format
  - using \*EXNC or \*EXNX 5-9
- from remote system 5-5
- from the AS/400 system 5-6
- maximum length 5-2, 5-6
- positional parameters 5-2
- prestart job 5-8
- separating parameters 5-2
- session activity 5-5
- specifying to send online messages 5-11
- syntax diagram 5-1
- user-defined parameters 5-2

**programming considerations**

- for remote systems 5-8
- office system 5-10
- receiving null records 5-9
- reverse-interrupt indicator 5-12
- sending BSCCEL commands 5-9

**programming example**

- COBOL/400 Source Program for the Local System G-7
- COBOL/400 Source Program for the Remote System G-16
- ILE C/400 Source Program for the Local System G-41
- ILE C/400 Source Program for the Remote System G-49
- RPG/400 Source Program for the Local System G-24
- RPG/400 Source Program for the Remote System G-32

## Q

**QHST journal** 5-12

**QICDMF file** 4-9

## R

**RANGE parameter** 3-1

**RCDLEN (record length) parameter** 4-5

**read operation**

- description 4-14
- using response indicators 4-18

**read-from-invited-program-devices operation**

- definition 4-12
- using 4-14

**reason codes** 5-6

**receive-detach response indicator** 4-19

**receive-end-of-group response indicator** 4-18

**receive-turnaround response indicator** 4-19

**receiving**

- data
  - during a transaction 4-14
  - from previously invited program device 4-14

**receiving** (*continued*)  
 data (*continued*)  
   from specific program 4-14  
 documents 5-11  
 multiple files 5-10  
 null records 5-9  
 response indicator 4-18  
 return code 0310 4-16

**record blocking** 4-13

**record length**  
 determining 4-8

**record length (RCDLEN) parameter** 4-5

**record separator character** 4-5

**record-identification function** 4-14

**Release (\*REL) command** 5-9

**release operation** 4-17

**remote BSCSEL (RMTBSCSEL) parameter** 4-7

**remote location name (RMTLOCNAME) parameter** 4-2

**remote system**  
 acquiring a program 4-10  
 BSCSEL command 5-9  
 COBOL/400 program G-12  
 ending a transaction 4-17  
 ILE C/400 program G-44  
 receiving multiple file formats 5-10  
 receiving null records 5-9  
 RPG/400 program G-29  
 sending data 4-12  
 sending multiple file formats 5-10  
 sending security information 4-12  
 starting a transaction 4-10  
 supported by BSCSEL 1-1  
 types 5-8  
 using BSCSEL 4-10  
 using office systems 5-10  
 using security 4-10

**Remove Intersystem Communications Function Program Device Entry (RMVICFDEVE) command** 4-2

**removing**  
 intersystem communications function (ICF) program device entry 4-2

**Rename Object (RNMOBJ) command** 2-1

**renaming**  
 object 2-1

**request-to-write function** 4-15

**requesting device**  
 using acquire operation 4-10  
 using with evoke function 4-11

**RESET parameter** 3-1

**response indicator**  
 description 4-18  
 not valid with write operation 4-18  
 receive-detach 4-19  
 receive-end-of-group 4-18

**response indicator** (*continued*)  
 receive-turnaround 4-19  
 using 4-18  
 using more than one on a read operation 4-18  
 valid with read operation 4-18

**Retrieve Configuration Source (RTVCFGSRC) command** 2-1

**retrieving**  
 configuration source 2-1  
 configuration status 2-1

**return code**  
 0310 (timer ends) 4-16  
 cancel-invite function 4-16  
 detailed description B-1  
 operation completed successfully 4-16  
 receive-detach response indicator 4-19  
 receive-end-of-group response indicator 4-19  
 receive-turnaround response indicator 4-19  
 request-to-write function 4-16  
 timer function 4-16  
 using 4-19

**reverse-interrupt indicator (RVI) character**  
 definition 4-15  
 programming considerations 5-12  
 sending with request-to-write function 4-15

**RMTBSCSEL (remote BSCSEL) parameter** 4-7

**RMTLOCNAME (remote location name) parameter** 4-2

**RMVICFDEVE (Remove Intersystem Communications Function Program Device Entry) command** 4-2

**RNMOBJ (Rename Object) command** 2-1

**RPG/400 program example** G-21, G-29

**RTVCFGSRC (Retrieve Configuration Source) command** 2-1

**RVI (reverse-interrupt indicator) character**  
 definition 4-15  
 programming considerations 5-12  
 sending with request-to-write function 4-15

## **S**

**safe indicator**  
 definition 4-19

**SECURE parameter** 4-7

**security**  
 starting a transaction 4-10  
 using 4-10

**security function** 4-12

**sending**  
 BSCSEL commands 5-9  
 data 4-12  
 multiple files 5-10  
 program start request 4-11  
 security information 4-12  
 separate record 4-13

## **sending** *(continued)*

with request-to-write function 4-15

## **session**

activity from a program start request 5-5

conducting consecutively 1-2

definition 4-9

ending 4-17

for multipoint tributary lines 1-2

for point-to-point lines 1-2

getting current status 4-17

number supported per line 1-2

starting 4-9

## **setting up BSCSEL (binary synchronous communications equivalence link) 2-1**

### **source program**

definition 4-2

notifying of errors 4-15

starting a session 4-10

using release operation 4-17

### **specifying**

idle disconnect timer parameter 4-16

parameters with evoke function 4-11

record blocking 4-13

security function 4-12

target program 4-11, 5-6

user-defined parameter 4-11

### **starting**

other programs 5-6

session 4-9

session from target program 4-10

target program 5-6

transaction 4-10, 4-11

### **status**

retrieving 2-1

## **status message from device 5-12**

### **STATUS parameter 3-1**

### **subdevice selection function**

description 4-13

sending as a separate record 4-13

### **switched line**

with allow-write function 4-16

with cancel-invite function 4-16

## **synchronization level keyword 4-11**

### **syntax diagram 5-1**

### **system-supplied format**

*See also* ICF Programming

description A-3

### **System/36**

CNFIGICF procedure for BSCSEL configuration E-1

configuring for BSCSEL example E-1

## **T**

### **target program**

data format used 5-6

definition 4-2

## **target program** *(continued)*

maximum length of name 5-6

specifying with evoke function 4-11, 5-6

started by evoke parameters 5-6

starting a session from 4-10

starting other programs 5-6

using user-defined parameters 5-2

## **telecommunications 1-1**

### **timer function**

description 4-16

receiving 0310 return code (timer ends) 4-16

### **transaction**

definition 4-10

ending 4-17

number active at one time 1-2, 4-11

sending data 4-12

starting 4-10

### **transmission control character**

definition 1-1

### **transmit in transparent mode (TRNSPY)**

**parameter 4-6**

### **transparent text mode**

definition 1-3, 4-6

### **TRNSPY (transmit in transparent mode)**

**parameter 4-6**

### **TRUNC (truncate trailing blanks) parameter 4-6**

### **truncate trailing blanks TRUNC) parameter 4-6**

## **U**

### **user identifier parameter 4-12**

### **user-defined parameter**

entering 5-2

specifying with evoke function 4-11

### **using**

response indicator 4-18

security 4-10

## **V**

### **variable length (VARLEN) keyword 4-8**

### **variable-length-data function 4-8**

### **VARLEN (variable length) keyword 4-8**

### **Vary Configuration (VRYCFG) command 3-1**

*See also* Communications Management

### **vary on wait (VRYWAIT) parameter 3-1**

### **varying**

configuration 3-1

### **VRYCFG (Vary Configuration) command 3-1**

*See also* Communications Management

### **VRYWAIT (vary on wait) parameter 3-1**

## **W**

### **WAITRCD (maximum record wait time)**

**parameter 4-12**



**Work with Configuration Status (WRKCFGSTS)**

**command** 2-1

**working with**

configuration status 2-1

**write operation**

blocking records 4-13

combining with other operations 4-12

description 4-12

using response indicators 4-18

**writing application programs** 4-1

**WRKCFGSTS (Work with Configuration Status)**

**command** 2-1



---

# Reader Comments—We'd Like to Hear from You!

AS/400 Advanced Series  
BSC Equivalence Link  
Programming  
Version 4

Publication No. SC41-5445-00

Overall, how would you rate this manual?

	Very Satisfied	Satisfied	Dissatisfied	Very Dissatisfied
Overall satisfaction				

How satisfied are you that the information in this manual is:

Accurate				
Complete				
Easy to find				
Easy to understand				
Well organized				
Applicable to your tasks				
T H A N K   Y O U !				

Please tell us how we can improve this manual:

---

---

---

---

---

May we contact you to discuss your responses?  Yes  No  
Phone: (\_\_\_\_) \_\_\_\_\_ Fax: (\_\_\_\_) \_\_\_\_\_ Internet: \_\_\_\_\_

**To return this form:**

- Mail it
- Fax it  
United States and Canada: **800+937-3430**  
Other countries: **(+1)+507+253-5192**
- Hand it to your IBM representative.

Note that IBM may use or distribute the responses to this form without obligation.

\_\_\_\_\_  
Name

\_\_\_\_\_  
Address

\_\_\_\_\_  
Company or Organization

\_\_\_\_\_  
Phone No.



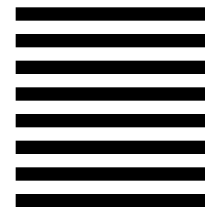
Fold and Tape

Please do not staple

Fold and Tape



NO POSTAGE  
NECESSARY  
IF MAILED IN THE  
UNITED STATES



# BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

ATTN DEPT 542 IDCLERK  
IBM CORPORATION  
3605 HWY 52 N  
ROCHESTER MN 55901-9986



Fold and Tape

Please do not staple

Fold and Tape





Printed in the United States of America  
on recycled paper containing 10%  
recovered post-consumer fiber.

SC41-5445-00



*Spine information:*



AS/400 Advanced Series

BSC Equivalence Link  
Programming

*Version 4*