Connect:Express®
OS/390

Options Guide

Version 4.2.0

Sterling Commerce
An IBM Company

Connect:Express OS/390 Options Guide
Version 4.2.0
First Edition

Printed in the United States of America.

CX390OPTG703

Contents

Preface

Chapter 1 CICS Interface

Chapter 2 IMS Interface

Chapter 3 RJE Interface

Index

Preface

The *Connect:Express OS/390 Options Guide* is for programmers and network operations staff who install and maintain the Connect:Express product.

This guide provides information about three options for the Connect:Express OS/390 product: the CICS interface, the IMS interface, and the RJE option. These optional features enable you to better integrate file transfer operations in your environment by providing tools to integrate file transfer requests and controls in applications. The asset protection file controls these features.

This guide assumes knowledge of the OS/390 operating system, including its applications, network, and environment. If you are not familiar with the OS/390 operating system, refer to IBM documentation for more information.

Chapter Overview

The *Connect:Express OS/390 Options Guide* is organized into the following chapters:

Chapter/Appendix	Description
Chapter 1 CICS Interface	This chapter describes the CICS interface option. This option provides the environment in which a CICS application can communicate with Connect:Express. Facilities enable a CICS application to request Connect:Express to perform certain functions on behalf of the application.
Chapter 2 IMS Interface	This chapter describes the IMS interface option which provides subroutine modules that can be called by an IMS application to communicate with Connect:Express. These subroutines enable an IMS application to request Connect:Express to perform certain functions on behalf of the application.
Chapter 3 RJE Interface	This chapter describes the RJE interface option. An RJE partner is a computer connected to the central site via BSC lines such as 2780/3780 or SDLC lines such as 3370. The Connect:Express RJE option enables this partner to send and receive files under the control of Connect:Express.

Product Documentation

Connect:Express documentation consists of the following manuals:

❖ The *Connect:Express OS/390 4.2.0 Release Notes* lists maintenance updates and any important notes.

❖ The *Connect:Express OS/390 4.2.0 Installation Guide* describes the planning and installation of Connect:Express.

- The *Connect:Express OS/390 4.2.0 User Guide* includes general information about using the TSO/ISPF interface, and serves as a reference of user and environment commands.
- The *Connect:Express OS/390 4.2.0 Utilities Guide* describes the optional Utilities package that you can integrate with Connect:Express.
- The *Connect:Express OS/390 4.2.0 FTP Guide* provides you with the information that you need to use Connect:Express with the FTP protocol.
- The *Connect:Express OS/390 4.2.0 Administration Guide* provides detailed information about transfer operations for system administrators and other advanced users of Connect:Express.
- The *Connect:Express OS/390 4.2.0 Options Guide* provides information about the CICS, IMS, and RJE interfaces available for Connect:Express.
- The *Connect:Express OS/390 4.2.0 PeSIT User Fields Guide* describes how you can exchange the PeSIT Pi37 and Pi99 fields with any PeSIT software.
- The *Connect:Express OS/390 4.2.0 Etebac3 User Guide* provides you with the information that you need to use Connect:Express with the Etebac3 protocol.
- The *Connect:Express HTTP Option Implementation Guide* provides you with the information that you need to implement HTTP access to Connect:Express OS/390 repository.
- The *Connect:Express OS/390 4.2.0 SSL Guide* includes general information on implementing secured file transfers.
- The *Connect:Express OS/390 4.2.0 Sysplex Supervision Guide* includes general information on implementing a group of Connect:Express Plex managers under control of a Connect:Express Plex supervisor.

Getting Support for Sterling Commerce Products

Sterling Commerce provides intuitive technical products and superior Help and documentation to enable you to work independently. However, if you have a technical question regarding a Sterling Commerce product, use the Sterling Commerce Customer Support Web site.

The Sterling Commerce Customer Support Web site at *www.sterlingcommerce.com* is the doorway to Web support, information, and tools. This Web site contains several informative links, including a solutions database, an issue tracking system, fix information, documentation, workshop information, contact information, sunset and retirement schedules, and ordering information. Refer to the Customer Support Reference Guide at *www.sterlingcommerce.com/customer/tech_support.html* for specific information on getting support for Sterling Commerce products.

Conventions Used in This Guide

The *Connect:Express OS/390 Options Guide* uses certain notational conventions. This section describes the conventions used in this guide.

Convention	Description
UPPERCASE LETTERS	Uppercase letters in the command format indicate that you type in information as shown.

Convention	Description
Lowercase letters	Lowercase letters or words in commands or syntax boxes require substitution by the user. For example, index1.index2.PARMLIB indicates that you must provide the first and second indexes of the string. "PARMLIB" is mandatory.
Bold Letters	Bold print in syntax boxes indicates Connect:Express commands and required parameters. For example, PLEX=N indicates that the parameter PLEX must be set to N.
Underlined Letters	Underlining indicates default values for parameters and subparameters. For example, PLEX=Y|N specifies that the default for PLEX is N.
Vertical Bars (|)	Vertical bars indicate that you can supply one of a series of values separated by the vertical bars. For example, RUN=H|C specifies that H or C is valid.
`Monospaced characters` (characters of equal width)	`Monospaced characters` represent information for screens, commands, Processes, and reports.
Punctuation	Code all commas and parentheses as they appear.
£ or #	The Pound character (£) and the hash character (#) are equivalent.

Implementing an Interface Option

Connect:Express options are interfaces that you can integrate into user applications. All optional feature objects are provided on the standard product tape. To enable an option, it must be included in your asset protection key.

Communication structures between applications and Connect:Express are described in the *MACLIB* product library. You should always use assembly Dsects or Cobol Copies in user programs.

Note: The *ISPCLIB* contains an ISPF EDIT MACRO that automatically converts an assembly-dsect to a cobol-copy (see M2COBASM).

Updating the Product

Updating the product should not impact user application programs if you use product structures (Dsects and Copies). The user program should clear structures before the initial communication with the monitor. If a compilation or link-edit is necessary, the release notes will inform you. This may be the case if a new function has been added.

Chapter 1

CICS Interface

This chapter describes the CICS interface option. This option provides the environment in which a CICS application can communicate with Connect:Express OS/390. Facilities enable a CICS application to request Connect:Express to perform certain functions on behalf of the application.

Overview

The Connect:Express CICS interface includes load modules for integration into the CICS subsystem as well as implementation examples. This interface enables a CISC application to manage files and communications with Connect:Express.

Normally, a CICS application does not have access to file management, but this is possible with the features listed below:

- Dynamic allocation, de-allocation, catalog, de-catalog, delete, of QSAM data set
- Open/Close of extra partition destinations (French BSIT application only)
- Selective access to Connect:Express journal file

A CICS application can use Connect:Express services to send a parameter list to the monitor to initiate a transfer request. A CICS application can also connect to the monitor, and then Connect:Express sends the journal record to the application.

During CICS initialization, a transaction is called to create the environment within CICS to communicate with Connect:Express. After the environment has been created, any CICS application can call Connect:Express to perform any of the functions described below. During CICS shutdown, an additional transaction is called to clean up this environment before CICS terminates.

Installing the CICS Option

The CICS option modules are provided on the standard product tape. To enable this option, it must be included in your asset protection key.

If an application uses the CICS interface, you must first initialize an environment in the CICS address space. This environment must then be released before CICS termination. Note that some programs are executed during CICS initialization and termination, while others are executed during communications between the CICS application and Connect:Express. A table (T3B2ZSSN) is provided to establish the communication between the user application and Connect:Express subsystem.

The installation of the Connect:Express OS/390 CICS option includes the following 3 phases.

1. Updating the CICS tables
2. Updating the T3B2ZSSN communication table
3. Updating the starting procedure or JCL of CICS

Updating the CICS Tables

To initialize the environment in the CICS address space, you must update the CICS tables, define programs from the Connect:Express product, and prepare for the initialization and termination processes.

CICS PPT

The following programs must be defined in the PPT of CICS with the option ASSEMBLY LANGUAGE.

Program	Description
L3B2ZALC	Dynamic allocation
L3B2ZAPL	Notification at end of transfer
L3B2ZEND	Disconnection from Connect:Express
L3B2ZUSJ	Access to Connect:Express journal file
L3B2ZZ20	Transfer request
L3B2ZJNL	Access to Connect:Express journal file
P3B2Z001	Initialization (load T3B2ZSSN)
P3B2Z002	Initialization
P3B2Z003	Normal termination
P3B2Z009	Clear interface
T3B2ZSSN	Communication table
B3TP906	Replaces L3B2ZOCP. Open/Close of extra partition destinations (French BSIT context only)

The following programs must be defined in the PPT of CICS with the option EXECKEY=CICS:

Program	Description
L3B2ZEND	Disconnect from Connect:Express
P3B2Z001	Initialization (load T3B2ZSSN)
P3B2Z002	Initialization
P3B2Z003	Normal termination
P3B2Z009	Clear interface

CICS PCT

The following transactions must be defined in the PCT of CICS, and have the option TASKDATAKEY=CICS.

Transaction	Description
TCIS	First program - P3B2Z001
TCIN	First program - P3B2Z002
TCIE	First program - P3B2Z003

Note: You can change the transaction names.

CICS PLTPI

For an automatic initialization of the Connect:Express OS/390 CICS interface, the P3B2Z001 program name must be added to the PLTPI list.

CICS PLTSD

The P3B2Z003 program is used to stop the Connect:Express OS/390 CICS interface before terminating CICS. To automate this process, the P3B2Z003 program name must be added to the PLTSD list at the first stage.

The programs defined in the PLTSD do not run when an Immediate Shutdown request is made to CICS. The P3B2Z009 program is then used to clear the CICS interface.

Updating the T3B2ZSSN Table

The T3B2ZSSn module is a table loaded by the P3B2Z001 program during CICS PLTPI initialization. The T3B2ZSSN table defines the following items.

- The Connect:Express subsystem name (SSN).
- The name of the initialization transaction which is automatically started by the P3B2Z001 program during PLTPI.

An example of the T3B2ZSSN table is provide in the *SAMPOPT* library of the product. The table T3B2ZSSN (see EX£T3SSN) supplied with the installation materials defines TOM1 as the subsystem name and TCIN as the initialization transaction name.

Note: Several CICS can use the same Connect:Express. In this case, each CICS must have a separate LOADLIB with its own T3B2ZSSN module. Each CICS can reference the same Connect:Express subsystem, but with a different application name.

The initialization transaction TCIN can be the same for all applications. You can check the list of applications through the Connect:Express TSO/ISPF option 2.5.

The P3B2Z009 program is used to clear the CICS interface. When several CICS are using the same Connect:Express subsystems, they must be coded with the 'APL=' parameter.

Updating the CICS Start Procedure

The *LOADLIB* of Connect:Express must be concatenated to both STEPLIB and DFHRPL of the CICS step. The *LOADLIB* of Connect:Express is APF.

When using the L3B2ZUSJ program, you must add a DD statement referring to the Connect:Express journal file, as shown below.

```
//SYSJNL DD DSN=XXXX.TOMn.SYSJNL,DISP=SHR
```

You must also add a step to execute the P3B2Z009 clearing program after the CICS step to allow the interface to stop normally if this operation was not successfully completed by the P3B2Z003 program.

```
//CLEAR EXEC PGM=P3B2Z009,COND=EVEN
```

OR

```
//CLEAR EXEC PGM=P3B2Z009,COND=EVEN,PARM='SSN=TOM?,APL=xxxxxxxx'
```

CICS Interface Transactions

Before running your application in the CICS environment, you must initialize the Connect:Express CICS interface. When CICS terminates, it must terminate the Connect:Express CICS interface. If problems occur when initializing the CICS interface, you can clean up the control blocks of the Connect:Express subsystem interface. Standard transactions and utilities are provided.

CICS Interface Initialization

The installation of the environment is performed by a transaction called TCIN. TCIN can be activated either automatically by PLTPI or by the TCIS transaction. In both cases, the initialization process must be completed before starting the user application. This transaction remains active until the interface termination.

CICS applications connected to the monitor are displayed on the Connect:Express Application (CICS,IMS,...) screen. (TSO/ISPF option 2.5)

CICS Interface Termination

The environment created by the TCIN transaction is released when this transaction terminates, either automatically by running the P3B2Z003 program declared in the PLTSD list during the first stage of Shutdown or by the TCIE transaction.

For an immediate shutdown, it is necessary to use the TCIE transaction to terminate the Connect:Express CICS interface before stopping CICS.

CICS Interface Troubles

The control blocks of the Connect:Express subsystem interface may be altered in the following situations:

❖ The environment is not well initialized.

❖ CICS has been canceled without running the P3B2Z009 program.

❖ An error occurred while processing the interface (initialization, call, termination).

In these cases, it is recommended to execute the steps below:

1. Pass the '$$LOAD$$' command to Connect:Express or run 'P1B2Z9RL' program.
2. Stop both CICS and Connect:Express.
3. Restart both CICS and Connect:Express.

Using the Connect:Express OS/390 CICS Interface

Once the Connect:Express OS/390 CICS environment has been initialized, a user program can call the interface to access Connect:Express services.

The modules provided are written in ASSEMBLY “COMMAND LEVEL” and can be called by any program written in COBOL, ASSEMBLY, or PL1 using the standard conventions of CICS COMMAND LEVEL, as shown below.

```
EXEC CICS LINK PROGRAM('L3B2Zxxx')
     COMMAREA(parmlist) LENGTH(parmlng)

L3B2Zxxx  =  called module name
parmlist  =  communication area acquired by the
             calling program and given to the called module
parmlng   =  length of the communication area
```

For COBOL programs, communication areas are defined in the *MACLIB* provided with the installation material and can be inserted using COPY commands.

Managing Files

Normally, a CICS application does not have access to file management. This is possible using L3B2ZALC for dynamic file allocation and de-allocation, and L3B2ZUSJ for selective access to the Connect:Express Journal File. These programs are discussed in the next two sections.

L3B2ZALC: Dynamic File Allocation/De-allocation

The following COBOL example defines the communication area structure.

```
01   Connect:Express  -ALC-PARMLIST.
    03 ALCSSNA        PICTURE   X(4).
    03 ALCPARML.
       05 ALCFUNC     PICTURE   X(1).
       05 ALCDDNM     PICTURE   X(8).
       05 ALCDSNM     PICTURE   X(44).
       05 ALCMEMB     PICTURE   X(8).
       05 ALCDISP     PICTURE   X(3).

       05 ALCVOLS     PICTURE   X(6).
       05 ALCUNIT     PICTURE   X(5).
       05 ALCSPAC     PICTURE   X(1).
       05 ALCPRIM     PICTURE   S9(2)      COMPUTATIONAL.
       05 ALCSECD     PICTURE   S9(2)      COMPUTATIONAL.
       05 ALCBLKS     PICTURE   S9(2)      COMPUTATIONAL.
       05 ALCLREC     PICTURE   S9(2)      COMPUTATIONAL.
       05 ALCRECF     PICTURE   X(2).

       05 ALCLRTCD    PICTURE   S9(2)      COMPUTATIONAL.
       05 ALCLRSCD    PICTURE   X(4).
```

The function to be processed determines the fields that you must complete. The tables below list the fields that you complete for each function.

Allocate a sequential file already catalogued

Field	Value
ALCFUNC	A
ALCDDNM	DD name padded with blank
ALCDSNM	Data set name padded with blank
ALCDISP	SHR or OLD

Deallocate a File

Field	Value
ALCFUNC	U
ALCDDNM	DD name padded with blank

Delete/decatalog a file allocated by user program

Field	Value
ALCFUNC	D
ALCDDNM	DD name padded with blank

Delete/decatalog of file

Field	Value
ALCFUNC	D
ALCDDNM	Blank
ALCDSNM	Data set name padded with blank

Allocate a Member

Field	Value
ALCFUNC	A
ALCDDNM	DD name padded with blank
ALCDSNM	Data set name padded with blank
ALCDISP	SHR or OLD
ALCMEMB	Member name padded with blank

Allocate a new Sequential File

Field	Value
ALCFUNC	A
ALCDDNM	DD name padded with blank
ALCDSNM	Data set name padded with blank
ALCDISP	NEW
ALCVOLS	Volume serial number
ALCUNIT	Unit name
ALCSPAC	Space type 'T' or 'C'
ALCPRIM	Primary allocation
ALCSECD	Secondary allocation

Field	Value
ALCBLKS	Block size
ALCLREC	Record length
ALCRECF	Record format

Allocation Return Codes

The table below describes the possible allocation return codes.

ALCRTCD	Description
0	Request successfully executed
1	ALCFUNC not equal to A, U, or D
2	ALCDISP not equal to OLD, SHR, or NEW
3	ALCDDNM not given
4	ALCDSNM omitted
5	ALCSPAC not equal to C or T
6	ALCPRIM zero or < zero
7	ALCSECD zero or < zero
8	ALCVOLS omitted
9	ALCUNIT omitted
10	ALCBLKS zero or < zero
11	ALCLREC zero or < zero
12	ALCRECF zero or < zero
16	SVC99 return code given in field ALCRSCD
20	Internal error, see field ALCRSCD

Reason Codes

The following table describes the possible reason codes with L3B2ZALC.

ALCRSCD	Description
9 302	Subsystem unknown or not started since last IPL
9 303	Connect:Express CICS notification interface not initiated
9 304	Request queue full
9 306	Invalid function
9 307	Notification interface (APL) already in WAIT state
9 308	Connect:Express CICS notification interface attach error
9 309	Notification interface (APL) request pending
9 310	Forced by user
9 311	Notification interface (APL) environment not initiated
9 312	Notification interface already attached
9 313	Notification interface (APL) queue full

Abend Codes

The table below describes the possible abend codes with L3B2ZALC.

ALCRSCD	Description
T301	DFHCOMMAREA not found
T302	Invalid subsystem
T303	Connect:Express CICS Interface not initialized (L3B2Z040)
T304	Connect:Express CICS Interface table full (gm2- L3B2Z040)
T305	Request queue full

L3B2ZUSJ: Selective Access to Connect:Express Journal File

The following COBOL example defines the communication area structure.

```
   01   Connect:Express  -USJ-PARMLIST.
              03 USJSSNA        PICTURE   X(4).
              03 USJPARML.
                 05 USJFUNC     PICTURE   X(1).
                 05 FILLER      PICTURE   X(3)      VALUE SPACES.
                 05 USJRETC.
                    07 USJRTCD  PICTURE   X(1)      VALUE ZERO.
                    07 USJRSCD  PICTURE   X(3)      VALUE ZERO.
                 05 USJNBRC     PICTURE   X(1)      VALUE ZERO.
                 05 USJLIBE     PICTURE   X(3)      VALUE SPACE.
              03 USJSLECT.
                 05 USJFILEN    PICTURE   X(8).
                 05 SCI-FUNC    PICTURE   X(1).
                 05 USJDSNAM    PICTURE   X(44).
                 05 USJDIREC    PICTURE   X(1).
                 05 USJSFCOD    PICTURE   X(5).
                 05 USJSFNAM    PICTURE   X(5).
                 05 USJPARID    PICTURE   X(8).
                 05 USJSORGT    PICTURE   X(1).
                 05 USJSORGN    PICTURE   X(5).
                 05 USJSDSTT    PICTURE   X(1).
                 05 USJSDSTN    PICTURE   X(5).
                 05 USJREQNB    PICTURE   X(8).
                 05 USJRDAT1    PICTURE   X(6).
                 05 USJRTIM1    PICTURE   X(6).
                 05 USJRDAT2    PICTURE   X(6).
                 05 USJRTIM2    PICTURE   X(6).
                 05 USJFILLR    PICTURE   X(9).
              03 USJRECOR.
                 05 FILLER      PICTURE   X(508)   VALUE SPACES.
```

The parameter area has the four sections listed below.

Section	Description
USJSSNA	Connect:Express subsystem name
USJPARML	Function codes and return codes
USJSLECT	Selection fields
USJRECOR	Journal record

The following table identifies the fields and possible values for each section.

Section	Field	Value
USJSSNA		Connect:Express subsystem name
USJPARML	USJFUNC	O open SYSJNL file. **Note**: The O function must precede one or several calls with the D function. The C function must be called after the D function indicates there is no record corresponding to the specified selection criteria (USJNBRC = 0).

Section	Field	Value
	Function code	D to obtain the first record at the first call, or the following records at successive calls. The records obtained successfully after each CALL with D function are placed in USJRECOR when they match the specified selection criteria. The field USJNBRC then is equal to 1. "C" Close SYSJNL file.
	FILLER	Reserved area
	USJRETC	Return code. If USJRETC is equal to zero this means that the request has been successfully completed. Otherwise, see the return codes list.
	USJNBRC	Can be equal to 0 or 1. **0** No or no more records of the Connect:Express journal file match the selection criteria. **1** One record is placed in the USJRECOR area.
	USJLIBE	Error message: blank means that the request has been successfully completed. Otherwise, see the error message list.
USJSLECT		Selection fields. It is possible to filter the journal file records by filling in the selection fields. A blank field means that any possible value is selected. The wild card * (asterisk) can be placed at the end of the selection string: for example, if DSNAM = SYST.BORDEAUX.*, then each record whose DSNAME field begins with "SYST.BORDEAUX" will be selected.
	USJFILEN	Symbolic name of the file
	USJDSNAM	Physical name of the file
	USJDIREC	Transfer direction, T for transmit, R for receive
	USJSFCOD	File type, for PeSIT only (used only in France)
	USJSFNAM	Logical file name, for PeSIT only
	USJPARID	Symbolic partner name
	USJSORGT	Origin type, for PeSIT only
	USJSORGN	Application ID, for PeSIT only
	USJSDSTT Journal record	Partner type
	USJSDSTN	Destination application ID, for PeSIT only
	USJREQNB	Request number
	USJRDAT1, USJRTIM1, USJRDAT2, USJRTIM2	Date and time selection: the selected records correspond to any transfers starting after RDAT1 date and RTIM1 time and ending before RDAT2 date and RTIM2 time.
USJRECOR		Journal record. See the Connect:Express OS/390 User Guide for more information about the journal record and D1B2PJNL structure.

Return Codes

USJRTCD	Description
0	Request successfully executed
not = 0	USJLIBE gives error message

9	Connect:Express CICS interface internal error
	USJLIBE = 'INV'USJFUNC not equal to O, D, C, or R
	USJLIBE = 'ACB'Generate ACB error
	USJLIBE = 'RPL'Generate RPL error
	USJLIBE = 'OPN'Open error
	USJLIBE = 'SHW'VSAM error
	USJLIBE = 'GET'READ error, USJRETC = RPL feed-back
	USJLIBE = 'CLS'Close error

Reason Codes

USJRSCD	Description
9 302	Subsystem unknown or not started since last IPL
9 303	Connect:Express CICS notification interface not initiated
9 304	Request queue full
9 306	Invalid function
9 307	Notification interface (APL) already in WAIT state
9 308	Connect:Express CICS notification interface attach error
9 309	Notification interface (APL) request pending
9 310	Forced by user
9 311	Notification interface (APL) environment not initiated
9 312	Notification interface already attached
9 313	Notification interface (APL) queue full

Abend Codes

USJRSCD	Description
T301	DFHCOMMAREA not found
T302	Invalid subsystem
T303	Connect:Express CICS Interface not initialized (L3B2Z040)
T304	Connect:Express CICS Interface table full (gm2- L3B2Z040)
T305	Request queue full

Managing CICS to Connect:Express Communications

A CICS application can use Connect:Express services to send a parameter list to the monitor to initiate a transfer request. A CICS application can also connect to the monitor, and then Connect:Express sends the journal record to the application. You can use the L3B2ZZ20 program to send a file transfer request to Connect:Express.

L3B2ZZ20: File Transfer Request to Connect:Express

The L3B2ZZ20 interface is based on the Application Program Interface described in Chapter 6 of the Connect:Express OS/390 User Guide. The following COBOL example shows the communication area structure.

```
01   Connect:Express  -REQ-PARMLIST.
    03 REQSSNA        PICTURE   X(4).

    03 REQ-SIT-PARML.
       05 SIT-FUNC    PICTURE   X(1)      VALUE  'T'.
       05 SIT-TYPE    PICTURE   X(1)      VALUE  'R'.
       05 SIT-REQN    PICTURE   X(8)      VALUE  '00000000'.
       05 FILLER      PICTURE   X(2)      VALUE  SPACES.
       05 SIT-RTCF    PICTURE   X(1)      VALUE  '0'.
       05 SIT-RSCF    PICTURE   X(3)      VALUE  '000'.
       05 SIT-EMSG    PICTURE   X(80)     VALUE  SPACES.

       05 SIT-PROT    PICTURE   X(1)      VALUE  'A'.
       05 SIT-CLAS    PICTURE   X(1).
       05 SIT-PRTY    PICTURE   9(1).
       05 SIT-RTYP    PICTURE   X(1)      VALUE  'N'
       05 FILLER      PICTURE   X(1)      VALUE  SPACES.

       05 SIT-ORGT    PICTURE   9(1).
       05 SIT-ORGN    PICTURE   9(5).
       05 SIT-DSTT    PICTURE   9(1).
       05 SIT-DSTN    PICTURE   9(5).

       05 SIT-DATE    PICTURE   9(6).
       05 SIT-TIME    PICTURE   9(6).

       05 SIT-FCOD    PICTURE   9(5).
       05 SIT-FNAM    PICTURE   X(5).
       05 SIT-DSNM    PICTURE   X(44).
       05 FILLER      PICTURE   X(133).

    03 REQ-SCI-PARML  REDEFINES REQ-SIT-PARML.
       05 SCI-FUNC    PICTURE   X(1).
       05 SCI-TYPE    PICTURE   X(1)      VALUE  'R'.
       05 SCI-REQN    PICTURE   X(8)      VALUE  '00000000'.
       05 FILLER      PICTURE   X(2)      VALUE  SPACES.
       05 SCI-RTCF    PICTURE   X(1)      VALUE  '0'.
       05 SCI-RSCF    PICTURE   X(3)      VALUE  '000'.
       05 SCI-EMSG    PICTURE   X(80)     VALUE  SPACES.

       05 SCI-PROT    PICTURE   X(1)      VALUE  'S'.
       05 SCI-CLAS    PICTURE   X(1).
       05 SCI-PRTY    PICTURE   9(1).
       05 SCI-RTYP    PICTURE   X(1).
       05 SCI-TDIR    PICTURE   X(1).
```

Continued

```
                 05 SCIFILNM    PICTURE   X(8).
                 05 SCIPARTN    PICTURE   X(8).
                 05 SCIDSNAM    PICTURE   X(44).
                 05 SCIMEMBR    PICTURE   X(8).
                 05 FILLER      PICTURE   X(144).
```

Two types of file transfer requests are supported. The PeSIT request which is only used in France, and the Standard request.

PeSIT Transfer Requests

The following table describes the fields and possible values for a PeSIT request.

Field	Value
REQSSNA	Subsystem name
SIT-CLAS	APM effector class
SIT-PRTY	Priority (0,1,2)
SIT-ORGT	Origin type (1,2)
SIT-ORGN	Origin number
SIT-DSTT	Destination type
SIT-DSTN	Destination number
SIT-DATE	Creation date
SIT-TIME	Creation time
SIT-FCOD	File type
SIT-FNAM	File logical name
SIT-DSNM	Physical name of the file

Standard Transfer Requests

The following table describes the fields and possible values for a standard transfer request.

Field	Value
SCI-CLAS	APM effector class
SCI-PRTY	Priority (0,1,2)
SCI-TDIR	Transfer direction (T,R)
SCIFILNM	Symbolic file name
SCIPARTN	Symbolic partner name
SCIDSNAM	Physical name of the file:edt.

Return Codes

For PeSIT transfer requests, the return codes, reason codes, and messages are returned in the fields SIT-RTCF, SIT-RSCF, and SIT-EMSG.

For standard transfer requests, the return codes, reason codes, and messages are returned in the fields SCI-RTCF, SCI -RSCF, and SCI -EMSG.

The 'EMSG' fields contain the error message or 'OK'. See Connect:Express HELP TRC on the TSO/ISPF screen or consult *Appendix B Error Codes and Messages* in the User Guide for a list of error codes.

Internal errors in the Connect:Express CICS interface are identified in the following tables.

Reason Codes (RTCF + RSCF)

Code	Description
9 302	Subsystem unknown or not started since last IPL
9 303	Connect:Express CICS notification interface not initiated
9 304	Request queue full
9 306	Invalid function
9 307	Notification interface (APL) already in WAIT state
9 308	Connect:Express CICS notification interface attach error
9 309	Notification interface (APL) request pending
9 310	Forced by user
9 311	Notification interface (APL) environment not initiated
9 312	Notification interface already attached
9 313	Notification interface (APL) queue full

Abend Codes (RTCF + RSCF)

Code	Description
T301	DFHCOMMAREA not found
T302	Invalid subsystem
T303	Connect:Express CICS Interface not initialized (L3B2Z040)
T304	Connect:Express CICS Interface table full (gm2- L3B2Z040)
T305	Request queue full

Managing Connect:Express to CICS Communications

You can use the L3B2ZAPL program to request transfer notifications from Connect:Express.

L3B2ZAPL: Transfer Notification Request.

The L3B2ZAPL interface is based on the Application Program Interface described in Chapter 6 of the Connect:Express OS/390 User Guide.

There are three steps to implement a transfer notification request.

1. The user application connects to Connect:Express at initialization to be identified.
2. The user application waits to receive a notification each time a transfer is completed. At each event (transmit-receive completion, normal or abnormal), Connect:Express immediately transmits the elements recorded in its journal file to the waiting transaction.
3. The application issues a disconnect request to Connect:Express.

The following example shows the communication area structure.

```
01   Connect:Express  -APL-PARMLIST.
    03 APLSSNA        PICTURE   X(4).
    03 APLPARML.
       05 APLAPLNM    PICTURE   X(8).
       05 APLREQST    PICTURE   X(1).
       05 APLRSRVD    PICTURE   X(3).
       05 APLREQRC    PICTURE   X(1).
       05 APLREQRS    PICTURE   X(3).
    03 APLANSWR.
       05 FILLER      PICTURE   X(508).
    03 APLANSW2       REDEFINES     APLANSWR.
       05 APLINTR-KEY PICTURE   X(4).
       05 APLINTR-COD PICTURE   X(4).
       05 FILLER      PICTURE   X(500).
```

The table below describes the fields in the communication area structure.

Field	Value
APLSSNA	Subsystem name of Connect:Express
APLAPLNM	Name of application
APLREQST	Function I, W, or T
APLRSRVD	Unused
APLREQRC	Return code
APLREQRS	Return code
APLANSWR	Reply area (see journal record format)
APLINTR-KEY	Reply key of journal for I and T
APLINTR-COD	Function key (INIT if It is OK, TERM if Connect:Express stops, FORC to force application stop)

Reason Codes (RC + RS)

Code	Description
9 302	**Subsystem unknown or not started since last IPL**
9 303	Connect:Express CICS notification interface not initiated
9 304	Request queue full
9 306	Invalid function
9 307	Notification interface (APL) already in WAIT state
9 308	Connect:Express CICS notification interface attach error
9 309	Notification interface (APL) request pending
9 310	Forced by user
9 311	Notification interface (APL) environment not initiated
9 312	Notification interface already attached
9 313	Notification interface (APL) queue full

Abend Codes (RC + RS)

Code	Description
T301	DFHCOMMAREA not found
T302	Invalid subsystem
T303	Connect:Express CICS Interface not initialized (L3B2Z040)
T304	Connect:Express CICS Interface table full (gm2- L3B2Z040)
T305	Request queue full

Chapter 2

IMS Interface

This chapter describes the IMS interface option which provides subroutine modules that can be called by an IMS application to communicate with Connect:Express OS/390. These subroutines enable an IMS application to request Connect:Express to perform certain functions on behalf of the application.

Overview

The IMS option of Connect:Express OS/390 enables an IMS application to manage files and communications with Connect:Express.

> Note: The IMS option modules are provided on the standard product tape. To enable this option, it must be included in your asset protection key.

Normally an IMS application does not have access to file management, however this is possible using the features listed below:

- Dynamic allocation, deallocation, catalog, decatalog, delete - of QSAM files
- Selective access to the Connect:Express journal file

An IMS application can use Connect:Express services by sending a parameter list to the monitor to initiate a transfer request or place the program in a wait state. An IMS application can also connect to the monitor, and then Connect:Express sends the journal record to the application.

During initialization of the IMS region that issues requests to Connect:Express, you must call a module which initializes an ESTAE environment. This environment ensures that the application disconnects from Connect:Express if the IMS region or task ABENDs.

Using the IMS Interface

After the Connect:Express IMS interface is enabled, a user program can use IMS services by calling one of the subroutine modules.

The modules are written in assembly and can be called by a user program running in a BMP or an MPP. The user program can be written in COBOL, ASSEMBLY, or PL1 using standard IBM conventions. For COBOL programs, communication areas are defined in the *MACLIB* provided with the installation material and can be inserted using COPY commands.

Because BMP and MPP are user applications that use IMS statements, they can use the Connect:Express interface module to communicate with Connect:Express. Because they are IMS regions, they can communicate with IMS.

Managing Files

Normally, an IMS application does not have access to file management. This is possible using L4B2ZALC for dynamic file allocation and de-allocation, and L0B2ZUSJ for selective access to the Connect:Express Journal File. These programs are discussed in the next two sections.

L4B2ZALC: Dynamic File Allocation/Deallocation

A user program running in BMP or MPP can call this module to perform any of the following functions:

- Dynamic allocation DISP=SHR
- Dynamic allocation DISP=NEW
- Scratch/decatalog

The screen below shows the communication area structure.

```
DCL 1 ALLOCATION_PARM ,
     2 ALLOCATION_FUNCTION,
        5 ALCFNC CHAR(1),          /*A:ALLOCATION,U:UNALLOC,D:DELETE*/
     2 ALLOCATION_DDNAME,
        5 ALCDDN CHAR(8),          /*DDNAME                        */
     2 ALLOCATION_DSN,
        5 ALCDSN CHAR(44),         /*DSN                           */
     2 ALLOCATION_MEMBER,
        5 ALCMBR CHAR(8),          /* MEMBER NAME                  */
     2 ALLOCATION_DISP,
        5 ALCDSP1 CHAR(3),         /* OLD/SHR/NEW                  */
 /*------------------------------
    NEXT PARMS: ONLY IF DISP=NEW
    -----------------------------*/
     2 ALLOCATION_VOLUME,
        5 ALCVOL CHAR(6),          /*VOL=SER=                      */
     2 ALLOCATION_UNIT,
        5 ALCUNT CHAR(5),          /*UNIT=                         */
     2 ALLOCATION_SPACE,
        5 ALCSPC CHAR(1),          /* SPACE T:TRACKS,C:CYLINDER    */
        5 ALCPRM BIN FIXED(15,0),  /* ALLOCATION PRIMAIRE          */
        5 ALCSEC BIN FIXED(15,0),  /* ALLOCATION SECONDAIRE        */
     2 ALLOCATION_DCB,
        5 ALCBLK BIN FIXED(15,0),  /* BLKSIZE=                     */
        5 ALCRCL BIN FIXED(15,0),  /* RECL=                        */
        5 ALCRFM CHAR(2),          /* RECFM= F/FB                  */
     2 ALLOCATION_CODE,
        5 ALCRTC BIN FIXED(15,0),  /* REASON CODE                  */
        5 ALCRSC CHAR(4);          /* RETURN CODE                  */
```

There are 5 functions that you can execute with this module. The following tables describe the values to enter for each function.

Dynamic Allocation of a Catalogued File

Field	Value
ALCFNC	'A'.
ALCDDN	DDNAME padded with blank
ALCDSN	Data set name padded with blank
ALCDSP1	SHR or OLD keywords

Dynamic Allocation of a Member

The partitioned data set must be catalogued, and the user must fill in the following fields.

Field	Value
ALCFNC	'A'
ALCDDN	DDNAME padded with blank
ALCDSN	Data set name padded with blank
ALCDSP1	SHR or OLD keywords
ALCMBR	Member name padded with blank

Dynamic Allocation of a New File

Field	Value
ALCFNC	'A'
ALCDDN	DDNAME padded with blank
ALCDSN	Data set name padded with blank
ALCDSP1	NEW keyword

All other parameters can be completed like a DD statement.

Note: The new data set is catalogued if the allocation was successful.

Dynamic De-allocation

The request must be done after closing the file.

Field	Value
ALCFNC	'U'
ALCDDN	DD name padded with blank

Scratch and de-catalog a file

The request must be done after closing the current file.

If the file has been previously allocated by a user program, complete the following fields.

Field	Value
ALCFNC	'D'
ALCDDN	DD name padded with blank

If the file was not previously allocated, complete the fields below.

Field	Value
FUNCTION	'D'
ALCDDN	Blank
ALCDSN	Data set name padded with blank

Return Codes and Messages

If the request was successfully executed, the field ALCRTC is equal to zero. The other possible values for ALCRTC are listed below.

Note: The error codes 5 to 12 are used for dynamic allocation of new files.

Execution Errors

Error Code	Description
1	FUNCTION not equal to A, U, or D
2	ALCDSP is invalid
3	ALCDDN is omitted
4	ALCDSN is omitted
5	ALCSPC not equal to T or C
6	ALCPRC equal or inferior to zero
7	ALCSEC equal or inferior to zero
8	ALCVOL is omitted
9	ALCUNT is omitted
10	ALCBLK equal to or less than zero
11	ALCRCL equal to or less than zero
12	ALCRFM equal to or less than zero

If ALCRTC = 16, the dynamic allocation or deallocation failed. In this case, the Return Code field contains the SVC99 return code, and a message appears on the screen.

```
L4B2ZALC dddddddd ffffffff ERR.CODE = xxxx
```

The table below describes each field.

Field	Description
Dddddddd	DD name
Ffffffff	"ALLOCATE" or "DESALLOC"
Xxxx	SVC99 return code

Example: Program Calling L4B2ZALC

The following screens show an example of an IMS application calling L4B2ZALC.

```
*PROCESS FLAG(I),A,S,GS,C,INC,NNUM,NEST;
  SAMPLE1: PROCEDURE OPTIONS(MAIN) ;
   /* *****************************************************
      SAMPLE PL/I PROGRAM  calling L4B2ZALC
    ***************************************************** */
  DCL PLIRETV BUILTIN;
  DCL PLIRETC BUILTIN;
  DCL L4B2ZALC EXTERNAL ENTRY OPTIONS (ASM,INTER,RETCODE) ;
  DCL SYSPRINT FILE STREAM OUTPUT ;

  DCL DDN FILE RECORD SEQUENTIAL INPUT ;
  DCL DDN2 FILE RECORD SEQUENTIAL OUTPUT ;
  DCL DDN3 FILE RECORD SEQUENTIAL OUTPUT ;

  DCL ZONE CHAR(80);
  DCL ZON2 CHAR(120);


  DCL 1 ALLOCATION_PARM ,
       2 ALLOCATION_FUNCTION,
          5 ALCFNC CHAR(1),          /*A:ALLOCATION,U:UNALLOC,D=DLET */
       2 ALLOCATION_DDNAME,
          5 ALCDDN CHAR(8),          /*DDNAME                        */
       2 ALLOCATION_DSN,
          5 ALCDSN CHAR(44),         /*DSN                           */
       2 ALLOCATION_MEMBER,
          5 ALCMBR CHAR(8),          /* MEMBER NAME                  */
       2 ALLOCATION_DISP,
          5 ALCDSP1 CHAR(3),         /* OLD/SHR/NEW                  */
   /*-------------------------------
      NEXT PARMS_: ONLY IF DISP=NEW
      -----------------------------*/
       2 ALLOCATION_VOLUME,
          5 ALCVOL CHAR(6),          /*VOL=SER=                      */
       2 ALLOCATION_UNIT,
          5 ALCUNT CHAR(5),          /*UNIT=                         */
       2 ALLOCATION_SPACE,
          5 ALCSPC CHAR(1),          /* SPACE T:TRACKS,C:CYLINDER    */
          5 ALCPRM BIN FIXED(15,0),  /* ALLOCATION PRIMAIRE          */
          5 ALCSEC BIN FIXED(15,0),  /* ALLOCATION SECONDAIRE        */
       2 ALLOCATION_DCB,
          5 ALCBLK BIN FIXED(15,0),  /* BLKSIZE=                     */
          5 ALCRCL BIN FIXED(15,0),  /* RECL=                        */
          5 ALCRFM CHAR(2),          /* RECFM= F/FB                  */
       2 ALLOCATION_CODE,
          5 ALCRTC BIN FIXED(15,0),  /* REASON CODE                  */
          5 ALCRSC CHAR(4);          /* RETURN CODE                  */
```

Continued

```
/*-----------------------------------------------*/
 PUT SKIP LIST(' ALLOCATION CATALOGUED FILE');
 ALCFNC='A';
 ALCDDN='DDN     ';
 ALCDSN='PSR$TST.MNF.BBBB';
 ALCDSP1='SHR';
 CALL L4B2ZALC (ALLOCATION_PARM) ;

 IF ALCRTC = 0 THEN GOTO ERROR;

 OPEN FILE(DDN) ;
 READ FILE(DDN) INTO(ZONE);
 PUT SKIP LIST(' ZONE',ZONE) ;


/*-----------------------------------------------*/
 PUT SKIP LIST(' ALLOCATION NEW FILE');
 ALCFNC='A';
 ALCDDN='DDN2';
 ALCDSN='PSR$TST.MNF.BBB2';
 ALCDSP1='NEW';
 ALCVOL='PSR002';
 ALCUNT='3380';
 ALCSPC='T';
 ALCPRM=1;
 ALCSEC=0;
 ALCRCL=120;
 ALCBLK=120;
 ALCRFM=' F';
 CALL L4B2ZALC (ALLOCATION_PARM) ;

 IF ALCRTC = 0 THEN GOTO ERROR;

 ZON2=ZONE;

 OPEN FILE(DDN2) ;
 WRITE FILE(DDN2) FROM(ZON2);


/*-----------------------------------------------*/
 PUT SKIP LIST(' ALLOCATION SECOND NEW FILE');
 ALCFNC='A';
 ALCDDN='DDN3';
 ALCDSN='PSR$TST.MNF.BBB3';
 ALCDSP1='NEW';
 ALCVOL='PSR002';
 ALCUNT='3380';
 ALCSPC='C';
 ALCPRM=1;
 ALCSEC=1;


 ALCRCL=120;
 ALCBLK=1200;
 ALCRFM=' F';
 CALL L4B2ZALC (ALLOCATION_PARM) ;

 IF ALCRTC = 0 THEN GOTO ERROR;
```

Continued

```
OPEN FILE(DDN3) ;
 WRITE FILE(DDN3) FROM(ZON2);


/*-----------------------------------------------*/
 CLOSE FILE(DDN);
 PUT SKIP LIST(' UNALLOC FILE BY DDN');
 ALCDDN='DDN';
 ALCFNC='U';
 CALL L4B2ZALC (ALLOCATION_PARM) ;

 IF ALCRTC = 0 THEN GOTO ERROR;
/*-----------------------------------------------*/
 PUT SKIP LIST(' DELETE NO CURRENTLY ALLOCATED DATA SET');
 ALCDDN='        ';
 ALCDSN='PSR$TST.MNF.BBBB                            ';
 ALCFNC='D';
 CALL L4B2ZALC (ALLOCATION_PARM) ;

 IF ALCRTC = 0 THEN GOTO ERROR;

/*-----------------------------------------------*/
 PUT SKIP LIST(' DELETE CURRENTLY ALLOCATED DATA SET');
 CLOSE FILE(DDN3);
 ALCDDN='DDN3    ';
 ALCDSN='      ';
 ALCFNC='D';
 CALL L4B2ZALC (ALLOCATION_PARM) ;

 IF ALCRTC = 0 THEN GOTO ERROR;

 GOTO FIN;

ERROR:
  PUT SKIP LIST (' FUNCTION ERROR: '||ALCFNC||',RETURN CODE='
   ||ALCRTC||',REASON CODE='||ALCRSC);

FIN:
 END ;
```

L0B2ZUSJ: Access to Connect:Express Journal

A user program running in BMP or MPP can call this module to access the Connect:Express Journal file. You can retrieve a record by record number or select a list of records with selection criteria.

Note: Assembler usage is recommended with this module, rather than COBOL or PL/I.

The following PL/I example defines the communication area structure.

```
   DCL  01  JOURNAL-COMMUNICATION,
              03 JOURNAL-PARM,
                 05 JNL-PRM-FNC    CHAR,
                 05 JNL-PRM-FILLER CHAR(3),
                 05 JNL-PRM-RTCOD  CHAR(4),
                 05 JNL-PRM-NBR    CHAR(1),
                 05 JNL-PRM-RSCOD  CHAR(3),
              03 JOURNAL-SELECTION,
                 05 JNL-SLC-FILEN CHAR(8),
                 05 JNL-SLC-DSNAM CHAR(44),
                 05 JNL-SLC-DIREC CHAR(1),
                 05 JNL-SLC-PSFTY CHAR(5),
                 05 JNL-SLC-PSNAM CHAR(5),
                 05 JNL-SLC-PARTN CHAR(8),
                 05 JNL-SLC-LOTYP CHAR,
                 05 JNL-SLC-LOAPN CHAR(5),
                 05 JNL-SLC-PATYP CHAR,
                 05 JNL-SLC-PAAPN CHAR(5),
                 05 JNL-SLC-REQNB CHAR(8),
                 05 JNL-SLC-MNDAT CHAR(6),
                 05 JNL-SLC-MNTIM CHAR(6),
                 05 JNL-SLC-MXDAT CHAR(6),
                 05 JNL-SLC-MXTIM CHAR(6),
                 05 JNL-SLC-FILLR CHAR(9),
              03 JOURNAL-ANSWER,
                 05 JNL-ANS-RECORD CHAR(508);
```

The parameter area includes the following three sections:

- JOURNAL-PARM: function codes and return codes
- JOURNAL-SELECTION: Selection fields
- JOURNAL-ANSWER: Journal record

When a program calls L0B2ZUSJ, a DD statement must refer to the Connect:Express journal file. Then, the user program submits a request to open the journal file, enters a request for one or more records, and sends a request to close the journal file.

The parameter area, JOURNAL-PARM, has the following fields.

Field	Value
JNL-PRM-FNC	Function code. The following values are valid. **O** Request for opening the journal file. **D** To obtain the first record at the first call, or the following records at the next calls. The records obtained successfully after each call with the D function are placed in JNL-ANS-RECORD when they match the selection criteria. The field JNL-PRM-NBR is equal to 1. **R** To obtain directly the record number x, this number must be put in hexadecimal XL4 in USJFILEN first positions. **C** Request for closing the journal file. The C function must be called as soon as the D function indicates that no record matches the specified selection criteria. The field JNL-PRM-NBR is equal to zero.
JNL-PRM-FILLER	Reserved
JNL-PRM-RTCOD	Return code
JNL-PRM-NBR	This field is equal to 1 if the selection is successful and 0 if no matching record is found.
JNL-PRM-RSCOD	Error message or blank as listed next page

You can filter the journal file records by filling in the selection fields. Complete only the fields that you want to use as selection criteria. You can use the wild card * (asterisk) at the end of the selection string. For example, if JNL-SLC-DSNAM = SYST.BORDEAUX.*, then each record whose DSNAME field begins with "SYST.BORDEAUX" is selected. The selection area, JOURNAL-SELECTION, includes the following fields.

Field	Value
JNL-SLC-FILEN	Symbolic file name
JNL-SLC-DSNAM	Data set name of file
JNL-SLC-DIREC	Transfer direction (T for transmit, R for receive)
JNL-SLC-PSFTY	File type (PeSIT)
JNL-SLC-PSNAM	Logical file name (PeSIT)
JNL-SLC-PARTN	Symbolic partner name (PeSIT)
JNL-SLC-LOTYP	Origin type (PeSIT)
JNL-SLC-LOAPN	Origin application number (PeSIT)
JNL-SLC-PATYP	Destination type (PeSIT)
JNL-SLC-PAAPN	Destination application number (PeSIT)
JNL-SLC-REQNB	Number of the request
JNL-SLC-MNDAT	Date and time selection: The selected records
JNL-SLC-MNTIM	correspond to transfers starting after JNL-SLC-MNDAT
JNL-SLC-MXDAT	date and JNL-SLC-MNTIM time and ending before
JNL-SLC-MXTIM	JNL-SLC-MXDAT date and JNL-SLC-MXTIM time

The results of your record search display in the JOURNAL-ANSWER section.

Return Codes

The JNL-PRM-RSCOD field indicates the error type. The table below lists the possible codes.

Code	Description
INV	Function code not equal to O, D, R or C
ACB	ACB generation failed
RPL	RPL generation failed
OPN	SYSJNL open error
SHW	VSAM error, SHOWCB failed
GET	Read error, RTCOD is equal to RPL "feedback"
CLS	Close error

Example: Program Calling L0B2ZUSJ

```
*PROCESS FLAG(I),A,S,GS,C,INC,NNUM,NEST;
 SAMPLE5: PROCEDURE OPTIONS(MAIN);

   /***********************************************************/
   /** L0B2ZUSJ calling sample                               **/
   /** to read all  Connect:Express  journal records         **/
   /***********************************************************/

  DCL L0B2ZUSJ EXTERNAL ENTRY OPTIONS (ASM,INTER,RETCODE);

  DCL  01  JOURNAL-COMMUNICATION,
           03 JOURNAL-PARM,
              05 JNL-PRM-FNC    CHAR,     /* O:OPEN,D:DISPLAY,C:CLOSE */
              05 JNL-PRM-FILLER CHAR(3),
              05 JNL-PRM-RTCOD  CHAR(4),  /* BLANC:OK  ERROR CODES    */
              05 JNL-PRM-NBR    CHAR(1),  /*v0:NO MORE REC.,1:ONE REC.*/
              05 JNL-PRM-RSCOD  CHAR(3),  /* BLANC:OK  ERROR CODES    */
           03 JOURNAL-SELECTION,
              05 JNL-SLC-FILEN CHAR(8),
              05 JNL-SLC-DSNAM CHAR(44),
              05 JNL-SLC-DIREC CHAR(1),
              05 JNL-SLC-PSFTY CHAR(5),
              05 JNL-SLC-PSNAM CHAR(5),
              05 JNL-SLC-PARTN CHAR(8),
              05 JNL-SLC-LOTYP CHAR,
              05 JNL-SLC-LOAPN CHAR(5),
              05 JNL-SLC-PATYP CHAR,
              05 JNL-SLC-PAAPN CHAR(5),
              05 JNL-SLC-REQNB CHAR(8),
              05 JNL-SLC-MNDAT CHAR(6),
              05 JNL-SLC-MNTIM CHAR(6),
              05 JNL-SLC-MXDAT CHAR(6),
              05 JNL-SLC-MXTIM CHAR(6),
           03 JOURNAL-ANSWER,
              05 JNL-ANS-RECORD CHAR(512);


  /* criteria-selected fields filled by
      appropriate value or by blanks     */

              JNL-SLC-FILEN=' ';
              JNL-SLC-DSNAM=' ';
              JNL-SLC-DIREC=' ';
              JNL-SLC-PSFTY=' ';
              JNL-SLC-PSNAM=' ';
              JNL-SLC-PARTN=' ';
              JNL-SLC-LOTYP=' ';
              JNL-SLC-LOAPN=' ';
              JNL-SLC-PATYP=' ';
              JNL-SLC-PAAPN=' ';
              JNL-SLC-REQNB=' ';
              JNL-SLC-MNDAT=' ';
              JNL-SLC-MNTIM=' ';
              JNL-SLC-MXDAT=' ';
              JNL-SLC-MXTIM=' ';
```

Continued

```
/*--------OPEN SYSJNLP-----------*/

        JNL-PRM-FNC='O';

        CALL L0B2ZUSJ (JOURNAL-PARM);

    IF JNL-PRM-RSCOD=' ' | JNL-PRM-RTCOD='0000' THEN GOTO ERROR;

/*--------DISPLAY-RECORDS --------*/

        JNL-PRM-FNC='D';

DISPLAY:

        CALL L0B2ZUSJ (JOURNAL-PARM);

    IF JNL-PRM-RSCOD=' ' | JNL-PRM-RTCOD='0000' THEN GOTO ERROR;
        IF JNL-PRM-NBR-= '0' THEN GOTO CLOSE;

        PUT SKIP DATA(JOURNAL-ANSWER);


        GOTO DISPLAY;

/*--------CLOSE SYSJNL ----------*/
CLOSE:

        JNL-PRM-FNC='C';

        CALL L0B2ZUSJ (JOURNAL-PARM);

    IF JNL-PRM-RSCOD=' ' | JNL-PRM-RTCOD='0000' THEN GOTO ERROR;

    ERROR:
      PUT SKIP LIST('ERROR FCT='||JNL-PRM-FNC||' RSCOD='||JNL-PRM-RSCOD);

FIN:

    END;
```

Managing IMS to Connect:Express Communications

An IMS application can use Connect:Express services by sending a parameter list to the monitor to initiate a transfer request or place the program in a wait state. You can use the L4B2ZWAI program to give the program a Wait status, and the L0B2ZZ20 program to send a transfer request.

L4B2ZWAI: Set Program in Wait State

A user program running in a BMP or MPP can call this module. The following screen shows the communication area structure.

```
DCL 1 WAIT_PARM,
     2 WAIT_SECOND,
        5 WAISCN BIN FIXED (31,0);
```

The value in the field WAISCN must be entered in seconds, and no return code is issued at the end of execution.

Example: Program Calling L4B2ZWAI

The following screen shows an example of an IMS application calling L4B2ZWAI.

```
*PROCESS FLAG(I),A,S,GS,C,INC,NNUM,NEST;
  BSIT07 : PROCEDURE OPTIONS(MAIN) ;
  DCL PLIRETV BUILTIN;
  DCL PLIRETC BUILTIN;
  DCL L4B2ZWAI EXTERNAL ENTRY OPTIONS (ASM,INTER,RETCODE) ;
  DCL SYSPRINT FILE STREAM OUTPUT ;

  DCL CALLRC BIN FIXED (31,0);

  DCL 1 WAIT_PARM ,
       2 WAIT_SECONDE,
          5 WAISCN BIN FIXED(31,0);   /*SECONDS   */

 /*------------------------------------------------*/
  PUT SKIP LIST(wait for 15 seconds);
  WAISCN=15;
  CALL L4B2ZWAI (WAIT_PARM) ;

  CALLRC = PLIRETV ;
  PUT SKIP LIST(' CALL-RC',CALLRC) ;

 /*------------------------------------------------*/
  PUT SKIP LIST(wait for 20 seconds);
  WAISCN=20;
  CALL L4B2ZWAI (WAIT_PARM) ;

  CALLRC = PLIRETV ;
  PUT SKIP LIST(' CALL-RC',CALLRC) ;

  END;
```

L0B2ZZ20: Transfer Request

The L0B2ZZ20 interface is based on the Application Program Interface described in Chapter 6 of the *Connect:Express OS/390 User Guide*.

A user program running in a BMP or MPP can call this module. The following COBOL example shows the communication area structure.

```
         01  REQUEST-PARM.
              03 TOMMONN  PICTURE X(4).
              03 TOMFUNC  PICTURE X(1)  VALUE 'T'.
              03 TOMTYPE  PICTURE X(1)  VALUE 'R'.
              03 TOMREQN  PICTURE X(8)  VALUE ZERO.
              03 TOMFIL1  PICTURE X(2).
              03 TOMRTCF  PICTURE X(1)  VALUE ZERO.
              03 TOMRSCF  PICTURE X(3)  VALUE ZERO.
              03 TOMEMSG  PICTURE X(80) VALUE  SPACES.

              03 TOMASIT  PICTURE X(1)  VALUE 'A'.
              03 TOM-CLAS PICTURE X(1).
              03 TOM-PRTY PICTURE X(1)  VALUE '1'.
              03 TOM-RTYP PICTURE X(1)  VALUE 'N'.
              03 TOM-FIL1 PICTURE X(1)  VALUE ' '.
              03 TOM-ORGT PICTURE X.
              03 TOM-ORGN PICTURE X(5).
              03 TOM-DSTT PICTURE X.
              03 TOM-DSTN PICTURE X(5).
              03 TOM-DATE PICTURE X(6).
              03 TOM-TIME PICTURE X(6).
              03 TOM-FCOD PICTURE X(5).
              03 TOM-FNAM PICTURE X(5).
              03 TOM-DSNM PICTURE X(44).
              03 FILLER   PICTURE X(128).
```

If the request is accepted, then TOMRTCF and TOMRSCF are equal to zero and the TOMREQN field contains the Connect:Express request number.

If the request is rejected, then TOMRTCF is equal to 2 and TOMRSCF contains the reason code. TOMEMSG contains an error message or 'OK'. Refer to Connect:Express HELP TRC on the TSO/ISPF screen or *Appendix B Error Codes and Messages* in the User guide for a list of error codes and messages.

Example: Program Calling L0B2ZZ20 for a PeSIT Transfer Request

```
IDENTIFICATION DIVISION.
PROGRAM-ID.  'TOMUSER'.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.

INPUT-OUTPUT SECTION.
FILE-CONTROL.
     SELECT PRT ASSIGN TO UT-S-PRT.

DATA DIVISION.
FILE SECTION.
FD   PRT         LABEL RECORDS ARE OMITTED
                 DATA RECORD IS TOMLNE
                 RECORDING MODE IS F.
01   TOMLNE.
     02   ASA    PICTURE X(1)  VALUE '1'.
     02   T1     PICTURE X(16) VALUE 'REQUEST NUMBER: '.
     02   C1     PICTURE X(8).
     02   T2     PICTURE X(13) VALUE 'RETURN CODE: '.
     02   C2     PICTURE X(2).
     02   T3     PICTURE X(13) VALUE 'REASON CODE: '.
     02   C3     PICTURE X(2).


WORKING-STORAGE SECTION.

01  REQUEST-PARM.
    03 TOMMONN PICTURE X(4).
    03 TOMFUNC PICTURE X(1) VALUE 'T'.
    03 TOMTYPE PICTURE X(1) VALUE 'R'.
    03 TOMREQN PICTURE X(8) VALUE '00000000'.
    03 TOMFIL1 PICTURE X(2) VALUE '  '.
    03 TOMRTCF PICTURE X(2).
    03 TOMRSCF PICTURE X(2).

    03 TOMASIT   PICTURE X(1)  VALUE 'A'.
    03  Connect:Express -CLAS PICTURE X(1) VALUE 'B'.
    03  Connect:Express -PRTY PICTURE X(1) VALUE '1'.
    03  Connect:Express -RTYP PICTURE X(1) VALUE 'N'.
    03  Connect:Express -FIL1 PICTURE X(1) VALUE ' '.
    03  Connect:Express -ORGT PICTURE X.


    03  Connect:Express -ORGN PICTURE X(5).
    03  Connect:Express -DSTT PICTURE X.
    03  Connect:Express -DSTN PICTURE X(5).
    03  Connect:Express -DATE PICTURE X(6).
    03  Connect:Express -TIME PICTURE X(6).
    03  Connect:Express -FCOD PICTURE X(5).
    03  Connect:Express -FNAM PICTURE X(5).
    03  Connect:Express -DSNM PICTURE X(44).
    03 FILLER    PICTURE X(144).
```

Continued

```
       PROCEDURE DIVISION.


           OPEN OUTPUT PRT.



           NOTE ****  SET  Connect:Express  SUBSYSTEM NAME *************.
           MOVE 'TOM4'  TO TOMMONN.

           NOTE ****  SET ORIGIN TYPE AND APPLICATION NUMBER  *****.
           MOVE '1'     TO  Connect:Express -ORGT.
           MOVE '11111' TO  Connect:Express -ORGN.

           NOTE ****  SET DEST.  TYPE AND APPLICATION NUMBER  *****.
           MOVE '1'     TO  Connect:Express -DSTT.
           MOVE '22222' TO  Connect:Express -DSTN.

           NOTE ****  SET DATE AND TIME ********************.
           MOVE '010188' TO  Connect:Express -DATE.
           MOVE '081256' TO  Connect:Express -TIME.

           NOTE ****  SET SYMBOLIC FILENAME AND DSNAME *****.
           MOVE '12345'   TO  Connect:Express -FCOD.
           MOVE '12345'   TO  Connect:Express -FNAM.
           MOVE 'A.B.C'   TO  Connect:Express -DSNM.

           CALL 'L0B2Z20' USING REQUEST-PARM.


           NOTE **** WRITE RETURN CODE AND REQUEST NUMBER ********.
           MOVE  TOMREQN TO  C1.
           MOVE  TOMRTCF TO  C2.
           MOVE  TOMRSCF TO  C3.
           WRITE TOMLNE.

           CLOSE PRT.

           NOTE *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
             *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *.

           STOP RUN.
```

Managing Connect:Express to IMS Communications

An IMS application can connect to the monitor, and then Connect:Express sends the journal record to the application. You use the L0B2ZAPL program to request transfer notifications, and the L4B2ZAPO program to initialize the ESTAE environment for L0B2ZAPL. The next two sections discuss these programs.

L0B2ZAPL: Transfer Notification Request

The L0B2ZAPL interface is based on the Application Program Interface described in Chapter 6 of the Connect:Express OS/390 User Guide.

A user program running in a BMP or an MPP calls this module and performs the functions listed below:

❖ The program connects to the Connect:Express interface

❖ The program waits for notification from Connect:Express about completed transfers

- The programs issues a disconnect request to Connect:Express

The following example shows the communication area structure.

```
01  REQUEST-L0B2ZAPL.

 05 APLTOMS  PICTURE X(4)    VALUE 'TOMP'. ---- TO ----> L0B2ZAPL
 05 APLNAME  PICTURE X(8)    VALUE 'BSIT-IMS'. --------> L0B2ZAPL
 05 APLFUNC  PICTURE X(1)    VALUE 'I'. or 'W' or 'T' -> L0B2ZAPL
 05 APLRSRV  PICTURE X(3)    VALUE LOW-VALUE.
 05 APLRTCD  PICTURE X(1)    VALUE ZERO.   <--- FROM ----L0B2ZAPL
 05 APLRSCD  PICTURE X(3)    VALUE ZERO.   <-------------L0B2ZAPL
 05 APLRJNL  PICTURE X(512)  VALUE SPACES. <-------------L0B2ZAPL
```

For each call to the L0B2ZAPL module, the fields must be completed or initialized as follows.

Field	Value
APLTOMS	Subsystem name.
APLNAME	Eight-character name padded with spaces.
APLFUNC	Function requested: **I** - Initialization. This function connects the user application to the Connect:Express interface. **W** - Wait notification. This function puts the user program in wait event from Connect:Express. At each end-of-transfer (normal or abnormal transmit-receive completion), Connect:Express immediately transmits the elements recorded on its journal file. The wait function can be repeated as many times as necessary. **T** - Termination. This function disconnects the user application from the Connect:Express interface.
APLRSRV	Reserved. Initialized with blanks.
APLRTCD	Initialized with ZERO (Connect:Express return code).
APLRSCD	Initialized with ZERO (Connect:Express reason code).
APLRJNL	Initialized with space (Connect:Express journal area).

Return Codes

Each time a user program calls the program L0B2ZAPL, the user program receives and processes a response. If the return code is not 0, the operation failed. A reason code may also be returned. The table below describes the (RTC) and reason (RSC) codes.

Code	Description

RTC = 0
RSC = 000
	Function I, W or T
	Request successfully executed

RTC = 1
RSC = 001
RSC = 002
RSC = 003
RSC = 004
	Subsystem error
	Subsystem not defined
	End of SSCVT chain
	TOMn not started since last IPL
	T1B2PAPL table not initialized

RTC = 2
RSC = 999
RSC = 888
RSC = 777
	Parameter error
	APLTOMSS is not valid
	APLNAME is not valid
	Function code not equal to I, W or T

Code	Description
RTC = 4	Initialization error
RSC = 666	T1B2PAPL table is not valid
RSC = 555	Maximum initialized applications number reached
RSC = 444	Interface error
RSC = 333	Application already initialized
RTC = 6	Connect:Express notification error
RSC = 008	Wait event error
RTC = 8	Termination error
RSC = 222	Requested application not connected
RSC = 111	Application not found in T1B2PAPL table

When an error occurs, you must stop the process and correct the parameters or the program.

APLRJNL Field

Connect:Express returns information to the user into the APLRJNL field. This field is mapped by the D1B2PJNL structure in the *MACLIB*. Normally, this information is displayed as a journal record. You can use this information to tell the application that Connect:Express is initializing or terminating.

When the user application is started before Connect:Express, Connect:Express notifies the application when it is in active status by sending a message in the Z45FILEN field (D1B2PJNL). This message contains the last journal record number in the first four bytes and the keyword “INIT” in the last four bytes of data.

When Connect:Express is stopped before the user application, the application is notified by a message in the Z45FILEN field (D1B2PJNL). The message contains the last journal record number in the first four bytes and the keyword “TERM” in the last four bytes of data.

L4B2ZAP0: Initialize the ESTAE Environment for L0B2ZAPL

The L4B2ZAP0 module initializes the ESTAE environment to disconnect the user application when a program ABENDs. You must write a CALL to the L4B2ZAP0 module before the first call to L0B2ZAPL with the 'I' function. The following example shows a call to L4B2ZAP0 with the same communication area as that used for L0B2ZAPL. Only APLTOMS and APLNAME fields are used by this module.

```
CALL L4B2ZAP0 USING REQUEST-L0B2ZAPL.

01  REQUEST-L0B2ZAPL.

 05 APLTOMS  PICTURE X(4)    VALUE 'TOMP'. ----- TO ---> L0B2ZAPL
 05 APLNAME  PICTURE X(8)    VALUE 'BSIT-IMS'. --------> L0B2ZAPL
```

Chapter 3

RJE Interface

This chapter describes the RJE interface. An RJE partner is a computer connected to the central site via BSC lines such as 2780/3780, or SDLC lines such as 3370. The Connect:Express RJE option enables this partner to send and receive files under the control of Connect:Express.

Overview

The Connect:Express OS/390 RJE interface consists of two programs called P1B2R002 and P1B2R003. P1B2R002 is for transmitting files, while P1B2R003 is for receiving files. An RJE partner is defined in the Connect:Express partner directory with the RJE ("J") partner type. Data is transferred through the JES2 SPOOL according to the data presentation rules described in *Connect:Express RJE Formats* on page 3-2.

Features

The RJE option provides security, standardization, and automation features.

You can secure RJE transfers with the Connect:Express RJE interface because transfers to and from RJE sites are recorded in the Connect:Express Partners and Files Directories. You can also implement SMF recording and RACF checking to enhance security.

The interface also enables you to standardize RJE transfers because Connect:Express controls the transfers with Partners and Files directories definitions, and gives the physical name (DSNAME) of the files transmitted from RJE sites based on the file directory definition. The RJE user can ignore the file characteristics, except the LRECL of the file managed by the monitor. RJE transfers are recorded in the Connect:Express Journal file, and messages are displayed in the Connect:Express SYSLOG file.

You can automate RJE transfers using Connect:Express automation processes such as start commands at the beginning or at the end of transfer.

Installing the RJE Option

The RJE option modules are provided on the standard product tape. To enable this option, it must be included in your asset protection key.

Customizing the RJE Interface

Complete the following steps to customize the RJE interface.

1. Update the $RJE module provided in the Connect:Express *SAMPLIB*.
2. Create an RJE symbolic Partner in the Connect:Express Partners Directory.
3. Create a symbolic File in the Connect:Express Files Directory.
4. Transmit or receive a file.

Connect:Express RJE Formats

Connect:Express uses an 80-character record as the transfer unit for an RJE file transfer. There are three types of records:

- Parameter record. This card identifies the partner and the file to transfer.
- Segmented data. Data is extracted from the file and records are 80 characters long.
- Logical end of file record.

The data flow can be made up of one or more file transfers, each one having its own parameter record, data records, and end of file record.

Parameter Record

The parameter record is placed before the file. The fields are identified in the following table.

FIELD	LENGTH	VALUE
Direction	1 byte	A: Transmit R: Receive
Reserved	3 bytes	Spaces
Record length	4 bytes	0080 (mandatory)
Reserved	4 bytes	Spaces
Symbolic file name	8 bytes	Must be defined in the files directory
Reserved	1 byte	Spaces
Record format	1 byte	F: Fixed
Reserved	2 bytes	Spaces
Original record length	5 bytes	LRECL: nnnnn
Reserved	1 byte	Space
Original block length	5 bytes	Same as LRECL
Reserved	1 byte	Space
Symbolic partner name	8 bytes	Must be defined in the partners directory
Reserved	1 octet	Space
Password	8 bytes	Password controlled by Connect:Express
Reserved	1 byte	Space
Transfer date	6 bytes	YYMMDD or omitted

FIELD	LENGTH	VALUE
Reserved	1 byte	Space
Transfer time	6 byte	HHMMSS or omitted
Reserved	13 bytes	Reserved

Segmented Data

Segmented data is extracted from the file. Each record has 80 characters.

FIELD	LENGTH	VALUE
Data	80 bytes	Segment record padded with blank for the last block if lower than 80 characters

Because the transmit unit is 80 characters, the transmitted file is segmented and recomposed in the target file. JES2 assumes transparency of data.

End-of-File Record

This record is placed at the end of segmented records. The fields are identified below.

FIELD	LENGTH	VALUE
E.O.F	3 bytes	OKF logical end of file
Reserved	77 bytes	Spaces

Using the RJE Interface

File transfer with the RJE interface is executed by batch jobs that report to Connect:Express through its subsystem interface. Files are exchanged through the JES2 spool.

File Transmission With P1B2R002

The P1B2R002 program transmits the files from the mainframe where Connect:Express is installed, to an RJE partner. This process allows the files to be transmitted in 80-character messages. The result is sent in the JES2 spool. The steps below describe the process of transmitting a file to an RJE partner.

1. Connect:Express receives a transfer request for a partner defined as an RJE partner type and validates it.
2. Connect:Express submits a JOB (Connect:Express -RJE). This JOB runs P1B2R002 which refers to the file to be transmitted and the SYSOUT FILE with the proper DEST value from the LUNAME field in the partner directory definition.
3. P1B2R002 carries out the data presentation of the selected file and writes the result in the JES2 SPOOL.
4. P1B2R002 records the transfer elements in the SYSLOG and SYSJNL of Connect:Express.

5. JES2 ensures the transmission of the SYSOUT to the connected partner. If there is no session available with the DEST partner, JES2 keeps the SYSOUT file until the next SIGN ON command is issued by this partner.

File Reception With P1B2R003

The P1B2R003 program receives files from a partner. First, it queries JES2, then Connect:Express, and formats the received file into the available file allocated by Connect:Express.

The steps below describe the process when a file is received from a partner.

1. The SIGN ON command connects the RJE equipment to the central site.
2. The user submits a flow of cards (JCL and data) to JES2, including the following:
   - One JOB card
   - One EXEC card for P1B2R003
   - Data in card format (parameter record, data, end of file record)
3. When running, the P1B2R003 program asks JES2 for its job number.
4. P1B2R003 sends a request to Connect:Express with the symbolic partner name, password, and symbolic file name. These elements are taken from the pilot record.
5. If Connect:Express is not running, P1B2R003 asks JES2 to be re-queued in the HOLD queue. The request is recorded in the SYSRCY file by the subsystem function if Connect:Express has been started since the last IPL. At the next hot start, Connect:Express releases the JOB by a command to JES2: $AJnnnn
6. If Connect:Express is running, it checks the validity of the request by verifying the directory information about this partner and file.
7. If the request is accepted, Connect:Express dynamically allocates the file and returns the data set name to P1B2R003. P1B2R003 allocates the target file, reads the card data flow, writes the records in proper format, and reports the transfer to the SYSREPRT FILE.
8. If the request is not accepted, Connect:Express returns the return code and reason code to P1B2R003. Errors are reported in the SYSREPRT file. At the end of execution, the SYSOUTs are returned to the RJE partner by JES2.

Using the Reception Utility P1B2R003

P1B2R003 is executed indirectly. The P1B2P500 module is executed with P1B2R003 as its first parameter. This ensures that the report is sent to the Connect:Express monitor if there is an ABEND of P1B2R003. (For example, Abend S122 or S37.) You can use P1B2R003 in one of the following ways.

- The RJE partner identifies himself with his symbolic partner name and provides the symbolic file name of a file containing a parameter card as the first card.
- The RJE partner adds a DD SYSPARM card to the JCL referring to a parameter card model.

EXEC PARM Field Structure

The EXEC PARM field is made up of five positional parameters.

1. Program name: P1B2R003
2. Subsystem name
3. Symbolic partner name or keyword '$'
4. Password or keyword '$'
5. Symbolic file name or keyword '$'

Note: The last three parameters must equal '$' when the parameter is provided at the beginning of the file.

End of Transfer Reporting

When the transfer has been completed, P1B2R003 reports details of the results to Connect:Express. If Connect:Express is not active, the subsystem interface directs the information to a routine that records it in the SYSRCY file.

Connect:Express executes the command for this file if the transfer was successful. The transfer information is then recorded in the SYSJNL file and SYSLOG file. If a request is not completed correctly, the request is set to Disabled (H). To start the request again, you must send the Enabled (E) command to Connect:Express.

The following screens provide examples of P1B2R003. The first example shows an EXEC PARM with no SYSPARM.

```
//...      JOB      ...
//S1       EXEC     PGM=P1B2P500,REGION=512K,
//         PARM='P1B2R003,TOM1,$,$,$'
//*
//*        SEND A FILE OF 3 RECORDS WITH LRECL = 160
//*
//STEPLIB  DD   DISP=SHR,DSN=XXXX.TOMV410.LOADLIB
//SYSUDUMP DD   SYSOUT=*
//SYSREPRT DD   DUMMY,DCB=BLKSIZE=133
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
A   0080    FILE0001 F  00160 00160 TEDRJE   PASSWORD
11111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111
22222222222222222222222222222222222222222222222222222222222222222222222222222222
22222222222222222222222222222222222222222222222222222222222222222222222222222222
33333333333333333333333333333333333333333333333333333333333333333333333333333333
33333333333333333333333333333333333333333333333333333333333333333333333333333333
OKF
/*
```

The example below shows P1B2R003 with a SYSPARM.

```
//...      JOB      ...
//S1       EXEC     PGM=P1B2P500,REGION=2048K,
//         PARM='P1B2R003,TOM1,TEDRJE,PASSWORD,FILE0001'
//*
//*        SEND A FILE OF 2 RECORDS WITH LRECL = 240
//*
//STEPLIB  DD   DISP=SHR,DSN=XXXX.TOMV222.LOADLIB
//SYSUDUMP DD   SYSOUT=*
//SYSREPRT DD   DUMMY,DCB=BLKSIZE=133
//SYSPRINT DD   SYSOUT=*
//SYSPARM  DD   SYSOUT=*
A   0080    xxxxxxxx F  00200 00200
//SYSIN    DD   *
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
OKF
/*
```

Error Messages

The table below lists possible error codes for the P1B2R003 program.

Code	Description
P1B2R0031I ERROR (xxxxxxxx) rc-0-000/ REQ:rrrrrrrr	Error during execution of P1B2R003
RC=03	SYSPARM open error
RC=05	SYSPARM logic error
RC=04	No valid parameter in SYSPARM file
RC=06	SYSPRT, SYSREPRT, SYSIN open error
RC=07	Memory allocation error
RC=08	Parameter card from SYSIN is invalid
RC=09	Length invalid in parameter card (not equal to 0080)
RC=10	DDNAME invalid in parameter card
RC=11	RECFM invalid in parameter card (not equal to F)
RC=12	Request error to Connect:Express
RC=14	Request rejected by Connect:Express
RC=15	Loading error of begin-transfer exit
RC=16	Loading error of end-of-transfer exit
RC=17	Begin-transfer exit error
RC=18	Dynamic allocation of received file in error

Code	Description
RC=19	Received file open error
RC=20	RECFM of received file is incompatible
RC=21	LRECL of received file is incompatible
RC=22	LRECL of received file is lower than 80 bytes
RC=23	Logical write error on received file
RC=24	Write error on received file
RC=25	Close error
RC=26	Memory management error
RC=17	End transfer exit error
RC=28	De-allocation of received file error
RC=29	Report end-of-transfer to Connect:Express error
RC=97	Job number not found
RC=98	IEFSSREQ macro error
P1B2R0032I TOMX INACTIVE ->	JOB NOT EXECUTED RC:..
P1B2R0033I TOMX INACTIVE ->	JOB NOT EXECUTED RC:..
P1B2R0034I TOMX INACTIVE ->	JOB NOT EXECUTED RC:..

Return Codes

The following table lists the possible return codes.

Code	Description
0	Request accepted and transfer normally ended
4	Request rejected
8	Request accepted and abnormal end-of-transfer
12	Parameter or run error

Index