

CICS® Transaction Server for VSE/ESA™



External CICS Interface

Release 1

CICS® Transaction Server for VSE/ESA™



External CICS Interface

Release 1

Note!

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 93.

First Edition (June 1999)

This edition applies to Release 1 of CICS Transaction Server for VSE/ESA, program number 5648-054, and to all subsequent versions, releases, and modifications until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

Order publications through your IBM representative or the IBM branch office serving your locality.

At the back of this publication is a page entitled "Sending your comments to IBM". If you want to make any comments, please use one of the methods described there.

© Copyright International Business Machines Corporation 1999. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Preface	v
What this book is about	v
Who this book is for	v
What you need to know to understand this book	v
How to use this book	v
Notes on terminology	vi
Chapter 1. Introduction	1
Overview	1
Benefits of the external CICS interface	5
Requirements for the external CICS interface	6
Chapter 2. The EXCI CALL interface	7
Initialize_User	8
Allocate_Pipe	11
Open_Pipe	13
DPL_Request	15
Close_Pipe	22
Deallocate_Pipe	24
Chapter 3. The EXEC CICS interface	31
EXEC CICS LINK command	31
Chapter 4. Defining connections to CICS	39
CONNECTION resource definition	39
SESSIONS resource definitions for EXCI connections	41
Inquiring on the state of EXCI connections	43
Chapter 5. The EXCI user-replaceable module	45
Chapter 6. External CICS interface options table, DFHXCOPT	47
Chapter 7. Compiling and link-editing external CICS interface client programs	49
The external CICS interface stub, DFHXCSTB	49
Language considerations	49
Sample application programs	50
Job control language to run an EXCI client program	55
Chapter 8. Security	57
MRO logon and bind-time security	57
Link security	58
User security	58
Surrogate user checking	59
Chapter 9. Problem determination	61
Trace	61
System dumps	61
Abends from related CICS programs	62
04xx (external CICS interface) abend codes	62

The EXCI service trap, DFHXCTRA	65
EXCI trace entry points	65
Chapter 10. Response and reason codes returned on EXCI calls	73
Reason code for response:	73
Reason codes for response:	73
Reason codes for response:	74
Reason codes for response:	76
Reason codes for response:	79
Chapter 11. Messages and Codes	85
Bibliography	87
Books from VSE/ESA 2.4 base program libraries	88
Books from VSE/ESA 2.4 optional program libraries	90
Notices	93
Trademarks and service marks	94
Index	95

Preface

What this book is about

This book is about the external CICS® interface—a programming interface that allows a VSE application program running in a VSE/ESA™ partition to link to a CICS application program that executes in a CICS region.

It contains reference information to help you write a VSE client program, and tells you how to set up and run the client program. It also tells you what you need to know to set up the connection between a CICS region and an EXCI client program.

Who this book is for

This book is for application programmers who want to write a VSE application program that uses the external CICS interface to link to a CICS application program.

What you need to know to understand this book

The book assumes that you are an application programmer experienced in at least one of the supported languages, which are assembler, COBOL, PL/I, and C.

How to use this book

This book is intended to be used as a reference book. The various chapters cover topics such as the two forms of the external CICS interface, how to set up and run a client program in a VSE partition.

Notes on terminology

The terms listed in Table 1 are commonly used in the CICS Transaction Server for VSE/ESA Release 1 library. See the *CICS Glossary* for a comprehensive definition of terminology.

<i>Table 1 (Page 1 of 2). Commonly used words and abbreviations in CICS Transaction Server for VSE/ESA Release 1</i>	
Term	Definition (and abbreviation if appropriate)
\$(the dollar symbol)	In the character sets and programming examples given in this book, the dollar symbol (\$) is used as a national currency symbol and is assumed to be assigned the EBCDIC code point X'5B'. In some countries a different currency symbol, for example the pound symbol (£), or the yen symbol (¥), is assigned the same EBCDIC code point. In these countries, the appropriate currency symbol should be used instead of the dollar symbol.
BSM	BSM is used to indicate the basic security management supplied as part of the VSE/ESA product. It is RACROUTE-compliant, and provides the following functions: <ul style="list-style-type: none"> • Signon security • Transaction attach security
C	The C programming language
CICSplex	A CICSplex consists of two or more regions that are linked using CICS intercommunication facilities. Typically, a CICSplex has at least one terminal-owning region (TOR), more than one application-owning region (AOR), and may have one or more regions that own the resources accessed by the AORs
CICS Data Management Facility	The new CICS Transaction Server for VSE/ESA Release 1 facility to which all statistics and monitoring data is written, generally referred to as "DMF"
CICS/VSE	The CICS product running under the VSE/ESA operating system, frequently referred to as simply "CICS"
COBOL	The COBOL programming language
DB2® for VSE/ESA	DATABASE 2™ for VSE/ESA which was previously known as "SQL/DS".

Table 1 (Page 2 of 2). Commonly used words and abbreviations in CICS Transaction Server for VSE/ESA Release 1

Term	Definition (and abbreviation if appropriate)
ESM	<p>ESM is used to indicate a RACROUTE-compliant external security manager that supports some or all of the following functions:</p> <ul style="list-style-type: none"> • Signon security • Transaction attach security • Resource security • Command security • Non-terminal security • Surrogate user security • MRO/ISC security (MRO, LU6.1 or LU6.2) • FEPI security.
FOR (file-owning region)—also known as a DOR (data-owning region)	A CICS region whose primary purpose is to manage VSAM and DAM files, and VSAM data tables, through function provided by the CICS file control program.
IBM® C for VSE/ESA	The Language Environment® version of the C programming language compiler. Generally referred to as “C/VSE”.
IBM COBOL for VSE/ESA	The Language Environment version of the COBOL programming language compiler. Generally referred to as “COBOL/VSE”.
IBM PL/I for VSE/ESA	The Language Environment version of the PL/I programming language compiler. Generally referred to as “PL/I VSE”.
IBM Language Environment for VSE/ESA	The common runtime interface for all LE-conforming languages. Generally referred to as “LE/VSE”.
PL/I	The PL/I programming language
VSE/POWER	Priority Output Writers Execution processors and input Readers. The VSE/ESA spooling subsystem which is exploited by the report controller.
VSE/ESA System Authorization Facility	The new VSE facility which enables the new security mechanisms in CICS TS for VSE/ESA R1, generally referred to as “SAF”
VSE/ESA Central Functions component	The new name for the VSE Advanced Function (AF) component
VSE/VTAM	“VTAM®”

Chapter 1. Introduction

This chapter gives a brief overview of the external CICS® interface (EXCI), covering the following topics:

- Overview
- Benefits of the external CICS interface
- Requirements for the external CICS interface

Overview

The external CICS interface is an application programming interface that enables a non-CICS program (a client program) running in VSE to call a program (a server program) running in a CICS region and to pass and receive data by means of a communications area. The CICS application program is invoked as if linked-to by another CICS application program.

This programming interface allows a user to allocate and open sessions (or pipes) to a CICS region, and to pass distributed program link (DPL) requests over them. The multiregion operation (MRO) interregion communication facility supports these requests, and each pipe¹ maps onto one MRO session, with a limit of 25 pipes per EXCI partition.

The client program and the CICS server region (the region where the server program runs or is defined) must reside in the same VSE image.

A client program that uses the external CICS interface can operate multiple sessions for different users (either under the same or separate VSE tasks) all coexisting in the same VSE partition without knowledge of, or interference from, each other.

Where a client program attaches another client program, the attached program runs under its own VSE task.

The programming interfaces

The external CICS interface provides two forms of programming interface: the EXCI CALL interface and the EXEC CICS interface.

The EXCI CALL interface: This interface consists of six commands that allow you to:

- Allocate and open sessions (pipes) to a CICS system from non-CICS programs running under VSE
- Issue DPL requests on these sessions from the non-CICS programs
- Close and deallocate the sessions on completion of the DPL requests.

¹ pipe. A single communication path between a sending process and a receiving process. In an external CICS interface implementation, each pipe maps onto one MRO session, where the client program represents the sending process and the CICS server region represents the receiving process.

The six EXCI commands are:

- Initialise_User
- Allocate_Pipe
- Open_Pipe
- DPL call
- Close_Pipe
- Deallocate_Pipe

The EXEC CICS interface: The external CICS interface provides a single, composite command—EXEC CICS LINK PROGRAM—that performs all six commands of the EXCI CALL interface in one invocation.

This command takes the same form as the distributed program link command of the CICS command-level application programming interface.

API restrictions for server programs

A CICS server program invoked by an external CICS interface request is restricted to the DPL subset of the CICS application programming interface. This subset (the DPL subset) of the API commands is the same as for a CICS-to-CICS server program.

See the *CICS Application Programming Guide* for details of the DPL subset for server programs.

Choosing between the EXEC CICS and the CALL interface

As illustrated in the various language versions of the CICS-supplied sample client program (see “Sample application programs” on page 50 for details), you can use both the CALL interface (all six commands) and the EXEC CICS LINK command in the same program, to perform separate requests. As a general rule, it is unlikely that you would want to do this in a production program.

Each form of the external CICS interface has its particular benefits.

- For low-frequency or single DPL requests, you are recommended to use the EXEC CICS LINK command.

It is easier to code, and therefore less prone to programming errors.

Note that each invocation of an EXEC CICS LINK command causes the external CICS interface to perform all the functions of the CALL interface, which results in unnecessary overhead.

- For multiple or frequent DPL requests from the same client program, you are recommended to use the EXCI CALL interface.

This is more efficient, because you need only perform the Initialize_User and Allocate_Pipe commands once, at or near the beginning of your program, and the Deallocate_Pipe once on completion of all DPL activity. In between these functions, you can open and close the pipe as necessary, and while the pipe is opened, you can issue as many DPL calls as you want.

Illustrations of the external CICS CALL interface

The diagrams in Figure 1 through Figure 4 on page 4 illustrate the external CICS interface using the EXCI CALL interface.

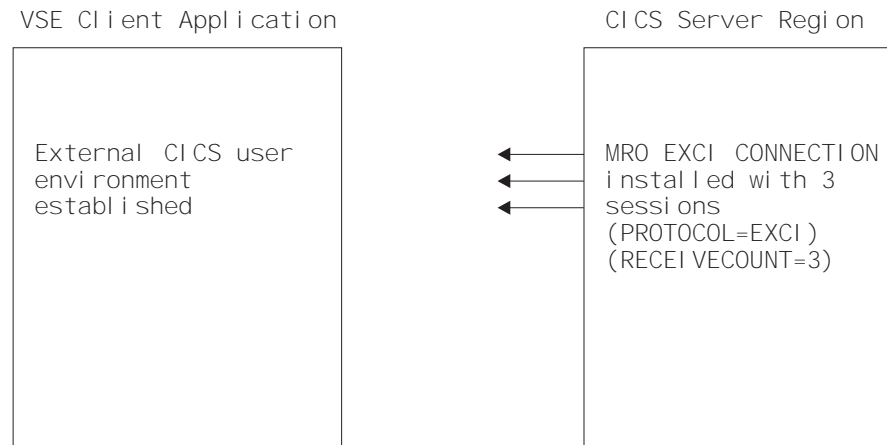


Figure 1. Stage 1: Status after an INITIALIZE_USER call

Notes to Figure 1:

1. In this diagram, the target CICS server region is running with IRC open, and one EXCI connection with three sessions installed, at the time the client application program issues an INITIALIZE_USER call.
2. The client application program partition is initialized with the EXCI user environment. There is no MRO activity at this stage, and no pipe exists.

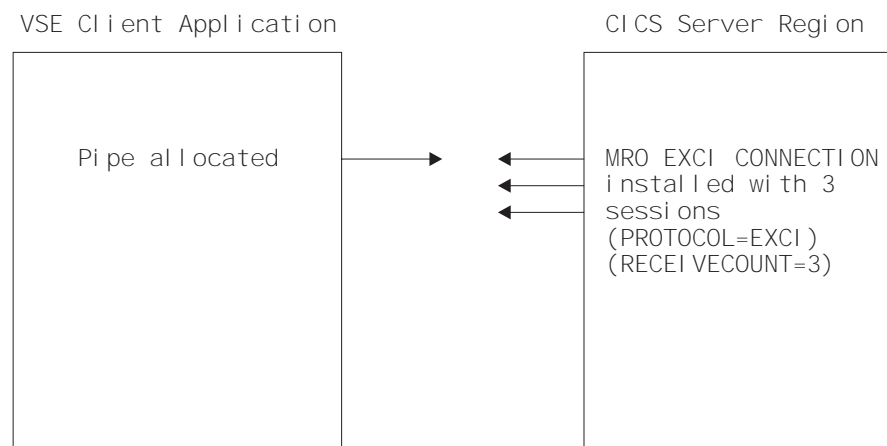


Figure 2. Stage 2: Status after the first ALLOCATE_PIPE call

Note to Figure 2: The external CICS interface logs on to MRO, identifying the target CICS server region.

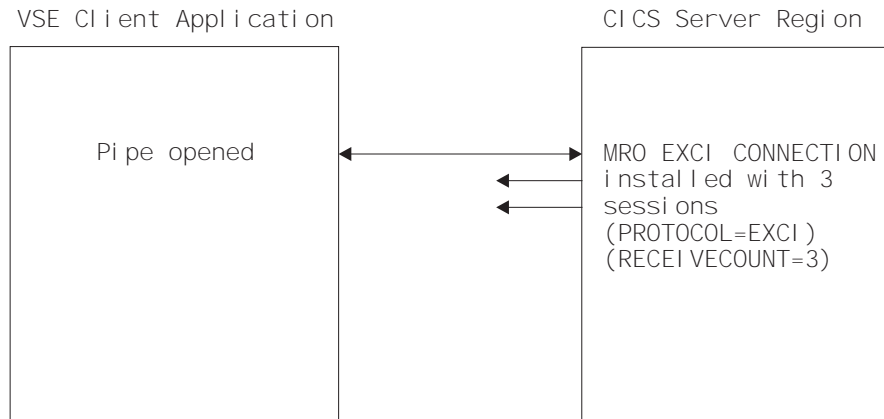


Figure 3. Stage 3: Status after the OPEN_PIPE call

Notes to Figure 3:

1. The external CICS interface connects to the CICS server region, and the pipe is now available for use.
2. The remaining two EXCI sessions are free, and can be used by further open pipe requests from the same, or a different, client application program (provided the connection is generic).

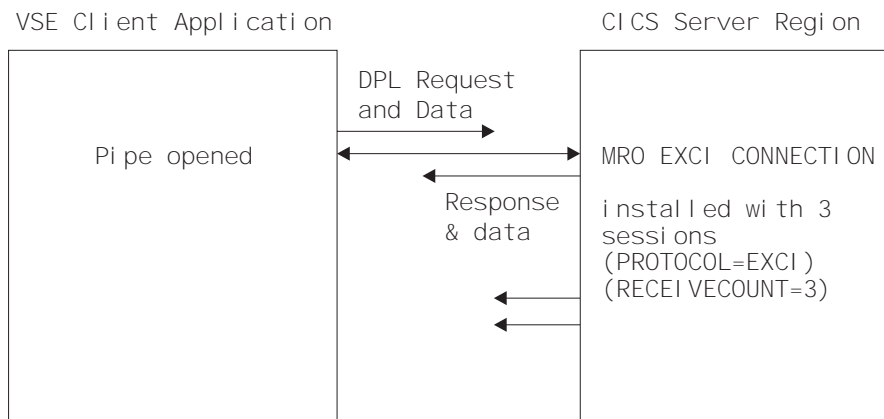


Figure 4. Stage 4: Status with one open pipe, processing a DPL call

Note to Figure 4: The external CICS interface passes the DPL request over the open pipe, with any associated data. The CICS server region returns a response and data over the open pipe.

Closing pipes: When the client application program closes a pipe, it remains allocated ready for use by the same user, and the status is as shown in Figure 2 on page 3. At this stage, the MRO session is available for use by another open pipe request, from the same or from a different client application program (provided the connection is generic).

Deallocating pipes: When the client application program deallocates a pipe, it logs off from MRO and frees all the storage associated with the session. This leaves the status as shown in Figure 1 on page 3.

Illustration of the EXCI EXEC CICS interface

Figure 5 illustrates the EXEC CICS interface, and how it resolves to the 6 EXCI CALLS.

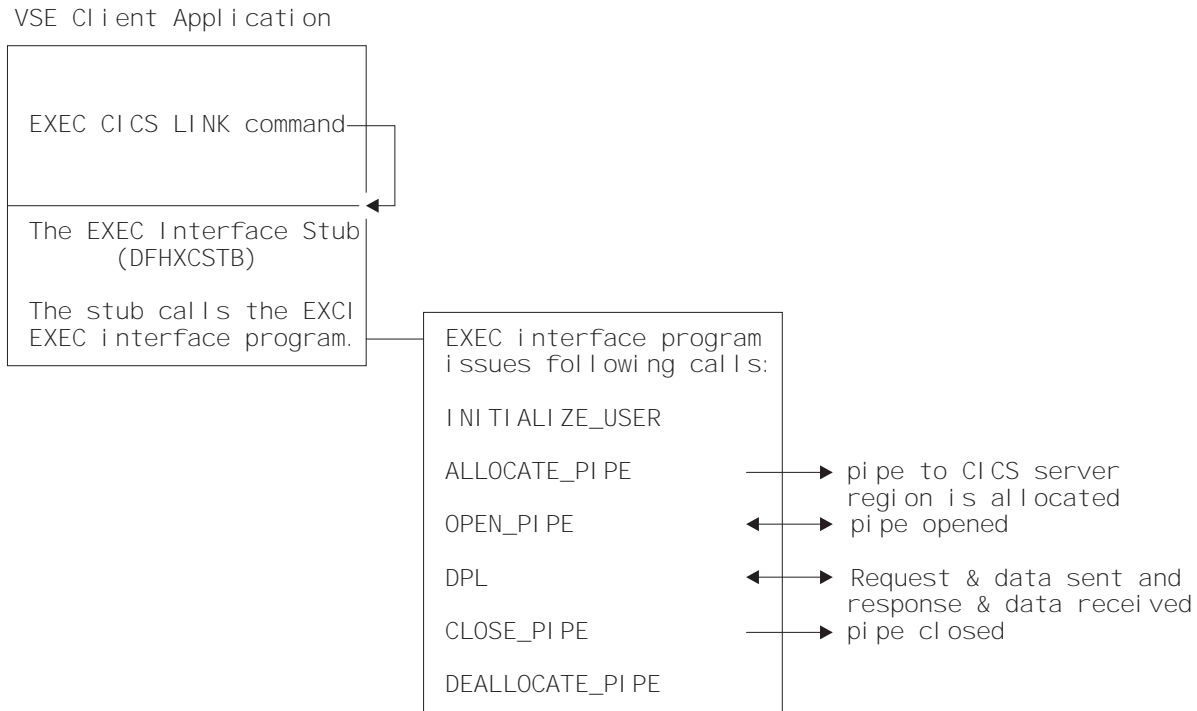


Figure 5. Illustration of the external CICS interface using the EXEC CICS command

Benefits of the external CICS interface

The external CICS interface makes CICS applications more easily accessible from non-CICS environments.

Programs running in VSE/ESA can issue an EXEC CICS LINK PROGRAM command, or use the EXCI CALL interface, to call a CICS application programs running in a CICS server region.

The provision of this programming interface means that, for example, VSE programs can:

- Update resources with integrity while CICS is accessing them.
- Take CICS resources offline, and back online, at the start and end of a VSE job. For example, you can:
 - Open and close CICS files.
 - Enable and disable transactions in CICS (and so eliminate the need for a master terminal operator during system backup and recovery procedures).

Requirements for the external CICS interface

Client programs running in a VSE partition can communicate only with CICS server partitions running under CICS Transaction Server for VSE/ESA™, Release 1, or later. This is because you need a level of the MRO connection definition that is enhanced to support the external CICS interface.

Also, the client program can connect to the server CICS region only through the CICS Transaction Server for VSE/ESA Release 1, or later, interregion communication program, DFHIRP.

For information about DFHIRP, and its requirement to be installed in the VSE shared virtual area (SVA), see the *CICS System Definition Guide*.

Chapter 2. The EXCI CALL interface

The EXCI CALL interface consists of six commands that allow you to:

- Allocate and open sessions to a CICS region from non-CICS programs running under VSE
- To issue distributed program link (DPL) requests on these sessions from the non-CICS programs.

The six EXCI commands are:

- Initialize_User
- Allocate_Pipe
- Open_Pipe
- DPL_Request
- Close_Pipe
- Deallocate_Pipe

The application program stub, DFHXCSTB

The EXCI commands invoke the external CICS interface via an application programming stub provided by CICS, called DFHXCSTB. You must include this stub when you link-edit your non-CICS client program.

The CALL interface commands

In the description of each command that follows, the syntax box illustrates the assembler form of the command. The commands are also supported by the C for VSE, COBOL for VSE, and PL/I for VSE programming languages, using the CALL conventions appropriate for the language.

There are examples of these CALLs, in all the supported languages, in the sample program client programs provided by CICS. See “Sample application programs” on page 50 for information about these.

Initialize_User

Initialize the user environment, including obtaining authority to use IRC facilities. The environment is created for the lifetime of the VSE task, so the command needs to be issued only once per user per VSE task. Further commands from this user must be issued under the same VSE task.

Note: A *user* is a program that has issued an Initialize_user request (or for which an Initialize_user request has been issued), with a unique name per VSE task. For example:

- A simple client program running under VSE can be a single user of the external CICS interface.
- A client program running under VSE can open several pipes and issue external CICS interface calls over them sequentially, on behalf of different vendor packages. In this case, from the viewpoint of the client program, each of the packages is a user, identified by a unique user name. Thus a single client program can operate on behalf of multiple users.
- A program running under VSE can attach several VSE subtasks, under each of which a vendor package issues external CICS interface calls on its own behalf. Each package is a client program in its own right, and runs under its own VSE subtask. Each is also a user, with a unique user name.

```
▶▶—CALL DFHXCIS,(version_number,—return_area,—user_token,—call_type,——————▶▶
▶—user_name),VL,MF=(E,(1))▶▶
```

Initialize_User command parameters

version_number

A full-word binary input area indicating the version of the external CICS interface parameter list being used. It must be set to 1 in the client program.

The equated value for this parameter in the CICS-supplied copybook DFHXCPLx (where *x* indicates the language) is VERSION_1. See page 27 for copybook details.

return_area

A 5-word output area to receive response and reason codes, and a message pointer field. For more details see “Return area for the EXCI CALL interface” on page 27.

user_token

A 1-word output area containing a 32-bit token supplied by the CICS external interface to represent the client program.

The user token corresponds to the *user-name* parameter. The client program must pass this token on all subsequent external CICS interface commands made for the user defined on the *user_name* parameter.

call_type

A 1-word input area indicating the function of the command. It must be set to 1 in the client program to indicate that this is an Initialize_User command.

The equated value for this call in the CICS-supplied copybook DFHCPLx (where x indicates the language) is INIT_USER. See page 27 for copybook details.

user_name

An input area holding a name that identifies the user of the external CICS interface. Generally, this is the client program. If this user is to use a *specific pipe*, the value in *user_name* must match that of the NETNAME attribute of the CONNECTION definition for the specific pipe. (See Chapter 4, “Defining connections to CICS” on page 39 for information about specific and generic pipes.)

Response and reason codes on the Initialize_User call

For all non-zero response codes, a unique reason code value identifies the reason for the response.

Note: All numeric response and reason code values are in decimal.

The following is a summary of the response and reason codes that the external CICS interface can return on the Initialize_User call:

Response OK:

0 Normal response

Response WARNING:

3 VERIFY_BLOCK_FM_ERROR

4 WS_FREEMAIN_ERROR

Response RETRYABLE:

201 NO_CICS_IRC_STARTED

Response USER_ERROR:

401 INVALID_CALL_TYPE

402 INVALID_VERSION_NUMBER

403 INVALID_USER_NAME

410 DFHMEBM_LOAD_FAILED

411 DFHMET4E_LOAD_FAILED

412 DFHXCURM_LOAD_FAILED

413 DFHXCTRA_LOAD_FAILED

419 CICS_AFCB_PRESENT

420 DFHXCOPT_LOAD_FAILED

Response SYSTEM_ERROR:

601 WS_GETMAIN_ERROR

602 XCGLOBAL_GETMAIN_ERROR

603 XCUSER_GETMAIN_ERROR

605 VERIFY_BLOCK_GM_ERROR

606 SSI_VERIFY_FAILED

607 CICS_SVC_CALL_FAILURE

622 ESTAEX_SETUP_FAILURE

623 ESTAEX_INVOKED

627 INCORRECT_SVC_LEVEL

Initialize_User

For more information about response codes, see “Response code values” on page 26.

For information about the reason codes, see Chapter 10, “Response and reason codes returned on EXCI calls” on page 73.

Allocate_Pipe

Allocate a single session, or pipe, to a CICS region. This command does not connect the client program to a CICS region; this happens on the Open_Pipe command. You can allocate up to 25 pipes in an EXCI partition.

```
▶—CALL DFHCIS, (version_number, —return_area, —user_token, —call_type, —pipe_token, —▶
▶—CICS_applid, —allocate_opts), VL, MF=(E, (1))—▶◀
└─null_ptr, ─┘
```

Allocate_Pipe command parameters

version_number

A full-word binary input area indicating the version of the external CICS interface parameter list being used. It must be set to 1 in the client program.

The equated value for this parameter in the CICS-supplied copybook DFHCPLx (where x indicates the language) is VERSION_1. See page 27 for copybook details.

return_area

A 5-word output area to receive response and reason codes, and a message pointer field. For more details see “Return area for the EXCI CALL interface” on page 27.

user_token

The 1-word token returned on the Initialize_User command.

call_type

A 1-word input area indicating the function of the command. It must be set to 2 in the client program to indicate that this is an Allocate_Pipe command.

The equated value for this call in the CICS-supplied copybook DFHCPLx (where x indicates the language) is Allocate_Pipe. See page 27 for copybook details.

pipe_token

A 1-word output area. CICS returns a 32-bit token in this area to represent the allocated session. This token must be used on any subsequent command that uses this session.

CICS_applid

An 8-byte input area containing the generic applid of the CICS region to which the allocated session is to be connected.

Although an applid is required to complete the Allocate_Pipe function, this parameter is optional on the Allocate_Pipe call. You can either specify the applid on this parameter to the Allocate_Pipe call, or in the user-replaceable module, DFHCURM, using the URMAPPPL parameter (DFHCURM is always invoked during Allocate_Pipe processing). You can also use the URMAPPPL parameter in DFHCURM to override an applid specified on the Allocate_Pipe call. See Chapter 5, “The EXCI user-replaceable module” on page 45 for information about the URMAPPPL parameter.

If you omit the applid from the call, you must ensure that the CALL parameter list contains a null address for *CICS_applid*. How you do this depends on the language you are using for the non-CICS client program. For an example of a

call that omits an optional parameter, see “Example of EXCI CALLs with null parameters” on page 28.

allocate_opts

A 1-byte input area to represent options specified on this command. The options specify which type of session is to be used—specific or generic. X'00' represents a specific session. X'80' represents a generic session.

The equated value for these options in the CICS-supplied copybook DFHXCPLx (where x indicates the language) are SPECIFIC_PIPE and GENERIC_PIPE. See page 27 for copybook details.

Response and reason codes on the Allocate_Pipe call

For all non-zero response codes, a unique reason code value identifies the reason for the response.

Note: All numeric response and reason code values are in decimal.

The following is a summary of the response and reason codes that the external CICS interface can return on the Allocate_Pipe call:

Response OK:

0 Normal response

Response USER_ERROR:

401 INVALID_CALL_TYPE
402 INVALID_VERSION_NUMBER
404 INVALID_USER_TOKEN

Response SYSTEM_ERROR:

604 XCPipe_GETMAIN_ERROR
608 IRC_LOGON_FAILURE
622 ESTAEX_SETUP_FAILURE
623 ESTAEX_INVOKED
628 IRP_LEVEL_CHECK_FAILURE

For information about response codes, see “Response code values” on page 26.

For information about the reason codes, see Chapter 10, “Response and reason codes returned on EXCI calls” on page 73.

Open_Pipe

Cause IRC to connect an allocated pipe to a receive session of the appropriate connection defined in the CICS region named either on the Allocate_Pipe command, or in DFHXCURM. The appropriate connection is either:

- The EXCI connection with a NETNAME value equal to the *user_name* parameter on the Initialize_User command (that is, you are using a specific connection, dedicated to this client program),

or

- The EXCI connection defined as generic.

Note: This command should be used only when there is a DPL call ready to be issued to the CICS region. When not in use, EXCI sessions should not be left open.

If sessions are left open, CICS may not be able to shut its IRC facility in an orderly manner. A normal shutdown of CICS waits if any EXCI sessions are not closed. CICS issues message DFHIR2321 indicating the following information:

- The netname of the session if it is on a specific connection
- The word GENERIC if the open sessions are on a generic connection.

Provided that at least one DPL_Request call has been issued on the session, message DFHIR2321 also shows the job name, EXEC name, and partition-id of the client job that is using the session, and the ID of the VSE image on which the client program is running.

```
▶▶—CALL DFHCIS, (version_number, —return_area, —user_token, —call_type, —————▶▶
▶▶—pipe_token), VL, MF=(E, (1))—▶▶
```

Open_Pipe command parameters

version_number

A full-word binary input area indicating the version of the external CICS interface parameter list being used. It must be set to 1 in the client program.

The equated value for this parameter in the CICS-supplied copybook DFHXCPLx (where *x* indicates the language) is VERSION_1. See page 27 for copybook details.

return_area

A 5-word output area to receive response and reason codes, and a message pointer field. For more details, see “Return area for the EXCI CALL interface” on page 27.

user_token

The 1-word token returned on the Initialize_User command.

call_type

A 1-word input area indicating the function of the command. This must be set to 3 in the client program to indicate that this is an Open_pipe command.

The equated value for this call in the CICS-supplied copybook DFHXCPLx (where *x* indicates the language) is Open_Pipe. See page 27 for copybook details.

pipe_token

A 1-word output area containing the token passed by CICS on the Allocate_Pipe command. It represents the pipe being opened on this command.

Response and reason codes on the Open_Pipe call

For all non-zero response codes, a unique reason code value identifies the reason for the response.

Note: All numeric response and reason code values are in decimal.

The following is a summary of the response and reason codes that the external CICS interface can return on the Open_Pipe call:

Response OK:

0 NORMAL

Response WARNING:

1 PIPE_ALREADY_OPEN

Response RETRYABLE:

202 NO_PIPE

203 NO_CICS

Response USER_ERROR:

401 INVALID_CALL_TYPE

402 INVALID_VERSION_NUMBER

404 INVALID_USER_TOKEN

418 INVALID_PIPE_TOKEN

Response SYSTEM_ERROR:

609 IRC_CONNECT_FAILURE

621 PIPE_RECOVERY_FAILURE

622 ESTAEX_SETUP_FAILURE

623 ESTAEX_INVOKED

For information about response codes, see "Response code values" on page 26.

For information about the reason codes, see Chapter 10, "Response and reason codes returned on EXCI calls" on page 73.

DPL_Request

Issue a distributed program link request across an open pipe connected to the CICS region on which the server (or target) application program resides. The command is synchronous, and the VSE task waits for a response from CICS. Once a pipe is opened, any number of DPL requests can be issued before the pipe is closed. To the server program, the link request appears just like a standard EXEC CICS LINK request from another CICS region, and it is not aware that it is sent from a non-CICS client program using EXCI.

```
▶▶CALL DFHCIS,(version_number,—return_area,—user_token,—call_type,—pipe_token,—▶▶
▶▶pgmname,—COMMAREA,—COMMAREA_len,—data_len,—transid,—uowid,—▶▶
▶▶—null_ptr,—null_ptr,—null_ptr,—▶▶
▶▶—userid,—dpl_retarea,—DPL_opts),VL,MF=(E,(1))▶▶
▶▶—null_ptr,—▶▶
```

DPL_Request command parameters

version_number

A full-word binary input area indicating the version of the external CICS interface parameter list being used. It must be set to 1 in the client program.

The equated value for this parameter in the CICS-supplied copybook DFHXCPLx (where x indicates the language) is VERSION_1. See page 27 for copybook details.

return_area

A 5-word output area to receive response and reason codes, and a message pointer field. For more details, see "Return area for the EXCI CALL interface" on page 27.

user_token

A 1-word input area specifying the user token returned to the client program on the Initialize_User command.

call_type

A 1-word input area indicating the function of the command. This must be set to 6 in the client program to indicate that the pipe is now being used for the DPL_Request call.

The equated value for this call in the CICS-supplied copybook DFHXCPLx (where x indicates the language) is DPL_Request. See page 27 for copybook details.

pipe_token

A 1-word input area specifying the token returned by EXCI on the Allocate_Pipe command. It represents the pipe being used for the DPL_Request call.

pgmname

The 8-character name of the CICS application program being called as the server program.

This is either the name as specified on a predefined PROGRAM resource definition installed in the CICS server region, or as it is known to a user-written autoinstall program if the program is to be autoinstalled. The program can be defined in the CICS server region as a local program, or it can be defined as

remote. Programs defined as remote enable “daisy-chaining”, where EXCI-CICS DPL calls become EXCI-CICS-CICS DPL calls.

COMMAREA

A variable length input area for the communications area (COMMAREA) between the client and server programs. The length is defined by *COMMAREA_len*.

This is the storage area that contains the data to be sent to the CICS application program. This area is also used to receive the updated COMMAREA from the CICS application program (the server program).

This parameter is optional. If it is not required, you must ensure that the CALL parameter list contains a null address for this parameter. How you do this depends on the language you are using for the non-CICS client program. For an example of a call that omits an optional parameter, see “Example of EXCI CALLs with null parameters” on page 28.

COMMAREA_len

A full-word binary input area. This parameter specifies the length of the COMMAREA. It is also the length of the server program’s COMMAREA (EIBCALEN).

If you specify a COMMAREA, you must also specify this parameter to define the length.

If you don’t specify a COMMAREA, this parameter is ignored.

data_len

A full-word binary input area. This parameter specifies the length of contiguous storage, from the start of the COMMAREA, to be sent to the server program.

This parameter restricts the amount of data sent to the server program, and should be used to optimize performance if, for example, the COMMAREA is large but the amount of data being passed is small.

Note that on return from the server program, the EXCI data transformer program ensures that the COMMAREA in the non-CICS client program is the same as that of the server program. This caters for the following conditions:

- The data returned is **more** than the data passed in the original COMMAREA.
- The data returned is **less** than the data passed in the original COMMAREA.
- There is **no data** returned because it is unchanged.
- The server is returning **null data**.

The value of *data_len* must not be greater than the value of *COMMAREA_len*. A value of zero is valid and results in no data being sent to the server program.

If you don’t specify a COMMAREA, this parameter is ignored.

transid

A 4-character input area containing the id of the CICS mirror transaction under which the server program is to run. This transaction must be defined to the CICS server region, and its definition **must observe the following rules**:

- It must *not* specify the server program as the initial program of the transaction

- It *must* specify the mirror program DFHMIRS, and the profile DFHCICSA.

Failure to specify DFHMIRS as the initial program means that a COMMAREA passed from the client application program is not passed to the CICS server program. Furthermore, the DPL request fails and the client application program receives a response of SYSTEM_ERROR and reason SERVER_PROTOCOL_ERROR.

When the CICS server region receives a DPL request, it attaches the mirror transaction and invokes DFHMIRS. The mirror program then passes control to the requested server program, passing the COMMAREA supplied by the client program. The COMMAREA passed to the server program is primed with the data only, the remainder of the COMMAREA being set to nulls.

The purpose of the *transid* parameter is to distinguish between different invocations of the server program. This enables you to run different invocations of the server program under transactions that specify different attributes. For example, you can vary the transaction priorities, or the security requirements.

A *transid* is optional. By default, the CICS server region uses the CICS-supplied mirror transaction, CSMI. If you don't want to specify *transid*, you must ensure that the CALL parameter list contains a null address for this parameter. How you do this depends on the language you are using for the non-CICS client program. For an example of a call that omits an optional parameter, see “Example of EXCI CALLs with null parameters” on page 28.

uowid

An input area containing the APPC unit-of-work identifier.

This parameter is optional. If you don't want to specify *uowid*, you must ensure that the CALL parameter list contains a null address for this parameter. How you do this depends on the language you are using for the non-CICS client program. For an example of a call that omits an optional parameter, see “Example of EXCI CALLs with null parameters” on page 28.

If specified, the *uowid* parameter is passed to the CICS server region, which uses it as the APPC UOWID for the first unit of work executed by the CICS server program. If the server program issues intermediate syncpoints before returning to the client program, CICS uses the supplied *uowid* for the subsequent units of work, but with the two byte sequence number incremented for each new logical unit of work. If the CICS server program updates remote resources, the client-supplied APPC UOWID is distributed to the remote systems that own the resources.

The *uowid* parameter is supplied on the EXCI CALL interface for correlation purposes only, to allow units of work that originated from a particular client program to be identified in CICS. The *uowid* is not provided for recovery purposes between CICS and the client program. No syncpoint coordination occurs between the client program and CICS, because all CICS server programs called from a client program run with SYNCONRETURN specified.

The *uowid* can be a maximum of 27 bytes long and has the following format:

- A 1-byte length field containing the overall length of the UOWID (excluding this field)
- A 1-byte length field containing the length of the logical unit name (excluding this field)

- A logical unit name field of variable length up to a maximum of 17 bytes.
To conform to APPC architecture rules, the LUNAME must be of the form AAAAAAAAA.BBBBBBBB, where AAAAAAAAA is optional and:
 - AAAAAAAAA andBBBBBBB are 1–8 byte names separated by a period
 - If AAAAAAAAA is omitted, the period must also be omitted
 - AAAAAAAAA andBBBBBBB must be type-1134 symbol strings (that is, character strings consisting of one or more EBCDIC uppercase letters A–Z and 0–9, the first character of which must be an uppercase letter).
- The clock value—the middle 6 bytes of an 8-byte store clock (STCK) value
- A 2-byte sequence number.

If you omit a unit-of-work identifier (by specifying a null pointer) the external CICS interface generates one for you, consisting of the following:

- 1-byte length field set to X'1A'
- 1-byte LU length field set to X'11'
- A 17-byte LU name consisting of:
 - An 8-byte eye-catcher set to 'DFHEXCIU'.
 - A 1-byte field containing a period (.)
 - A 4-byte field containing the VSE, in characters, under which the client is running.
 - A 4-byte field containing the address space id (ASID) in which the VSE client program is running. The field contains the four character EBCDIC representation of the two byte hex address space id.
- The clock value—the middle 6 bytes of an 8-byte store clock (STCK) value
- A two byte sequence number set to X'0001'.

userid

An 8-character input area containing the external security manager (ESM) userid for user security checking in the CICS region. The external CICS interface passes this userid to the CICS server region for user resource and command security checking in the server application program.

A userid is required only if the MRO connection specifies the ATTACHSEC(IDENTIFY) attribute. If the connection specifies ATTACHSEC(LOCAL), the CICS server region applies link security checking only.

See Chapter 8, “Security” on page 57 for information about external CICS interface security considerations.

This parameter is optional. However, if you don't specify a userid, the external CICS interface passes the security userid under which the client program is running. For example, if the client program is running as a VSE batch job, the external CICS interface obtains and passes the userid specified on the USER parameter of the ID statement in the batch job JCL.

If you want to let *userid* default, you must ensure that the CALL parameter list contains a null address for this parameter. How you do this depends on the language you are using for the non-CICS client program. For an example of a

call that omits an optional parameter, see “Example of EXCI CALLs with null parameters” on page 28.

dpl_retarea

A 12-byte output area into which the DPL_Request processor places responses to the DPL request. Generally, these responses are from CICS, but in some cases the error detection occurs in the external CICS interface, which returns exception conditions that are the equivalent of those returned by an EXEC CICS LINK command.

This field is only meaningful in the following circumstances:

- The response field of the EXCI *return-area* has a zero value, or
- The EXCI *return-area* indicates that the server program has abended (response=USER_ERROR and reason=SERVER_ABENDED).

The 12-bytes form three fields, providing the following information:

1. The first field, a fullword value, contains a RESP value from the DPL_Request call. See “Exception conditions” on page 33 for the RESP values that can be returned on a DPL_Request call.

If the DPL_Request call reaches CICS, this field contains the EIBRESP value, otherwise it contains an equivalent response set by the external CICS interface. If this field is set by the external CICS interface, RESP is further qualified by a RESP2 value in the second field.

A zero value is the normal response, which equates to EXEC_NORMAL in the return codes copybooks.

2. The second field, a fullword value, may contain a RESP2 value from the link request, further qualifying the RESP value in field 1.

If the DPL_Request call reaches CICS, the RESP2 field is always null (CICS does not return RESP2 values across MRO links).

If the RESP field is set by the external CICS interface, it is further qualified by a RESP2 value in this second field. For example, if the *data_len* parameter specifies a value greater than the *COMMAREA_len* parameter, the external CICS interface returns the RESP value 22 (which equates to EXEC LENGERR in the return codes copybooks), and a RESP2 value of 13.

See the LINK conditions in *CICS Application Programming Reference* manual for full details of the possible RESP and RESP2 values.

Note: Special use of the RESP2 field by the data transformer program. If any error occurs in the transformer, the error is returned in RESP2.

3. The third field, a 4-character field, contains:

- The abend code if the server program abended
- 4 blanks if the server program did not abend.

If a server program abends, it is backed out to its last syncpoint which may be the start of the task, or an intermediate syncpoint. The server program can issue intermediate syncpoints because SYNCONRETURN is forced.

DPL_opts

A 1-byte input area indicating options to be used on the DPL_Request call.

For CICS Transaction Server for VSE/ESA Release 1, X'80' is the only valid

option, and it indicates that SYNCONRETURN is specified. SYNCONRETURN is mandatory.

The equated value for this parameter in the CICS-supplied copybook DFHXCPLx (where x indicates the language) is SYNCONRETURN. See page 27 for copybook details.

SYNCONRETURN specifies that the server region is to take a syncpoint on successful completion of the server program. Note that although SYNCONRETURN is mandatory, this does not prevent a server program from taking its own explicit syncpoints.

Response and reason codes on the DPL call

For all non-zero response codes, a unique reason code value identifies the reason for the response.

Note: All numeric response and reason code values are in decimal.

The following is a summary of the response and reason codes that the external CICS interface can return on the DPL call:

Response OK:

0 NORMAL

Response WARNING:

6 IRP_IOAREA_FM_FAILURE
7 SERVER_TERMINATED

Response RETRYABLE:

203 NO_CICS

Response USER_ERROR:

401 INVALID_CALL_TYPE
402 INVALID_VERSION_NUMBER
404 INVALID_USER_TOKEN
406 PIPE_NOT_OPEN
407 INVALID_USERID
408 INVALID_UOWID
409 INVALID_TRANSID
414 IRP_ABORT_RECEIVED
415 INVALID_CONNECTION_DEFN
416 INVALID_CICS_RELEASE
417 PIPE_MUST_CLOSE
418 INVALID_PIPE_TOKEN
422 SERVER_ABENDED

Response SYSTEM_ERROR:

612 TRANSFORM_1_ERROR
613 TRANSFORM_4_ERROR
614 IRP_NULL_DATA_RECEIVED
615 IRP_NEGATIVE_RESPONSE
616 IRP_SWITCH_PULL_FAILURE
617 IRP_IOAREA_GM_FAILURE
619 IRP_BAD_IOAREA

620 IRP_PROTOCOL_ERROR
622 ESTAEX_SETUP_FAILURE
623 ESTAEX_INVOKED
624 SERVER_TIMEDOUT
625 STIMER_SETUP_FAILURE
626 STIMER_CANCEL_FAILURE
629 SERVER_PROTOCOL_ERROR

For information about response codes, see “Response code values” on page 26.

For information about the reason codes, see Chapter 10, “Response and reason codes returned on EXCI calls” on page 73.

Close_Pipe

Disconnect an open pipe from CICS. The pipe remains in an allocated state, and its tokens remain valid for use by the same user. To reuse a closed pipe, the client program must first reissue an Open_Pipe command using the pipe token returned on the Allocate_Pipe command for the pipe. Pipes should not be left open when not in use because this prevents CICS from shutting down its IRC facility in an orderly manner. Therefore, the Close_Pipe command should be issued as soon as possible after all DPL_Request calls have completed.

```
▶▶—CALL DFHXCIS,(version_number,—return_area,—user_token,——————▶▶  
▶—call_type,pipe_token),VL,MF=(E,(1))—▶▶
```

Close_Pipe command parameters

version_number

A full-word binary input area indicating the version of the external CICS interface parameter list being used. It must be set to 1 in the client program.

The equated value for this parameter in the CICS-supplied copybook DFHXCPLx (where x indicates the language) is VERSION_1. See page 27 for copybook details.

return_area

A 5-word output area to receive response and reason codes, and a message pointer field. For more details, see “Return area for the EXCI CALL interface” on page 27.

user_token

The 1-word input area specifying the token, returned to the client program by EXCI on the Initialize_User command, that represents the user of the pipe being closed.

call_type

A 1-word input area indicating the function of the command. This must be set to 4 in the client program to indicate that this is a Close_Pipe command.

The equated value for this call in the CICS-supplied copybook DFHXCPLx (where x indicates the language) is Close_Pipe. See page 27 for copybook details.

pipe_token

A 1-word input area specifying the token, returned to the client program by EXCI on the original Allocate_Pipe command, that represents the pipe being closed.

Response and reason codes on the Close_Pipe call

For all non-zero response codes, a unique reason code value identifies the reason for the response.

Note: All numeric response and reason code values are in decimal.

The following is a summary of the response and reason codes that the external CICS interface can return on the Close_Pipe call:

Response OK:

0 NORMAL

Response WARNING

2 PIPE_ALREADY_CLOSED

Response USER_ERROR

401 INVALID_CALL_TYPE

402 INVALID_VERSION_NUMBER

404 INVALID_USER_TOKEN

418 INVALID_PIPE_TOKEN

Response SYSTEM_ERROR

610 IRC_DISCONNECT_FAILURE

622 ESTAEX_SETUP_FAILURE

623 ESTAEX_INVOKED

For information about response codes, see “Response code values” on page 26.

For information about the reason codes, see Chapter 10, “Response and reason codes returned on EXCI calls” on page 73.

Deallocate_Pipe

Deallocate a pipe from CICS. On completion of this command, the pipe can no longer be used, and its associated tokens are invalid. This command should be issued for pipes that are no longer required. This command frees storage associated with the pipe.

```
▶▶—CALL DFHXCIS, (version_number, —return_area, —user_token, —call_type, —————▶  
▶—pipe_token), VL, MF=(E, (1))—▶◀
```

Deallocate_Pipe command parameters

version_number

A full-word binary input area indicating the version of the external CICS interface parameter list being used. It must be set to 1 in the client program.

The equated value for this parameter in the CICS-supplied copybook DFHXCPLx (where x indicates the language) is VERSION_1. See page 27 for copybook details.

return_area

A 5-word output area to receive response and reason codes, and a message pointer field. For more details, see “Return area for the EXCI CALL interface” on page 27.

user_token

A 1-word input area containing the token returned on the Initialize_User command.

call_type

A 1-word input area indicating the function of the command. This must be set to 5 in the client program to indicate that this is a Deallocate_Pipe command.

The equated value for this call in the CICS-supplied copybook DFHXCPLx (where x indicates the language) is Deallocate_Pipe. See page 27 for copybook details.

pipe_token

A 1-word input area containing the token passed back on the original Allocate_Pipe command, that represents the pipe now being deallocated.

Response and reason codes on the Deallocate_Pipe call

For all non-zero response codes, a unique reason code value identifies the reason for the response.

Note: All numeric response and reason code values are in decimal.

The following is a summary of the response and reason codes that the external CICS interface can return on the Deallocate_Pipe call:

Response OK:

0 NORMAL

Response WARNING:

5 XPIPE_FREEMAIN_ERROR

6 IRP_IOAREA_FM_FAILURE

Response USER_ERROR:

401 INVALID_CALL_TYPE

402 INVALID_VERSION_NUMBER

404 INVALID_USER_TOKEN

405 PIPE_NOT_CLOSED

418 INVALID_PIPE_TOKEN

Response SYSTEM_ERROR:

611 IRC_LOGOFF_FAILURE

622 ESTAEX_SETUP_FAILURE

623 ESTAEX_INVOKED

For information about response codes, see “Response code values” on page 26.

For information about the reason codes, see Chapter 10, “Response and reason codes returned on EXCI calls” on page 73.

Response code values

The values that can be returned in the response field are shown in Table 2 (all values are in decimal):

Table 2. EXCI response codes (returned in response field of *return_area*)

Value	Meaning	Explanation
0	OK	For all EXCI CALL commands other than the DPL command, the call was successful. If an OK response is received for a DPL command, you must also check <i>Dpl_retarea</i> to ensure CICS did not return a condition code. If the EIBRESP field of <i>Dpl_retarea</i> is zero, the DPL call was successful.
4	WARNING	The external CICS interface detected an error, but this did not stop the CALL command completing successfully. The reason code field describes the error detected.
8	RETRYABLE	<p>The EXCI CALL command failed. This class of failure relates to errors in the setup of the system environment, and not errors in the external CICS interface or client program. The reason code documents the specific error in the environment setup.</p> <p>The external CICS interface command can be reissued without changing the client program once the environment error has been corrected. The environmental errors concerned are ones that do not require a VSE re-IPL. Each reason code value for a RETRYABLE response documents whether the CALL can be reissued directly, or whether the pipe being used has to be closed and reopened first.</p>
12	USER_ERROR	The EXCI CALL command failed. This class of error means there is an error either in the client program, or in the CICS server program, or in the CICS server region. An example of an error in the CICS server system would be a failed security check, or an abend of the CICS server program, in which case the abend code is set in the abend code field of <i>DPL_retarea</i> . Each reason code value for a response of USER_ERROR explains whether the command can be reissued directly, or whether the pipe being used has to be closed and reopened first.
16	SYSTEM_ERROR	The EXCI CALL command failed. This class of error means that the external CICS interface has detected an error. The reason code value identifies the specific error. If the error can be corrected, then the command can be reissued. Each reason code value for a SYSTEM_ERROR response explains whether the command reissued directly, or whether the pipe being used has to be closed and reopened first.

Return area for the EXCI CALL interface

The format of the 5-word return area for the EXCI CALL interface is as follows:

1. One-word response field
2. One-word reason field
3. Two one-word subreason fields—subreason field-1 and subreason field-2
4. One-word CICS message pointer field. This is zero if there is no message present. If a message is present, this field contains the address of the storage area containing the message, which is formatted as follows:
 - A 2-byte LL field. LL is the length of the message plus the length of the LLBB field.
 - A 2-byte BB field, set to binary zero.
 - A variable length field containing the text of the message.

Return area and function call EQUATE copybooks

CICS provides four language-specific copybooks that map the storage areas for the *return_area* and *dpl_retarea* parameters of the EXCI CALL commands. The copybooks also provide EQUATE statements for each type of EXCI CALL.

These copybooks, and the libraries they are supplied in for the supported languages, are shown in Table 3.

Copybook name	Language	Library
DFHXCPLD	Assembler	PRD1BASE
DFHXCPLH	C	PRD1BASE
DFHXCPLD	COBOL	PRD1BASE
DFHXCPLL	PL/I	PRD1BASE

Reviewers

Check library naming conventions

Return codes

All the possible return codes are contained in a CICS-supplied copybook, which you must include in the program source of your external, non-CICS program. The names of the copybooks for the supported languages, and the libraries they are supplied in, are shown in Table 4.

Copybook name	Language	Library
DFHXCRCRCD	Assembler	PRD1BASE
DFHXCRCRCH	C	PRD1BASE
DFHXCRCRCD	COBOL	PRD1BASE
DFHXCRCRCL	PL/I	PRD1BASE

Dpl_Retarea return codes

These are the same as for CICS-to-CICS EXEC CICS DPL commands but with the following additions for the EXCI call interface:

Table 5. Exceptional conditions. RESP and RESP2 values returned to DPL_RETAREA

Condition	RESP2	Meaning
INVREQ	21	SYNCONRETURN_NOT_SPECIFIED
LENGERR	22	COMMAREA_LEN_TOO_BIG
LENGERR	23	COMMAREA_BUT_NO_COMMAREA_LEN

SYSIDERR also may be returned on an EXCI DPL request if the DPL request was to a program defined as remote and the link between CICS regions is down. In this situation, SYSIDERR is returned in the first word of the DPL_Retarea (code 53). The reason for the error is placed in the second word of this area; for programming information about the reason, see the SYSIDERR section of the notes on EIBRCODE in the *CICS Application Programming Reference* manual.

TERMERR also may be returned on an EXCI DPL request if the DPL request was to a program defined as remote, and an unrecoverable error occurs during conversation with the mirror on the remote CICS region. For example, suppose client program BATCH1 issues an EXCI DPL request to CICSA for program PROG1, which is defined as remote, and the request is function-shipped to CICSB where the program resides. If the session between CICSA and CICSB fails, or CICSB itself fails whilst executing the program PROG1, then TERMERR is returned to CICSA, and in turn to BATCH1.

No unique EXCI_DPL_RESP2 values are returned for TERMERR, PGMIDERR, NOTAUTH, and ROLLBACK.

Example of EXCI CALLS with null parameters

If you omit an optional parameter, such as *userid* on a DPL call, you must ensure that the parameter list is built with a null address for the missing parameter. The example that follows illustrates how to issue an EXCI DPL call with the *userid* and *uowid* parameters omitted in a COBOL program.

DPL CALL without userid and uowid (COBOL): In this example, the DPL parameters used on the call are defined in the WORKING-STORAGE SECTION, as follows:

DPL parameter	COBOL variable	
version_number	01 VERSION-1	PIC S9(8) COMP VALUE 1.
return_area	01 EXCI-RETURN-CODE	structure
user_token	01 USER-TOKEN	PIC S9(8) COMP VALUE ZERO.
call_type	03 DPL-REQUEST	PIC S9(8) COMP VALUE 6.
Pipe_token	01 PIPE-TOKEN	PIC S9(8) COMP VALUE ZERO.
pgmname	01 TARGET-PROGRAM	PIC X(8) VALUE "DFH\$AXCS".
commarea	01 COMMAREA.	structure
commarea_len	01 COMM-LENGTH	PIC S9(8) COMP VALUE 98.
data_len	01 DATA-LENGTH	PIC S9(8) COMP VALUE 18.
transid	01 TARGET-TRANSID	PIC X(4) VALUE "EXCI".

DPL parameter **COBOL variable**

dpl_retarea	01 EXCI-DPL-RETAREA.	structure
dpl_opts	01 SYNCONRETURN	PIC X VALUE X"80".

The variable used for the null address is defined in LINKAGE SECTION, as follows:

```
LINKAGE SECTION.  
01 NULL-PTR      USAGE IS POINTER.
```

Using the data names specified in WORKING-STORAGE SECTION as described above, and the NULL-PTR name as described in the LINKAGE SECTION, the following invocation of the DPL function omits the *uowid* and the *userid* parameters, and replaces them in the parameter list with the NULL-PTR variable:

```
DPL-SECTION.  
*  
    SET ADDRESS OF NULL-PTR TO NULLS.  
*  
    CALL 'DFHXCIS' USING VERSION-1    RETAREA      USER-TOKEN  
                                     DPL-REQUEST    PIPE-TOKEN    TARGET-PROGRAM  
                                     COMMAREA        COMM-LENGTH   DATA-LENGTH  
                                     TARGET-TRANSID NULL-PTR      NULL-PTR  
                                     DPL-RETAREA    SYNCONRETURN.
```

This example is taken from the CICS-supplied sample external CICS interface program, DFH0CXCC, which is supplied in PRD1.BASE. For an example of how to omit the same parameters from the DPL call in the other supported languages, see the following sample programs:

- DFH\$AXCC** The assembler sample
- DFH\$PXCC** The PL/I sample
- DFH\$DXCC** The C sample.

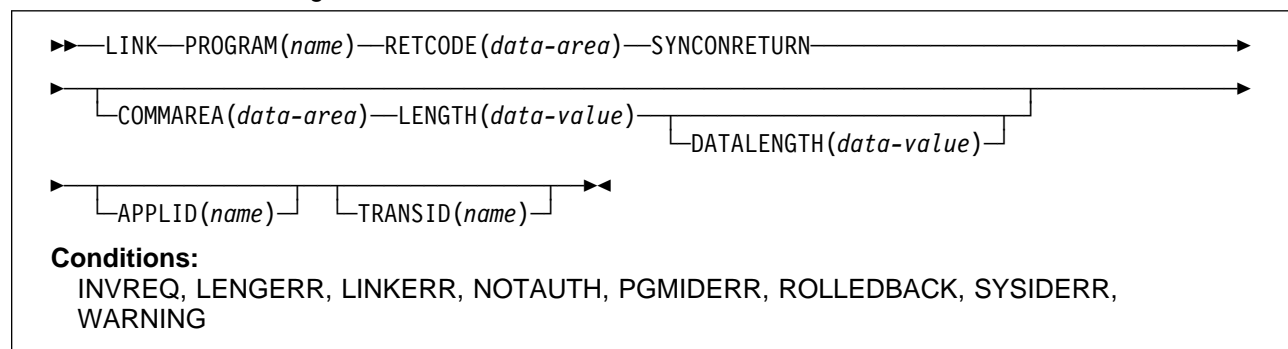
Chapter 3. The EXEC CICS interface

This chapter describes the EXEC CICS LINK PROGRAM command for the external CICS interface.

The external CICS interface provides this as a single, composite command, to invoke all the calls of the EXCI CALL interface. Each time you issue an EXEC CICS LINK PROGRAM command in a client application program, the external CICS interface invokes on your behalf each of the six EXCI calls.

EXEC CICS LINK command

Link from a VSE client program to the specified server program in a server CICS region.



With the exception of the APPLID and RETCODE parameters, the external CICS interface parameters for an EXEC CICS LINK command are the same as for a CICS-CICS DPL command.

This book describes only those parameters that you can use with the external CICS interface. For programming information about the EXEC CICS LINK PROGRAM command, see the *CICS Application Programming Reference* manual.

Note that the LENGTH and DATALENGTH parameters specify half-word binary values, unlike the corresponding *COMMAREA_len* and *data_len* parameters of the EXCI CALL interface, which specify full-word values.

An external CICS interface EXEC CICS LINK command always uses a generic connection.

LINK PROGRAM options

The parameters that you can use on the external CICS interface form of the LINK command, are as follows:

APPLID

specifies the generic APPLID of the target CICS server region.

Although an applid is required for an external CICS interface command, this parameter is optional on the LINK command itself because you can also specify it in the user-replaceable module, DFHXCURM. If you omit the generic APPLID from the LINK command, you must ensure it is specified by the user-replaceable module, DFHXCURM, on the URMAPPL parameter. You can also use the URMAPPL parameter in DFHXCURM to override an applid

specified on the LINK command. See Chapter 5, “The EXCI user-replaceable module” on page 45 for information about the URMAPPL parameter.

COMMAREA(data-area)

specifies a communication area that is to be made available to the invoked program. In this option, a pointer to the data area is passed.

See the *CICS Application Programming Guide* for more information about passing data to CICS application programs.

DATALENGTH(data-value)

specifies a halfword binary value that is the length of a contiguous area of storage, from the start of the COMMAREA, to be passed to the invoked program. If the amount of data being passed in a COMMAREA is small, but the COMMAREA itself is large so that the linked-to program can return the requested data, you should specify DATALENGTH in the interest of performance.

LENGTH(data-value)

specifies a halfword binary value that is the length in bytes of the communication area.

PROGRAM(name)

specifies the program name (1-8 characters) of the CICS server application program to which control is to be passed unconditionally. The specified name must either have been defined as a program to CICS, or the CICS server region must be capable of autoinstalling a definition for the named program.

Note the use of quotes:

```
EXEC CICS LINK PROGRAM('PROGX')
```

PROGX is in quotes because it is the program name.

```
EXEC CICS LINK PROGRAM(DAREA)
```

DAREA is not in quotes because it is the name of a data area that contains the 8-character program name.

RETCODE

specifies a 20-byte area into which the external CICS interface places return code information. This area is formatted into 5 one-word fields as follows:

- RESP** The primary response code indicating whether the external CICS interface LINK command caused an exception condition during its execution.
- RESP2** The secondary response code that further qualifies, where necessary, some of the conditions raised in the RESP parameter.
- ABCODE** Contains a valid CICS abend code if the server program abended in the server region.
- MSGLEN** Indicates the length of the message (if any) issued by the CICS server region during the execution of the server program. Note that the length is the actual length of the message text only, and does not include this one-word length field.
- MSGPTR** This is the address of the message text returned by the CICS server region.

Note: MSGLEN and MSGPTR are only valid on a LINKERR condition, with the RESP2 value 414.

SYNCONRETURN

specifies that the CICS server region, named on the APPLID parameter, is to take a syncpoint on successful completion of the server program.

SYNCONRETURN is mandatory for an external CICS interface LINK command.

TRANSID(name)

specifies the name of the mirror transaction that the remote region is to attach, and under which it is to run the server program. If you omit the TRANSID option, the CICS server region attaches CSMI.

Note: The TRANSID option specified on the LINK command overrides any TRANSID option specified on the program resource definition installed in the CICS server region.

While you can specify your own name for the mirror transaction initiated by DPL requests, the transaction *must* be defined in the server region, and the transaction definition must specify the mirror program, DFHMIRS. Defining your own transaction to invoke the mirror program gives you the freedom to specify appropriate values for some other options on the transaction resource definition.

See also the important rules about specifying transid with a DPL_Request on page 16 .

Exception conditions

Most of the exception conditions that are returned on the external CICS interface LINK command are the same as for the CICS-to-CICS distributed program link command. Those that are the same, and their corresponding numeric values are as follows:

INVREQ	16
LENGERR	22
PGMIDERR	27
SYSIDERR	53
NOTAUTH	70
TERMERR	81
ROLLEDBACK	82

These exception condition codes are returned in the RESP field.

RESP and RESP2

References to the RESP and RESP2 fields in this section are to the first 2 fields of the RETCODE parameter.

The exception conditions that are specific to the external CICS interface are as follows:

- Some of the RESP2 values on the error conditions INVREQ and LENGERR are specific to the external CICS interface.
- The exception conditions WARNING and LINKERR are specific to the external CICS interface.

The WARNING and LINKERR exceptions are a result of responses to individual EXCI calls issued by the external CICS interface in response to an EXEC CICS LINK command. These WARNING and LINKERR exceptions correspond to EXCI call responses as follows:

WARNING (RESP value 4)

This is returned when the EXCI module handling the EXEC CICS LINK request receives a USER_ERROR or SYSTEM_ERROR response to a Close_Pipe or Deallocate_Pipe request issued on behalf of an EXEC CICS LINK command. The RESP value is set to WARNING because the DPL request to CICS completed successfully, but an error occurred in subsequent processing.

The RESP2 field is set to the EXCI reason code, which gives more information about the error.

LINKERR (RESP value 88)

This is returned when the EXCI module handling the EXEC CICS LINK request receives a RETRYABLE, USER_ERROR or SYSTEM_ERROR response to an EXCI call issued on behalf of the EXEC CICS LINK command. The DPL request has failed. The RESP2 field is set to the EXCI reason code, which gives more information about the error.

See Chapter 10, “Response and reason codes returned on EXCI calls” on page 73 for descriptions of EXCI reason codes.

Note: The external CICS interface ignores any WARNING conditions that occur in response to EXCI calls it issues on behalf of an EXEC CICS LINK command. It treats the WARNING on an EXCI call as a good response and continues normally. If no other errors occur, the EXEC CICS command completes with a zero response in the EXEC_RESP field.

Retries on an EXEC CICS LINK command

If the external CICS interface receives a RETRYABLE response on an EXCI call that it makes on behalf of an EXEC CICS LINK command, it automatically retries the EXEC CICS LINK command up to five times, providing more serious errors do not occur. If the RETRYABLE response is still received after the fifth retry, the RESP field is set to LINKERR, and the reason returned on the EXCI CALL request that causes the exception is returned in the RESP2 field.

The external CICS interface retries the EXEC CICS LINK command by first closing and deallocating the pipe, then reissuing the six EXCI CALL commands. During Allocate_Pipe processing, the EXCI CALL interface calls the user-replaceable module, DFHXCURM, to give you the opportunity to change the APPLID of the CICS system to which the request has been sent. See Chapter 5, “The EXCI user-replaceable module” on page 45 for details of DFHXCURM.

Table 6 lists all the exception conditions and RESP2 values that are specific to the EXEC CICS LINK command for the external CICS interface.

Table 6 (Page 1 of 4). Exceptional conditions. RESP and RESP2 values returned from the EXEC API

Condition (RESP)	RESP2	Meaning
INVREQ (16)	21	SYNCONRETURN has not been specified

Table 6 (Page 2 of 4). Exceptional conditions. RESP and RESP2 values returned from the EXEC API

Condition (RESP)	RESP2	Meaning	
LENGERR (22)	22	COMMAREA length greater than 32763 bytes specified	
	23	COMMAREA specified but no LENGTH parameter specified	
	WARNING (4)	401	Invalid <i>call_type</i> parameter value specified on Close_Pipe or Deallocate_Pipe call
		402	Invalid <i>version_number</i> parameter specified on Close_Pipe or Deallocate_Pipe call
		404	Invalid <i>user_token</i> specified on Close_Pipe or Deallocate_Pipe call
		405	A Deallocate_Pipe call has been issued against a pipe that is not yet closed
		418	An invalid pipe token has been issued on a Close_Pipe or Deallocate_Pipe call
		610	There has been a CICS IRP logoff failure on a Deallocate_Pipe call
		611	There has been a CICS IRC disconnect failure on a Close_Pipe call
		622	There has been a VSE ESTAEX setup failure on a Close_Pipe or Deallocate_Pipe call
623	A program check on a Close_Pipe or Deallocate_Pipe call has caused the ESTAEX to be invoked		
LINKERR (88)	201	Command has been issued on a VSE image which has had no IRC activity since the previous IPL	
	202	There are no available sessions	
	203	CICS has not yet been brought up, or has not yet opened IRC	
	401	Invalid parameter	
	402	Invalid version number	
	403	User name is all blanks	
	404	Invalid address in user token	
	405	Command has been issued against a pipe that is not closed	
	406	Command has been issued against a pipe that is not open	
	407	Userid of all blanks has been passed	
	408	Error in UOWID parameter	
409	Transid consisting of all blanks or zero has been passed		
410	Load of message module, DFHMEBM, failed		
411	Load of message module, DFHMET4E, failed		

Table 6 (Page 3 of 4). Exceptional conditions. RESP and RESP2 values returned from the EXEC API

Condition (RESP)	RESP2	Meaning
LINKERR (88)	412	Load of DFHXCURM failed
	413	Load of DFHXCTRA failed
	414	If run as a CICS-to-CICS linked-to program, this server program would have resulted in an error with a appropriate message sent to the terminal. Running the program as an EXCI server program returns the message addressed by the MSGPTR field of the RETCODE area
	415	Target connection is an MRO connection, not an EXCI connection
	416	Command has been issued against a pre-CICS Transaction Server for VSE/ESA Release 1 system
	417	Command has been issued against a pipe in the MUST CLOSE state. Further EXCI EXEC CICS LINK commands will have unpredictable results and are, therefore, not permitted
	418	Pipe_token does not address an XCPIPE control block, or there is a mismatch between user_token and pipe_token
	419	CICS runs, or did run, under the VSE subtask that this command is attempting to use. This is not permitted and the command fails
	420	Load of DFHXCOPT failed
	422	The server has abended
	601	A VSE GETMAIN of working storage failed. This error leads to user abend 408
	602	A VSE GETMAIN failed. This error leads to user abend 403
	603	A VSE GETMAIN failed. This error leads to user abend 410
	604	A VSE GETMAIN failed
	605	A VSE GETMAIN for the VERIFY block failed. This error leads to user abend 409
	607	An SVC call failed. This error leads to user abend 406
	608	Logon to IRP failed
	609	Connect to IRP failed
	610	Disconnect from IRP failed
	611	Logoff from IRP failed
	612	Invalid data input to transformer_1
	613	Invalid data input to transformer_4
	614	CICS has responded but has not sent any data
	615	CICS cannot satisfy the request
	616	IRP_SWITCH_PULL request (to read data sent from CICS into a larger input/output area) has failed
	617	A GETMAIN for a larger input/output area failed

Table 6 (Page 4 of 4). Exceptional conditions. RESP and RESP2 values returned from the EXEC API

Condition (RESP)	RESP2	Meaning
LINKERR (88)	619	IRP has had a problem with the input/output area passed from the client program
	620	IRP has disconnected from EXCI
	621	A DISCONNECT command is issued in an error situation following an IRP CONNECT. The DISCONNECT has failed, indicating a serious error
	622	XCPRH ESTAEX set-up command failed. This error leads to user abend 402.
	623	XCPRH ESTAEX invoked due to program check during the processing of this command. ESTAEX attempts backout and takes a SYSDUMP. Further requests are permitted although the pipe is now in a MUST CLOSE state
	624	The DPL request has been passed to CICS but the time specified in DFHXCOPT has been exceeded. The request is aborted
	625	A VSE STIMER macro call failed
	626	A VSE STIMER CANCEL request failed
	627	The CICS SVC is at the incorrect level This error leads to user abend 407
	628	DFHIRP is at the incorrect level
	903	AN XCEIP ESTAEX set-up command failed
	904	An XCEIP ESTAEX invoked due to program check during the processing of this command. ESTAEX attempts backout and takes a SYSDUMP.

See “Return codes” on page 27 for details of the various copybooks that contain full details of all response and reason codes, including equated values.

Note: All numeric response and reason code values are shown in decimal.

Translation required for EXEC CICS LINK command

Application programs that use the EXEC CICS LINK form of the external CICS interface command must translate their programs before assembly or compilation. You do this using the version of the CICS translator that is appropriate for the language of your client program, specifying the translator option EXCI.

The translator option EXCI is mutually exclusive with the CICS and DLI options.

For more information about translating programs that contain EXEC CICS commands, see the *CICS Application Programming Guide*.

For information about compiling and link-editing external CICS interface client programs, see page 49.

Chapter 4. Defining connections to CICS

Connections between an EXCI client program and a CICS region require connection definitions in the CICS region. You define these using the CONNECTION and the SESSIONS resource definition facilities provided by CICS.

The following options are provided specifically for the external CICS interface:

- CONNTYPE on the CONNECTION resource definition
- EXCI on the PROTOCOL attribute of the CONNECTION and SESSIONS resource definitions.

CONNECTION resource definition

The EXCI option is provided on the PROTOCOL attribute of the CONNECTION resource definition to indicate that the connection is for use by a VSE program using the external CICS interface.

The CONNTYPE attribute is provided on the CONNECTION resource definition. For EXCI connections, this indicates whether the connection is generic or specific. It is not to be used for any protocol other than the external CICS interface.

```
Connection ==> ....
Group      ==> .....
Description ==> .....

CONNECTION IDENTIFIERS
Netname   ==> .....
INDsys    ==> ....

REMOTE ATTRIBUTES
...

CONNECTION PROPERTIES
Accessmethod ==> IRC           Vtam | IRc | INdirect | Xm
Protocol     ==> EXCI        Appc | Lu61 | EXCI
Conntype     ==>             Generic | Specific
SInglesess   ==> No          No | Yes
...
```

Figure 6. The DEFINE panel for CONNECTION

CONNTYPE({SPECIFIC|GENERIC})

For external CICS interface connections, indicates the nature of the connection.

SPECIFIC

The connection is for communication from a non-CICS client program to the CICS region, and is specific. A specific connection is an MRO link with one or more sessions dedicated to a single user in a client program.

Note: A *user* is a program that has issued an Initialize_user request (or for which an Initialize_user request has been issued), with a unique name per VSE task. For example:

- A simple client program running under VSE can be a single user of the external CICS interface.
- A client program running under VSE can open several pipes and issue external CICS interface calls over them sequentially, on behalf of different vendor packages. In this case, from the viewpoint of the client program, each of the packages is a user, identified by a unique user name. Thus a single client program can operate on behalf of multiple users.
- A program running under VSE can attach several VSE subtasks, under each of which a vendor package issues external CICS interface calls on its own behalf. Each package is a client program in its own right, and runs under its own VSE subtask. Each is also a user, with a unique user name.

For a specific connection, NETNAME is mandatory.

GENERIC

The connection is for communication from a non-CICS client program to the CICS system, and is generic. A generic connection is an MRO link with a number of sessions to be shared by multiple EXCI users. For a generic connection you cannot specify the NETNAME attribute.

Note: You can install only one generic EXCI connection in a CICS region.

NETNAME

For an external CICS interface connection, NETNAME corresponds to the name of the user of a specific pipe, as specified on the *user_name* parameter of an INITIALISE_USER call.

For an external CICS interface specific pipe, specify a NETNAME.

For external CICS interface generic pipes, you must leave NETNAME blank.

PROTOCOL({APPC|LU61|EXCI|blank})

The type of protocol that is to be used for the link.

blank

For MRO between CICS regions. Leave the PROTOCOL blank for MRO, and on the SESSIONS definition specify LU6.1 as the PROTOCOL.

APPC (LUTYPE6.2 protocol)

Advanced program-to-program communication, or APPC protocol. This is the default value for ACCESSMETHOD(VTAM). Specify this for CICS-CICS ISC.

LU61

LUTYPE6.1 protocol. Specify this for CICS-CICS ISC or CICS-IMS™ ISC, but not for MRO.

EXCI

The external CICS interface. Specify this to indicate that this connection is for use by a non-CICS client program using the external CICS interface.

SESSIONS resource definitions for EXCI connections

You indicate on the PROTOCOL attribute of the SESSIONS resource definition whether the sessions allocated on the MRO connection are for use by the external CICS interface.

```
Sessions ==> .....
Group    ==> .....
Description ==> .....

SESSION IDENTIFIERS
Connection ==> ....
SESSName  ==> ....
NETnameq  ==> .....
M0dename  ==> .....

SESSION PROPERTIES
Protocol  ==> Appc          Appc | Lu61 | EXCI
...
```

Figure 7. The DEFINE panel for SESSIONS

PROTOCOL({APPC|LU61|EXCI})

Indicates the type of protocol that is to be used for an intercommunication link (ISC or MRO).

APPC (LUTYPE6.2)

Advanced program-to-program communication (APPC) protocol. Specify this for CICS-CICS ISC.

LU61

LUTYPE6.1 protocol. Specify this for CICS-CICS ISC, for CICS-IMS, or for MRO.

EXCI

The external CICS interface. Specify this to indicate that the sessions are for use by a non-CICS client program using the external CICS interface. If you specify EXCI, leave SENDCOUNT blank.

RECEIVECOUNT({blank|number})

The number of MRO, LUTYPE6.1, or EXCI sessions that usually receive before sending.

For MRO, receive sessions can only receive before sending.

blank

These sessions can send only; there are no receive sessions.

number

Specifies the number of receive sessions on connections that specify blank, LU61, or EXCI on the protocol parameter of the CONNECTION definition. CICS uses the number to generate the last two or three characters of the session names (see RECEIVEPFX for details).

If you are using the default receive prefix (<), or your own 1-character prefix, specify a number in the range 1 through 999.

If you specify a 2-character prefix, the number is restricted to the range 1 through 99.

Except for external CICS interface (EXCI) connections, the RECEIVECOUNT in this system should equal SENDCOUNT in the other system.

RECEIVEPFX(<|prefix)

Specifies a 1-or 2-character prefix that CICS is to use as the first 1 or 2 characters of the receive session names (the names of the terminal control table terminal entries (TCTTEs) for the sessions).

Prefixes must not cause a conflict with an existing connection or terminal name.

< (MRO and EXCI sessions)

For MRO sessions, if you do not specify your own receive prefix, CICS enforces the default prefix—the less-than symbol (<), which is used in conjunction with the receive count to generate receive session names.

CICS creates the last three characters of the session names from the alphanumeric characters A through Z, and 1 through 9. These 3-character identifiers begin with the letters AAA, and continue in ascending sequence until the number of session entries reaches the limit set by the RECEIVECOUNT value. Note that receive session names are generated **after** the send sessions, and they follow in the same sequence.

For example, if the last session name generated for the send sessions is <AAJ, using the default prefix < CICS generates the receive session names as <AAK, <AAL, <AAM, and so on. (This method of generation of session identifiers is the same as for APPC sessions, except for the initial prefix symbol.)

Note: If you specify your own prefix, CICS generates the session names as in earlier releases, which is the same as for LUTYPE6.1 sessions.

prefix (LUTYPE6.1 sessions)

If the sessions are on LUTYPE6.1 ISC connections, specify a 1-or 2-character prefix. Do not use the default < symbol for LUTYPE6.1 sessions.

For LUTYPE6.1 sessions (and MRO if you specify your own 1-or 2-character prefix) CICS generates session names by appending a number to the prefix, either in the range 1 through 99, or 1 through 999. The number begins with 1 and is incremented by 1 until the specified RECEIVECOUNT is reached.

SENDCOUNT(blank|number)

The number of MRO or LUTYPE6.1 sessions that usually send before receiving.

For MRO, send sessions must send before they can receive.

blank

These sessions can receive only; there are no send sessions.

Leave this field blank when the sessions are on an external CICS interface (EXCI) connection.

number

Specifies the number of send sessions on connections that specify blank or LU61 on the protocol parameter of the CONNECTION definition. CICS

uses the number to generate the last two or three characters of the session names (see SENDPFX for details).

If you are using the default send prefix (>), or your own 1-character prefix, specify a number in the range 1 through 999.

If you specify a 2-character prefix, the number is restricted to the range 1 through 99.

Except for external CICS interface (EXCI) connections, SENDCOUNT in this system should equal RECEIVECOUNT in the other system.

SENDPFX(>|prefix)

Specifies a 1-or 2-character prefix that CICS is to use as the first 1 or 2 characters of the send session names (the names of the terminal control table terminal entries (TCTTEs) for the sessions).

Prefixes must not cause a conflict with an existing connection or terminal name.

> (MRO sessions)

For MRO sessions, if you do not specify your own send prefix, CICS enforces the default prefix—the greater-than symbol (>), which is used in conjunction with the send count to generate send session names.

CICS creates the last three characters of the session names from the alphanumeric characters A through Z, and 1 through 9. These 3-character identifiers begin with the letters AAA, and continue in ascending sequence until the number of session entries reaches the limit set by the SENDCOUNT value.

For example, using the default prefix (>) CICS generates session names as >AAA, >AAB, >AAC, and so on. (This method of generation of session identifiers is the same as for APPC sessions, except for the initial symbol.)

Note: If you specify your own prefix, CICS generates the session names as in earlier releases, which is the same as for LUTYPE6.1 sessions.

prefix (for LUTYPE6.1 sessions)

If the sessions are on LUTYPE6.1 ISC connections, specify a 1-or 2-character prefix. Do not use the default > symbol for LUTYPE6.1 sessions.

For LUTYPE6.1 sessions (and MRO if you specify your own 1-or 2-character prefix) CICS generates session names by appending a number to the prefix, either in the range 1 through 99, or 1 through 999. The number begins with 1 and are incremented by 1 until the specified SENDCOUNT is reached.

Inquiring on the state of EXCI connections

If you have access, through a CICS terminal, to the CICS server region, you can inquire about batch jobs that are connected to CICS through the interregion communication facility (IRC). These jobs run a client application program and use the external CICS interface to link to a server program in CICS.

To obtain this information about batch jobs linked to CICS through MRO, use the CEMT INQUIRE EXCI command. This command enables you to identify the names of batch jobs currently connected to CICS through the interregion

communication (IRC) facility. For jobs using the external CICS interface, the identification consists of:

```
jobname.execprogramname.partitionid
```

Information about jobs using the external CICS interface is available only when the job has issued at least one DPL request. A non-zero task number indicates that a DPL request is currently active. A zero task number indicates an external CICS interface session is still open (connected) for that job, although no DPL request is currently active.

See the *CICS-Supplied Transactions* manual for more information about the CEMT command.

Chapter 5. The EXCI user-replaceable module

This chapter contains Product-sensitive Programming Interface information.

The external CICS interface provides a user-replaceable module, DFHXCURM. The load module and the source in PRD1.BASE.

DFHXCURM is invoked in the non-CICS region during the processing of Allocate_Pipe commands, and after the occurrence of any retryable error. The retryable responses are:

- The target CICS region is not available
- There are no pipes available on the target CICS region
- There has been no IRC activity since the VSE/ESA IPL.

As supplied, DFHXCURM is effectively a dummy program because of a branch instruction that bypasses the sample logic and returns control to the external CICS interface caller. To use the sample logic, remove the branch instruction and assemble and link-edit the module. Customizing DFHXCURM allows you to do the following:

- When invoked during Allocate_Pipe processing, you can change the specified CICS APPLID, in order to route the request to another CICS system.
- When invoked after a retryable error you can store information regarding CICS availability. You can then use this information on the next invocation of DFHXCURM for Allocate_Pipe processing, so that you can decide to which CICS system to route the request.

DFHXCURM is called using standard VSE/ESA register conventions, with register 1 containing the address of the parameter list, and register 14 the return address of the caller. The parameters addressed by register 1 are mapped in the EXCI_URM_PARAMS DSECT, which is contained within the DFHXCPLD copybook. The parameters passed to DFHXCURM are as follows:

URMINV

The address of a full-word that contains the reason for the invocation of DFHXCURM, defined by the following equates:

URM_ALLOCATE	EQU 1	This invocation is for an Allocate_Pipe
URM_NO_CICS	EQU 2	The target CICS region is not available
URM_NO_PIPE	EQU 3	There are no pipes available
URM_NO_CICS_IRC	EQU 4	There has been no IRC activity since the VSE/ESA IPL

URMCICS

The address of an 8-byte area that contains the generic APPLID of the target CICS system, as specified on the *CICS_applid* parameter of the Allocate_Pipe command, or on the APPLID parameter of the EXEC CICS LINK command.

When specified by one of these commands, you can change the APPLID to that of a different target CICS region.

If the *CICS_applid* parameter is omitted from the Allocate_Pipe call, or APPLID is omitted from the EXEC CICS LINK command, the field addressed by this parameter contains 8 blanks. In this case, you must specify an APPLID in DFHXCURM before returning control to the caller.

URMAPPL

The address of an 8-byte area that contains the client program's user name as specified on the *my_name* parameter of the Initialize_User command. Note that if DFHXCURM is invoked for an EXEC CICS LINK command, this name is always set to DFHXCEIP.

URMPROG

The address of an 8-byte area that contains the name of the target program (if available). This name is available only if DFHXCURM is invoked for an EXEC CICS LINK command. For an external CICS interface Allocate_Pipe command, the program name is not known until the DPL call is issued.

URMOPTS

The address of a 1-byte area that contains the allocate options, which can be X'00' or X'80', as specified on the *allocate_opts* parameter. This address is valid for an Allocate_Pipe request only.

URMANCH

The address of a 4-byte area that is provided for use by DFHXCURM only. A typical use for this is to store a global anchor address of an area used to save information across a number of invocations of DFHXCURM. For example, you can GETMAIN the necessary storage and save the address in the 4-byte area addressed by this parameter. The initial value of the 4-byte area is set to zero.

Chapter 6. External CICS interface options table, DFHXCOPT

The EXCI options table, generated by the DFHXCOPT macro, enables you to specify a number of parameters that are required by the external CICS interface.

CICS provides the default DFHXCOPT table in source form, which you can tailor to your own requirements. The source of the default table and the load module are supplied in PRD1.BASE.

You assemble and link-edit the modified DFHXCOPT table into a suitable library defined in the LIBDEF PHASE, SEARCH library chain of the job that runs the VSE/ESA client program. Unlike the tables you specify for CICS regions, the DFHXCOPT table cannot be suffixed, and the external CICS interface component loads the first table of this name that it finds in the LIBDEF PHASE, SEARCH library chain.

Table 7 shows the format of the DFHXCOPT macro and its parameters.

Table 7. The DFHXCOPT macro parameters		
	DFHXCO	TYPE={ CSECT DSECT} [,MSGCASE={ MIXED UPPER}] [,TIMEOUT={0 number}] [,TRACE={ OFF 1 2}] [,TRACESIZE={16 number-of-kilobytes}] [,TRAP={ OFF ON}]
	END	End your parameters with the following END statement. DFHXCOPT

TYPE={**CSECT**|DSECT}

Indicates the type of table to be generated.

CSECT

A regular control section that is normally used.

DSECT

A dummy control section.

MSGCASE={**MIXED**|UPPER}

Specifies whether the DFHEXxxxx messages are to be issued in mixed or upper case.

MIXED

Code this if messages are to be issued in mixed case.

UPPER

Code this if messages are to be issued in upper case.

TIMEOUT={0|number}

Specifies the time interval, in hundredths of a second, during which the external CICS interface waits for a DPL command to complete.

0 Specifies that you do not want any time limit applied, and that the external CICS interface is to wait indefinitely for a DPL command to complete.

number

Specifies the time interval, in hundredths of a second, that the external CICS interface is to wait for a DPL command to complete. The number represents hundredths of a second, from 1 up to a maximum of 2147483647. For example:

6000 Represents a timeout value of one minute

30000 Represents a timeout value of five minutes

60000 Represents a timeout value of ten minutes.

TRACE={OFF|1|2}

Specifies whether you want external CICS interface internal tracing, and at what level.

OFF

External CICS interface internal tracing is not required. However, even with normal tracing switched off, exception trace entries are always written to the internal trace table.

1 Exception and level-1 trace entries are written to the internal trace table.

2 Exception, level-1, and level-2 trace entries are written to the internal trace table.

TRACESIZE={16|number-of-kilobytes}

Specifies the size in kilobytes of the internal trace table for use by the external CICS interface. This table is allocated in virtual storage, above 16MB, if available. You should ensure that there is enough virtual storage for the trace table by specifying a large enough partition size on the VSE/ESA ALLOC statement.

16 16KB is the default size of the trace table, and also the minimum size.

number-of-kilobytes

The number of kilobytes of storage to be allocated for the internal trace table, in the range 16KB through 1048576KB. The trace table storage exists for the duration of the VSE jobstep. The table is page-aligned and occupies a whole number of pages. If the value specified is not a multiple of the page size (4KB), it is rounded up to the next multiple of 4KB.

TRAP={OFF|ON}

Specifies whether the service trap module, DFHXCTRA, is to be used. DFHXCTRA is supplied as a user-replaceable module, in which IBM® service personnel can add code to trap errors.

OFF

Code this if you do not want to use DFHXCTRA.

ON

Code this if you require DFHXCTRA.

Chapter 7. Compiling and link-editing external CICS interface client programs

This chapter discusses the following topics:

- The external CICS interface stub, DFHXCSTB
- The required linkage-editor modes
- The CICS-supplied procedures for the external CICS interface
- Language considerations
- The CICS-supplied sample EXCI application client program
- Job control language to run an EXCI client program.

The external CICS interface stub, DFHXCSTB

All programs that use the external CICS interface to pass DPL requests to a CICS server region must include the CICS-supplied program stub, DFHXCSTB.

The stub intercepts all external CICS interface commands, whether they are EXCI CALL interface commands, or EXEC CICS LINK commands, and ensures they are passed to the appropriate external CICS interface routine for processing.

DFHXCSTB is a common stub, designed for inclusion in programs written in all the supported languages. It is supplied in the PRD1.BASE library.

The required linkage editor modes

You must specify AMODE(31) for your EXCI client program.

Language considerations

There are some language requirements that apply to writing a VSE/ESA client program that uses the external CICS interface. These affect programs written in PL/I and C.

PL/I considerations

PL/I programs written to the external CICS interface must provide their parameters on the CALL to DFHXCIS in the form of an assembler-style parameter list.

The EXCI copybook for PL/I, DFHXCPLL, contains the necessary definition of the DFHXCIS entry point, as follows:

```
DCL DFHXCIS ENTRY      OPTIONS(INTER ASSEMBLER);
```

The same rule applies for the EXCI LINK command, and in this case the CICS translator ensures that the correct parameter list is built.

For an example of an EXCI client program written in PL/I, see the source of the sample program, DFH\$PXCC.

C considerations

C programs written to the external CICS interface must provide their parameters on the CALL to DFHXCIS in the form of an assembler-style parameter list. You ensure this by declaring the entry point to DFHXCIS with OS LINKAGE.

The EXCI copybook for C, DFHXCPLH, contains the necessary definition of the DFHXCIS entry point, as follows:

```
#pragma linkage(dfhxcis,0S)
```

The same rule applies for the EXCI LINK command, and in this case the CICS translator ensures that the correct parameter list is built.

For an example of an EXCI client program written in C, see the source of the sample program, DFH\$DXCC.

Sample application programs

CICS provides a number of sample programs that are designed to help you write your own application programs. To help with writing programs that use the external CICS interface, CICS provides a sample VSE/ESA client program and a sample CICS server program.

The samples show you how to code client applications that use both the EXCI CALL interface and EXEC CICS LINK command.

Description of the sample applications

The sample external CICS interface programs are included with the CICS Transaction Server for VSE/ESA Release 1 product as part of the VSE/ESA base tape.

The sample VSE/ESA client program is provided in assembler language, COBOL for VSE, C for VSE, and PL/I for VSE. The sample CICS server program is provided in assembler only. Assembler language programs are in source and executable form. COBOL, PL/I, and C programs are provided in source form only. Each version of the client program has basically the same function, but programming methods vary somewhat according to the language used.

The sample programs, shown in Table 8 on page 51, are supplied in source form in PRD1BASE. The sample assembler server program is also supplied in executable form in PRD1BASE. The assembler client program is supplied in PRD1BASE.

Note: The assembler version of the client program uses SAM, which requires the program to be link-edited in RMODE(24). The assembler source code includes the required RMODE(24) statement. Normally, EXCI client programs run AMODE(31),RMODE(ANY).

<i>Table 8. The external CICS interface sample programs</i>		
Language	Name	Type of program
Assembler	DFH\$AXCC	Client program
Assembler	DFH\$AXCS	Server program
COBOL	DFH0CXCC	Client program
PL/I	DFH\$PXCC	Client program
C	DFH\$DXCC	Client program

Each version of the client is divided into three separate sections as follows:

1. The first section issues a single EXEC CICS LINK command to inquire on the state of the sample VSAM file, FILEA, in the target CICS system.

If the file is in a suitable state, processing continues to sections two and three, which together provide complete examples of the use of the EXCI CALL interface.
2. The second section initiates a specific MRO connection to the target CICS system and, once the pipe is open, performs a series of calls that each retrieve a single sequential record from the sample VSAM file, until no more records are available.
3. The third section is a simple routine to close the target sample file once processing of the data is complete. It also terminates the MRO connection now that the link is no longer required.

Some of the parameters used on the EXCI CALL and EXEC CICS LINK commands in the client program need to be tailored for your own target CICS server region. Change these as required, then re-translate, compile (or assemble) and link-edit the program.

The variables and their values specified in the sample programs are given in Table 9.

<i>Table 9. Parameters used in the sample client programs</i>	
Variable name in sample program	Value
TARGET_FILE	FILEA
TARGET_TRANSID	EXCI
TARGET_SYSTEM	DBDCCICS
TARGET_PROGRAM	DFH\$AXCS
APPLICATION	BATCHCLI

The assembler version of the client program is supplied pregenerated in an executable form. All versions of the program accept a run-time parameter to specify the target server region APPLID. For the pregenerated assembler version this avoids you having to reassemble the program to specify the applid of your own CICS server region. You can also use the sample client program with different CICS regions without needing to modify the program each time.

Installing the EXCI sample definitions

Resource definitions that support the EXCI sample programs are included in the CICS system definition file (CSD) in groups DFH\$EXCI and DFH\$FILA.

Note that the sample definitions, while included in the CSD, are not included in the IBM-defined group list DFHLIST. Thus, if CICS is initialized with GRPLIST=DFHLIST, you must install the EXCI resource definition groups before using the samples. Alternatively, you can add the sample groups to your startup group list, so that they are installed automatically at system initialization.

The resource definition groups that must be installed are as follows:

DFH\$EXCI This contains definitions for the sample server transaction, server program, EXCI connections and sessions.

Only one server program is included—in assembler language, called DFH\$AXCS.

The sample application is designed to run the transaction EXCI, which is defined to invoke the DFHMIRS mirror program and references profile DFHCICSA. The required transaction definition for EXCI is included in the group.

Sample CONNECTION and SESSIONS definitions for specific and generic connections are included.

Note: The generic connection definition supplied in the sample group DFH\$EXCI specifies ATTACHSEC(IDENTIFY). This security option causes the server program DFH\$EXCS to fail with an ATCY abend if you run the sample programs in an environment that does not have an external security manager (ESM) installed and active.

If you want to run the external CICS interface sample programs without any security active, you must alter the generic connection resource definition to specify ATTACHSEC(LOCAL).

DFH\$FILA This contains the definition for the supplied sample VSAM file, FILEA, which is referenced by the EXCI sample programs.

Once these are installed, ensure that interregion communication (IRC) is open. If IRC is not opened during CICS initialization, set it open using the CEMT SET IRC OPEN command.

Running the EXCI sample applications

If you want to use the COBOL, PL/I, or C version of the EXCI client program, you must translate, compile, and link-edit the program into a suitable library.

You can use the sample JCL shown in Figure 12 on page 56 as a basis for creating your own batch job to run the client program. Note the use of the OS390 parameter on the EXEC statement: this is mandatory.

If you use the pregenerated assembler version, specify the APPLID of your target CICS server region as a parameter on the EXEC statement for the client program, as follows:

```
/* -----*  
// EXEC      PGM=DFH$AXCC,PARM='applid',OS390
```

Results of running the EXCI sample applications

An example of the output produced by successful execution of the pregenerated assembler version of the client program, DFH\$AXCC, is shown in Figure 8 on page 54.

If an error occurs while running the application, then, assuming the error is not severe, messages are written to the SYSLST output log displaying the reasons and/or return codes that cause processing to be terminated. Several examples of error-invoked output are shown in Figure 9, Figure 10, and Figure 11 on page 55.

```

***** EXCI Sample Client Program *****
*
* EXEC Level Processor.
*   Setting up the EXEC level call.
*   The Link Request has successfully completed.
*   Server Response:
*     The file is set to a browsable state.
*
* CALL Level Processor.
*   Initialize_User call complete.
*   Allocate_Pipe call complete.
*   Open_Pipe call complete.
*   The connection has been successful.
*   The target file follows:
*
***** Top of File *****
000102F. ALDSON          WARWICK, ENGLAND  9835618326 11 81$1111.11Y00007300
000104S. BOWLER         LONDON,ENGLAND   1284629326 11 81$0999.99Y00007400
000106B. ADAMS          CROYDON, ENGLAND  1948567326 11 81$0087.71Y00007500
000111GENE BARLOWE     SARATOGA,CALIFORNIA 4612075301 02 74$0111.11Y00007600
000762GEORGE BURROW   SAN JOSE,CALIFORNIA 2231212101 06 74$0000.00Y00007700
000983H. L. L. CALL    WASHINGTON, DC    3451212021 04 75$9999.99Y00007800
001222J.R.REYNOLDS    BOBLINGEN, GERMANY 7031555110 04 73$3349.99Y00007900
001781HAROLD JAMES    SINDELINGEN,GERMANY7031999021 06 77$0009.99Y00008000
003210B.CREPIN        NICE, FRANCE     1234567026 11 81$3349.99Y00008100
003214HUBERT C HERBERT SUNNYVALE, CAL.   3411212000 06 73$0009.99N00008200
003890PHILIPPE SMITH, JR NICE, FRANCE     0000000028 05 74$0009.99N00008300
004004STAN SMITH      DUBLIN, IRELAND  7111212102 11 73$1259.99N00008400
004445S. GALSON       SOUTH BEND, S.DAK. 6121212026 11 81$0009.99N00008500
004878D.C. CURRENT    SUNNYVALE, CALIF. 3221212010 06 73$5399.99N00008600
005005J. S. LAVERENCE SAN FRANCISCO, CA. 0000000101 08 73$0009.99N00008700
005444JEAN LAWRENCE  SARATOGA, CALIF.  6771212020 10 74$0809.99N00008800
005581JOHN ALDEN III  BOSTON, MASS.    4131212011 04 74$0259.99N00008900
006016DR W. T. KAR    NEW DELHI, INDIA  7033121121 05 74$0009.88Y00009000
006670WILLIAM KAPP    NEW YORK, N.Y.    2121212031 01 75$3509.88N00009100
006968D. CONRAD       WARWICK, ENGLAND  5671382126 11 81$0009.88Y00009200
007007BRIGITTE EICRN  STUTTGART, GERMANY 7031100010 10 75$5009.88N00009300
007248B. C. WILLIAMSON REDWOOD CITY, CALF. 3331212111 10 75$0009.88N00009400
007779MRS. W. WELCH  SAN JOSE, CALIF.  4151212003 01 75$0009.88Y00009500
100000G. NEADS        TORONTO, ONTARIO 0341512126 11 81$0010.00Y00009600
111111C. MEARS        OTTAWA, ONTARIO  5121200326 11 81$0011.00Y00009700
200000A. BONFIELD     GLASGOW, SCOTLAND 6373829026 11 81$0020.00Y00009800
222222J. WIEBERS      FRANKFURT, GERMANY 2003415126 11 81$0022.00Y00009900
300000K. TRENCHARD  NEW YORK, U.S.    6473980126 11 81$0030.00Y00010000
333333D. MYRING       CARDIFF, WALES   7849302026 11 81$0033.00Y00010100
400000W. TANNER       MILAN, ITALY     2536373826 11 81$0040.00Y00010200
444444A. FISHER       CALGARY, ALBERTA 7788982026 11 81$0044.00Y00010300
500000J. DENFORD      MADRID, SPAIN    4445464026 11 81$0000.00Y00010400
555555C. JARDINE      KINGSTON, N.Y.   3994442026 11 81$0005.00Y00010500
600000F. HUGHES       DUBLIN, IRELAND  1239878026 11 81$0010.00Y00010600
666666A. BROOKMAN     LA HULPE, BRUSSELS 4298384026 11 81$0016.00Y00010700
700000A. MACALLA      DALLAS, TEXAS    5798432026 11 81$0002.00Y00010800
777777D. PRYKE        WILLIAMSBURG, VIRG. 9187613126 11 81$0027.00Y00010900
800000H. BRISTOW      WESTEND, LONDON  2423338926 11 81$0030.00Y00011000
888888B. HOWARD       NORTHAMPTON, ENG. 2369163926 11 81$0038.00Y00011100
900000D. WOODSON      TAMPA, FLA.      3566812026 11 81$0040.00Y00011200
999999R. JACKSON      RALEIGH, N.Y.    8459163926 11 81$0049.00Y00011300
***** End of File *****
*
* Closing Dpl Request has been attempted.
* Close_Pipe call complete.
* Deallocate_Pipe call complete.
*
***** End of EXCI Sample Client Program *****

```

Figure 8. Successful execution


```

===== EXCI Sample Client Program =====
*
* EXEC Level Processor.
* Setting up the EXEC level call.
* The Link Request has failed. Return codes are;
* Resp = 00000088 Resp2 = 00000203 Abend Code:
* >>> Aborting further processing <<<<
*
===== End of EXCI Sample Client Program =====

```

Figure 9. No CICS return code. The target CICS region specified by the client program is not found, or IRC was not opened.

```

===== EXCI Sample Client Program =====
*
* EXEC Level Processor.
* Setting up the EXEC level call.
* The Link Request has successfully completed.
* Server Response:
* The file could not be found.
* >>> Aborting further processing <<<<
*
===== End of EXCI Sample Client Program =====

```

Figure 10. No file found. The target file name to the server program was not found on the target CICS system.

```

===== EXCI Sample Client Program =====
*
* EXEC Level Processor.
* Setting up the EXEC level call.
* The Link Request has failed. Return codes are;
* Resp = 00000088 Resp2 = 00000414 Abend Code:
* A message was received from the target CICS system:
*
DFHAC2001 04/29/93 16:43:03 IYAHZCAZ Transaction 'BAD_' is unrecognized. Check
that the transaction name is correct.
*
* >>> Aborting further processing <<<<
*
===== End of EXCI Sample Client Program =====

```

Figure 11. Incorrect transaction identifier. The target transid passed in the external CICS interface call is not defined on the target CICS system. Note the message received from the target CICS system.

Job control language to run an EXCI client program

An EXCI client program runs in a VSE partition, for example, as a batch job.

Note the following requirements when writing the JCL for your client program:

- Include in the LIBDEF PHASE,SEARCH statement those libraries that contain the CICS-supplied external CICS interface modules and also the client program.

The external CICS interface modules are supplied in PRD1.BASE. These are:

```
DFH$AXCC
DFHMEBM
```

DFHMET4E
DFHXCEIX
DFHXCOPT
DFHXCPRX
DFHXCSTB
DFHXCTRA
DFHXCURM

- You are recommended to include a LIBDEF DUMP,CATALOG=*library.sublibrary* statement. The external CICS interface uses SYSDUMP for some error conditions.
- The EXCI job must run in a partition large enough to allow for the size of the internal trace table specified by the TRACESZE parameter in the DFHXCOPT options table.

Figure 12 shows a sample job that you can use or modify to start a client program.

```
// JOB EXCI  accounting_information
/******
/*    JCL to execute an external CICS interface client program    *
/******
// ID USER=userid,PWD=password
// LIBDEF PHASE,SEARCH=(lib.sublib,lib2,lib3,...)
// LIBDEF DUMP,CATALOG=lib.sublib
// EXEC  PGM=pgmname,OS390
```

Figure 12. Sample job for starting an EXCI client program

Notes:

1. The job userid, specified on the USER parameter of the ID statement in the batch job JCL, must be defined to an external security manager (ESM), if batch security is used.
2. In addition to being used for job step initiation security, the job userid is also used for MRO logon and bind-time security checking.

See Chapter 8, “Security” on page 57 for information about security in when using the external CICS interface.
3. See “Installing the EXCI sample definitions” on page 52 for information about modifying the sample connection definitions before you run the sample application programs in an environment that does not have an external security manager (ESM) installed and active.

Chapter 8. Security

CICS supports a number of security checks that operate on requests received from a VSE client program. These security checks use the security authorization facility (SAF) interface, and require the services of an external security manager (ESM).

The security checks are described in the following topics:

- MRO logon and connect security, performed by DFHIRP
- Link security, performed by the CICS server region
- User security checking in the server application program
- Surrogate user checking performed by the external CICS interface in the client program address space.

MRO logon and bind-time security

DFHIRP, the CICS interregion communication program, performs two security checks against users that want to:

1. Logon to IRP (**specific connections only**)
2. Connect to a CICS region (also referred to as bind-time security).

Generic EXCI connections

The discussion about logon security checking in this section applies only to EXCI connections that are defined as SPECIFIC. The MRO logon security check is not performed for generic connections.

The VSE client program is treated just the same as another CICS region as far as MRO logon and connect (bind-time) security checking is concerned. This means that when the client program logs on to the interregion communication program, IRP performs logon and bind-time security checks against the userid under which the client program is running. In the remainder of this chapter, we refer to this as the batch region's userid.

To enable your client program to logon successfully to IRP, and to connect to the target server region, first ensure that you define the batch region's userid in a user profile to the ESM. When you have defined the batch region's userid to the ESM, you can then give the batch region the appropriate logon and bind-time authorizations.

1. Logon authorization (specific connections only)

Authorize the batch region's userid to logon with the *user_name* defined on the INITIALIZE_USER command. Generally, depending on the ESM, the batch region requires UPDATE authority to the relevant IRP logon security profile.

Failure to authorize the batch region's userid to logon to IRP causes Allocate_Pipe processing to fail with RESPONSE(SYSTEM_ERROR) and REASON(IRC_LOGON_FAILURE). The subreason field-1 for a logon security check failure returns decimal 204.

2. Bind-time authorization

Ensure the batch region's userid has the appropriate authority to connect to the target CICS server region. Generally, depending on the ESM, the batch region requires READ authority to the relevant IRP bind security profile.

Failure to authorize the batch region's userid to connect to the server region causes Open_Pipe processing to fail with RESPONSE(SYSTEM_ERROR) and REASON(IRC_CONNECT_FAILURE). The subreason field-1 for a bind-time security check failure returns decimal 176.

See the ESM's documentation for information about how to specify MRO logon and bind-time security.

Link security

The target CICS server region performs link security checking against requests from the client program. These security checks cover transaction attach security (when attaching the mirror transaction), and resource and command security checking within the server application program. The link userid that CICS uses for these security checks is the batch region's userid.

To ensure these link security checks do not cause security failures, you must ensure that the link userid is authorized to the following resource profiles, as appropriate:

- The profile for the mirror transaction, either CSMI for the default, or the mirror transaction specified on the *transid* parameter. This is required for transaction attach security checking.
- The profiles for all the resources accessed by the CICS server application program—files, queues (transient data and temporary storage), programs, and so on. This is required for resource security checking.
- The CICS command profiles for the SPI commands issued by the CICS server application program—INQUIRE, SET, DISCARD and so on. This is required for command security checking.

User security

The target CICS server region performs user security checking against the userid passed on a DPL CALL request. User security checking is performed only when connections specify ATTACHCSEC(IDENTIFY).

User security is performed in addition to any link security.

For user security, in addition to any authorizations you make for link security, you must also authorize the userid specified on the DPL CALL request.

Note that there is no provision for specifying a userid on the EXEC CICS LINK command. In this case, the external CICS interface passes the batch region's userid. User security checking is therefore performed against the batch region's userid if the connection definition specifies ATTACHSEC(IDENTIFY).

Note: If your connection resource definitions for the external CICS interface specify ATTACHSEC(IDENTIFY), your server programs will fail with an ATCY abend if you run them in an environment that does not have an external security manager (ESM), installed and active.

If you want to run external CICS interface server programs without any security active, you must specify ATTACHSEC(LOCAL).

Surrogate user checking

A surrogate user check is performed to verify that the batch region's userid is authorized to issue DPL calls for another user (that is, is authorized as a surrogate of the userid specified on the DPL_request call).

EXCI client jobs are subject to surrogate user checking if SURROGCHK=YES (the default) is specified in the EXCI options table, DFHXCOPT. If you specify SURROGCHK=YES (or allow it to default) authorize the batch region's userid as a surrogate of the userid specified on all DPL_request calls.

If surrogate user checking is enabled (SURROGCHK=YES), but no userid is specified on the DPL call, no surrogate user check is performed, because the userid on the DPL call defaults to the batch region's userid. For this bypass of surrogate user checking to be successful, ensure that you have correctly omitted the userid on the DPL call. See "Example of EXCI CALLs with null parameters" on page 28 for information about the correct way to specify a null pointer when omitting an EXCI call parameter.

If you don't want surrogate user security checking, specify SURROGCHK=NO in the DFHXCOPT options table (note SURROGCHK=YES is the default).

Surrogate user checking is useful when the batch region's userid is the same as the CICS server region userid, in which case the link security check (see "Link security" on page 58) is bypassed. In this case, a surrogate user check is recommended, because the USERID specified on the DPL call is not an authenticated userid (no password is passed).

If the batch region's userid and the CICS region userid are different, link security checking is enforced. With link security, a non-authenticated userid passed on a DPL call cannot acquire more authority than allowed by the link security check. It can acquire only the same, or less, authority than allowed by the link security check.

Chapter 9. Problem determination

This chapter contains Diagnosis, Modification or Tuning information.

This chapter describes some of the aids to problem determination provided by the external CICS interface. It covers:

- Trace
- System dumps
- VSE/ESA abends
- The EXCI service trap, DFHXCTRA
- EXCI trace entry points.

Details of the external CICS interface messages and abend codes are given *VSE/ESA Messages and Codes - Volume 1*.

See also the "OS/390® API Return Codes" topic in *VSE/ESA Messages and Codes - Volume 1* for information about OS/390 return codes.

Trace

The external CICS interface writes trace data to an internal trace table. The internal trace table resides in the partition GETVIS in 31-bit storage, if available. Trace data is formatted and included in any dumps produced by the external CICS interface.

Trace entries are issued by the external trace interface destined for the internal trace table. They are listed in "EXCI trace entry points" on page 65.

Note: The external CICS interface maintains a separate trace table for each user VSE task in an external CICS interface application program.

The external CICS interface does not support any form of auxiliary trace.

To format external CICS interface trace entries, you use the same FID and ID as for CICS (that is, FID=X'EF', and ID=X'F6C').

System dumps

The external CICS interface produces VSE/ESA SDUMPs for error conditions. These dumps contain all the external CICS interface control blocks, as well as trace entries.

Formatting system dumps

You can use the CICS INFOANA exit, DFHPD410, to format the system dumps. The following keywords are available for use when formatting an external CICS interface dump using DFHPD410:

KE

Formats PSW and registers, and all external CICS interface control blocks.

LD

Formats a load map of where the external CICS interface modules are loaded in the address space, and gives their PTF level.

MRO

Formats the MRO control blocks for the external CICS interface address space, including common control blocks that reside in the VSE/ESA common service area (CSA). This option also formats some MRO blocks that reside in the CICS address space for pipes connected to CICS.

TR

Formats the external CICS trace table. You can format the trace table in abbreviated and full forms (TR=1 gives you the abbreviated trace).

SU

Produces a dump summary.

Multiple VSE tasks

If the external CICS interface takes a system dump when there is more than one VSE task in use, it dumps only the control blocks and trace table for the VSE task that requested the dump.

If you take a dump of the external CICS address space using a console command, the CICS verb exit routine, DFHPD410 formats the control blocks and trace tables for every VSE task using EXCI that it finds in the dump.

Abends from related CICS programs

Certain abends can occur in other CICS programs during the implementation of an external CICS interface client program. These include:

- The CICS translator
- The system dump formatter, DFHPD410
- The resource definition offline utility program, DFHCSDUP.

See the *VSE/ESA Messages and Codes* manual for details.

04xx (external CICS interface) abend codes

0401

Explanation: An external CICS interface (EXCI) request was issued using the CALL API or the EXEC API, and the EXCI stub DFHXCSTB link-edited with the application detected that it was running in AMODE 24. The external CICS interface only supports calls made in AMODE 31.

System Action: The application terminates abnormally.

User Response: Change the application so that EXCI calls are made in AMODE 31, or relink-edit the application AMODE 31.

Module: DFHXCSTB.

0402

Explanation: The external CICS interface module DFHXCPRH issued a VSE/ESA ESTAEX macro to establish a recovery environment, but a nonzero return code was returned from VSE/ESA.

System Action: The application terminates abnormally with a dump.

User Response: Examine the dump and any associated VSE/ESA messages produced to determine why the VSE/ESA ESTAEX request failed.

If the error occurred while processing an INITIALIZE_USER request on behalf of the application, an attempt to format the dump using the CICS INFOANA dump formatter does not produce any formatted output. This is because the error occurred too early in EXCI initialization for there to be any control blocks.

Module: DFHXCPRH

0403

Explanation: The external CICS interface module DFHXCPRH issued a VSE/ESA GETMAIN request to obtain storage for its XCGLOBAL block, but a nonzero return code was returned from VSE/ESA.

System Action: Module DFHXCPRH issues a VSE/ESA abend with abend code 0403 which invokes its ESTAEX routine to clear up its environment. A dump is taken before returning control to the application. An application using the EXCI CALL API receives RESPONSE(SYSTEM_ERROR) REASON(XCGLOBAL_GETMAIN_ERROR) in its return area. The subreason1 field of the return area contains the R15 return code from VSE/ESA indicating why the GETMAIN failed. An application using the EXCI EXEC API receives RESP(LINKERR) RESP2(602).

User Response: Use the VSE/ESA R15 return code obtained from the application or from the dump to determine why the VSE/ESA GETMAIN request failed. If the reason is insufficient storage, increase the region size of the batch application.

An attempt to format the dump produced with the CICS dump formatter does not produce any formatted output for the job because the error occurred too early in EXCI initialization for there to be any control blocks.

Module: DFHXCPRH

0404

Explanation: The external CICS interface module DFHXCPRH needed to take an VSE/ESA SDUMP for an earlier reported problem. However the error has occurred too early in EXCI initialization for full EXCI dump services to be available.

System Action: Module DFHXCPRH issues a VSE/ESA abend with abend code 0404 which invokes its ESTAEX routine from which an SDUMPX is taken, to capture the earlier reported problem.

User Response: Examine the dump to determine the cause of the earlier reported problem.

An attempt to format the dump produced with the CICS dump formatter does not produce any formatted output for the job because the error occurred too early in EXCI initialization for there to be any control blocks.

Module: DFHXCPRH

0406

Explanation: The external CICS interface module DFHXCPRH called the CICS SVC to initialize the EXCI environment. The CICS SVC call failed.

System Action: Module DFHXCPRH issues a VSE/ESA abend with abend code 0406 which invokes its ESTAEX routine to clear up its environment. A system dump is taken before returning control to the application. An application using the EXCI CALL API receives RESPONSE(SYSTEM_ERROR) REASON(CICS_SVC_CALL_FAILURE) in its return area. The subreason1 field of the return area contains the R15 return code from the CICS SVC indicating why it failed. An application using the EXCI EXEC API receives RESP(LINKERR) RESP2(607).

User Response: Use the VSE/ESA R15 return code obtained from the application or from the dump to determine why the CICS SVC call failed.

An attempt to format the system dump produced with the CICS INFOANA dump formatter does not produce any formatted output for the job because the error occurred too early in EXCI initialization for there to be any control blocks.

Module: DFHXCPRH

0407

Explanation: The external CICS interface module DFHXCPRH issued a call to the CICS SVC to check whether the SVC in use is at the correct level to be used with the external CICS interface. The check failed indicating that the CICS SVC is not at the correct level.

System Action: Message DFHEX0100 is output, and module DFHXCPRH issues a VSE/ESA abend with abend code 0407 which invokes its ESTAEX routine to clear up its environment. A system dump is taken before returning control to the application. An application using the EXCI CALL API receives RESPONSE(SYSTEM_ERROR) REASON(INCORRECT_SVC_LEVEL) in its return area. An application using the EXCI EXEC API receives RESP(LINKERR) RESP2(627).

User Response: See the explanation of message DFHEX0100 for guidance.

An attempt to format the system dump produced with the CICS INFOANA dump formatter does not produce any formatted output for the job because the error occurred too early in EXCI initialization for there to be any control blocks.

Module: DFHXCPRH

0408

Explanation: The external CICS interface module DFHXCPRH issued a VSE/ESA GETMAIN request for its working storage but a nonzero return code was returned from VSE/ESA.

System Action: Module DFHXCPRH issues a VSE/ESA abend with abend code 0408 which invokes its ESTAEX routine to clear up its environment. A system dump is taken before returning control to the application. An application using the EXCI CALL API receives RESPONSE(SYSTEM_ERROR) REASON(WS_GETMAIN_ERROR) in its return area. The subreason1 field of the return area contains the R15 return code from VSE/ESA indicating why the GETMAIN failed. An application using the EXCI EXEC API receives RESP(LINKERR) RESP2(601).

User Response: Use the VSE/ESA R15 return code obtained from the application or from the dump to determine why the VSE/ESA GETMAIN request failed. If the reason is insufficient storage, increase the region size of the batch application.

An attempt to format the system dump produced with the CICS INFOANA dump formatter does not produce any formatted output for the job because the error occurred too early in EXCI initialization for there to be any control blocks.

Module: DFHXCPRH

0410

Explanation: The external CICS interface module DFHXCPRH issued a VSE/ESA GETMAIN request for an XCUSER block but a nonzero return code was returned from VSE/ESA.

System Action: Module DFHXCPRH issues a VSE/ESA abend with abend code 0410 which invokes its ESTAEX routine to clear up its environment. A system dump is taken before returning control to the application. An application using the EXCI CALL API receives RESPONSE(SYSTEM_ERROR) REASON(XCUSER_GETMAIN_ERROR) in its return area. The subreason1 field of the return area contains the R15 return code from VSE/ESA indicating why the GETMAIN failed. An application using the EXCI EXEC API receives RESP(LINKERR) RESP2(603).

User Response: Use the VSE/ESA R15 return code obtained from the application or from the dump to determine why the VSE/ESA GETMAIN request failed. If the reason is insufficient storage, increase the region size of the batch application.

Module: DFHXCPRH

0412

Explanation: The external CICS interface dump module DFHXCEIP was processing an EXCI EXEC API request and detected that the EXEC parameter list passed to it contained a function that is not supported by the external CICS interface.

System Action: The application is abnormally terminated with a dump.

User Response: This error indicates the parameter list being passed to the EXCI has not been generated by the CICS translator. The translator should always be used. Correct the application to specify the correct EXCI EXEC API command.

An attempt to format the system dump produced with the CICS INFOANA dump formatter may not produce any formatted output for the job if this was the first EXCI request for this VSE task.

Module: DFHXCEIP

0413

Explanation: The external CICS interface dump module DFHXCEIP was processing an EXCI EXEC API request and detected that the EXEC parameter list passed to it did not require the mandatory RETCODE parameter in which return codes are returned to the application.

An attempt to format the system dump produced with the CICS INFOANA dump formatter may not produce any formatted output for the job if this was the first EXCI request for this VSE task.

System Action: The application is abnormally terminated with a dump.

User Response: This error indicates the parameter list being passed to the EXCI has not been generated by the CICS translator. The translator should always be used. Correct the application to specify RETCODE.

Module: DFHXCEIP

0414

Explanation: The external CICS interface module DFHXCEIP issued a VSE/ESA ESTAEX macro to establish a recovery environment but a nonzero return code was returned from VSE/ESA.

System Action: The application terminates abnormally with a dump.

User Response: Examine the dump and any associated VSE/ESA messages to determine why the VSE/ESA ESTAEX request failed.

An attempt to format the system dump produced with the CICS INFOANA dump formatter may not produce any formatted output for the job if this was the first EXCI request for this VSE task.

Module: DFHXCEIP

0415

Explanation: The external CICS interface module DFHXCEIP detected an error early in EXCI initialization before EXCI dump services were available. DFHXCEIP issues abend 0415 so that its ESTAEX routine is invoked from where an system dump is taken instead to capture the error.

System Action: The application terminates abnormally with a dump.

User Response: Examine the system dump to determine the cause of the earlier reported error.

An attempt to format the system dump produced with the CICS INFOANA dump formatter does not produce any formatted output for the job because the error occurred too early in EXCI initialization for there to be any control blocks.

Module: DFHXCEIP

The EXCI service trap, DFHXCTRA

A user-replaceable program, DFHXCTRA, is available for use under the guidance of IBM service personnel. It is the equivalent of DFHTRAP used in CICS. It is invoked every time the external CICS interface writes a trace entry.

DFHXCTRA can perform one or all of the following actions:

1. Request the external CICS interface to write a trace entry on its behalf
2. Instruct the external CICS interface to take an SDUMP
3. Instruct the external CICS interface to disable DFHXCTRA

The CICS-supplied sample version of DFHXCTRA performs all four of the above functions if it detects a trace entry that indicates that a FREEMAIN error occurred while trying to free an EXCI pipe control block.

The source for DFHXCTRA is supplied in PRD1.BASE. The parameter list passed to DFHXCTRA is defined in the copybook DFHXCTRD, which is supplied in PRD1.BASE. DFHXCTRD also defines all the external CICS interface trace points for use by DFHXCTRA.

EXCI trace entry points

<i>Table 10 (Page 1 of 8). External CICS interface trace entries</i>				
Point ID	Module	Lvl	Type	Data
EX 0001	DFHXCPRH	Exc	PIPE_ALREADY_OPEN	1 Caller's parameter list 2 Call type 3 Caller's user name 4 Return codes and message pointer 5 Pipe token
EX 0002	DFHXCPRH	Exc	PIPE_ALREADY_CLOSED	1 Caller's parameter list 2 Call type 3 Caller's user name 4 Return codes and message pointer 5 Pipe token
EX 0003	DFHXCPRH	Exc	VERIFY_BLOCK_FM_ERROR	1 Caller's parameter list 2 Call type 3 Caller's user name 4 Return codes and message pointer
EX 0005	DFHXCPRH	Exc	XCIPIP_FM_ERR	1 Caller's parameter list 2 Call type 3 Caller's user name 4 Return codes and message pointer 5 Pipe token
EX 0006	DFHXCPRH	Exc	IRP_IOAREA_FM_ERR	1 Caller's parameter list 2 Call type 3 Caller's user name 4 Return codes and message pointer
EX 0201	DFHXCPRH	Exc	NO_CICS_IRC_STARTED	1 Caller's parameter list 2 Call type

Table 10 (Page 2 of 8). External CICS interface trace entries

Point ID	Module	Lvl	Type	Data
EX 0202	DFHXCPRH	Exc	NO_PIPE	3 Caller's user name 4 Return codes and message pointer 1 Caller's parameter list 2 Call type 3 Caller's user name 4 Return codes and message pointer 5 Pipe token
EX 0203	DFHXCPRH	Exc	NO_CICS_ON_OPEN	6 Target CICS applid 1 Caller's parameter list 2 Call type 3 Caller's user name 4 Return codes and message pointer 5 Pipe token
EX 0204	DFHXCPRH	Exc	NO_CICS_ON_DPL_1	6 Target CICS applid 1 Caller's parameter list 2 Call type 3 Caller's user name 4 Return codes and message pointer 5 Pipe token
EX 0205	DFHXCPRH	Exc	NO_CICS_ON_DPL_2	6 Target CICS applid 1 Caller's parameter list 2 Call type 3 Caller's user name 4 Return codes and message pointer 5 Pipe token
EX 0206	DFHXCPRH	Exc	NO_CICS_ON_DPL_3	6 Target CICS applid 1 Caller's parameter list 2 Call type 3 Caller's user name 4 Return codes and message pointer 5 Pipe token
EX 0403	DFHXCPRH	Exc	INVALID_APPL_NAME	6 Target CICS applid 1 Caller's parameter list 2 Call type 3 Caller's user name 4 Return codes and message pointer
EX 0405	DFHXCPRH	Exc	PIPE_NOT_CLOSED	1 Caller's parameter list 2 Call type 3 Caller's user name 4 Return codes and message pointer 5 Pipe token
EX 0406	DFHXCPRH	Exc	PIPE_NOT_OPEN	1 Caller's parameter list 2 Call type 3 Caller's user name 4 Return codes and message pointer 5 Pipe token
EX 0407	DFHXCPRH	Exc	INVALID_USERID	1 Caller's parameter list 2 Call type 3 Caller's user name
EX 0408	DFHXCPRH	Exc	INVALID_UOWID	1 Caller's parameter list 2 Call type 3 Caller's user name 4 UOWID
EX 0409	DFHXCPRH	Exc	INVALID_TRANSID	1 Caller's parameter list 2 Call type 3 Caller's user name
EX 0414	DFHXCPRH	Exc	ABORT_RECEIVED	1 Caller's parameter list 2 Call type 3 Caller's user name 4 Target CICS applid
EX 0415	DFHXCPRH	Exc	INVALID_CONNECTION	5 Message to be returned 1 Caller's parameter list 2 Call type

Table 10 (Page 3 of 8). External CICS interface trace entries

Point ID	Module	Lvl	Type	Data
EX 0416	DFHXCPRH	Exc	INVALID_CICS_RELEASE	3 Caller's user name 4 Connection name 5 Target CICS applid 1 Caller's parameter list 2 Call type 3 Caller's user name
EX 0417	DFHXCPRH	Exc	PIPE_MUST_CLOSE	4 Target CICS applid 1 Caller's parameter list 2 Call type 3 Caller's user name
EX 0418	DFHXCPRH	Exc	INVALID_PIPE_TOKEN	4 Pipe token 1 Caller's parameter list 2 Call type 3 Caller's user name
EX 0422	DFHXCPRH	Exc	SERVER_ABENDED	4 Pipe token 1 Caller's parameter list 2 Call type 3 Caller's user name
EX 0603	DFHXCPRH	Exc	XCUSER_GM_ERROR	4 Return codes and message pointer 5 DPL return area 1 Caller's parameter list 2 Call type 3 Caller's user name
EX 0604	DFHXCPRH	Exc	XCPIPE_GM_ERROR	4 Return codes and message pointer 1 Caller's parameter list 2 Call type 3 Caller's user name
EX 0607	DFHXCPRH	Exc	SVC_CALL_FAILED	4 Return codes and message pointer 1 Caller's parameter list 2 Call type 3 Caller's user name
EX 0608	DFHXCPRH	Exc	IRP_LOGON_FAILURE	4 Return codes and message pointer 1 Caller's parameter list 2 Call type 3 Caller's user name
EX 0609	DFHXCPRH	Exc	IRP_CONNECT_FAIL	4 Return codes and message pointer 5 Target CICS applid 6 Logon name 1 Caller's parameter list 2 Call type 3 Caller's user name
EX 0610	DFHXCPRH	Exc	IRP_DISC_FAIL	4 Return codes and message pointer 5 Target CICS applid 6 Pipe token 1 Caller's parameter list 2 Call type 3 Caller's user name
EX 0611	DFHXCPRH	Exc	IRP_LOGOFF_FAILED	4 Return codes and message pointer 5 Target CICS applid 6 Pipe token 1 Caller's parameter list 2 Call type 3 Caller's user name
EX 0612	DFHXCPRH	Exc	TRANSFORM_1_ERROR	4 Return codes and message pointer 5 Target CICS applid 6 Pipe token 1 Caller's parameter list 2 Call type 3 Caller's user name
EX 0613	DFHXCPRH	Exc	TRANSFORM_4_ERROR	4 Return codes and message pointer 1 Caller's parameter list 2 Call type 3 Caller's user name

Table 10 (Page 4 of 8). External CICS interface trace entries

Point ID	Module	Lvl	Type	Data
EX 0614	DFHXCPRH	Exc	IRP_NULL_DATA	4 Return codes and message pointer 1 Caller's parameter list 2 Call type 3 Caller's user name
EX 0615	DFHXCPRH	Exc	IRP_NEG_RESPONSE	4 Return codes and message pointer 5 Target CICS applid 1 Caller's parameter list 2 Call type 3 Caller's user name
EX 0616	DFHXCPRH	Exc	IRP_SWITCH_PULL_ERR	4 Return codes and message pointer 5 Target CICS applid 1 Caller's parameter list 2 Call type 3 Caller's user name
EX 0617	DFHXCPRH	Exc	IRP_IOAREA_GM_ERR	4 Return codes and message pointer 5 Target CICS applid 6 Pipe token 1 Caller's parameter list 2 Call type 3 Caller's user name
EX 0619	DFHXCPRH	Exc	IRP_BAD_IOAREA	4 Return codes and message pointer 1 Caller's parameter list 2 Call type 3 Caller's user name
EX 0620	DFHXCPRH	Exc	IRP_PROTOCOL_ERR	4 IOAREA address 1 Caller's parameter list 2 Call type 3 Caller's user name 4 Target CICS applid 5 Pipe token
EX 0621	DFHXCPRH	Exc	PIPE_RECOVERY_FAILURE	1 Caller's parameter list 2 Call type 3 Caller's user name 4 Target CICS applid 5 Pipe token
EX 0622	DFHXCPRH	Exc	ESTAEX_SETUP_FAIL	1 Caller's parameter list 2 Call type 3 Caller's user name
EX 0623	DFHXCPRH	Exc	ESTAEX_INVOKED	4 Return codes and message pointer 1 Caller's parameter list 2 Call type 3 Caller's user name
EX 0624	DFHXCPRH	Exc	TIMEDOUT	4 Return codes and message pointer 5 OS/390 abend code (see "OS/390 API Abend Codes" in section "VSE/Advanced Functions & SVC Errors" of VSE/ESA Messages and Codes - Volume 1) 1 Caller's parameter list 2 Call type 3 Caller's user name
EX 0625	DFHXCPRH	Exc	STIMER_SETUP_FAIL	4 Server program name 5 Target CICS applid 1 Caller's parameter list 2 Call type 3 Caller's user name
EX 0626	DFHXCPRH	Exc	STIMER_CANCEL_FAIL	4 Return codes and message pointer 1 Caller's parameter list 2 Call type 3 Caller's user name
EX 0627	DFHXCPRH	Exc	INCORRECT_SVC_LEVEL	4 Return codes and message pointer 1 Caller's parameter list 2 Call type 3 Caller's user name

Table 10 (Page 5 of 8). External CICS interface trace entries

Point ID	Module	Lvl	Type	Data
EX 0800	DFHXCPRH	Exc	RESP shows LENGERR	<ul style="list-style-type: none"> 4 SVC instruction 1 Caller's parameter list 2 Call type 3 Caller's user name 4 Return codes and message pointer 5 COMMAREA length 6 Data length
EX 0801	DFHXCPRH	Exc	RESP shows INVREQ	<ul style="list-style-type: none"> 1 Caller's parameter list 2 Call type 3 Caller's user name 4 Return codes and message pointer 5 DPL options specified
EX 0802	DFHXCPRH	Exc	RESP shows PGMIDERR	<ul style="list-style-type: none"> 1 Caller's parameter list 2 Call type 3 Caller's user name 4 Program name 5 Target CICS applid
EX 0803	DFHXCPRH	Exc	RESP shows ROLLEDBACK	<ul style="list-style-type: none"> 1 Caller's parameter list 2 Call type 3 Caller's user name 4 Program name 5 Target CICS applid
EX 0804	DFHXCPRH	Exc	RESP shows NOTAUTH	<ul style="list-style-type: none"> 1 Caller's parameter list 2 Call type 3 Caller's user name 4 Program name 5 Target CICS applid
EX 0805	DFHXCPRH	Exc	RESP shows SYSIDERR	<ul style="list-style-type: none"> 1 Caller's parameter list 2 Call type 3 Caller's user name 4 Program name 5 Target CICS applid
EX 0806	DFHXCPRH	Exc	RESP shows TERMERR	<ul style="list-style-type: none"> 6 DPL_Retarea 1 Caller's parameter list 2 Call type 3 Caller's user name 4 Program name 5 Target CICS applid
EX 0904	DFHXCTRP	Exc	Overlength trace data field	1 XCTRP parameter list
EX 0905	DFHXCTRA	Exc	DFHXCTRA trace entry	1 User-specified data
EX 1000	DFHXCPRH	EX 1	Entry	For INIT_USER commands: <ul style="list-style-type: none"> 1 Caller's parameter list 2 Call type 3 Caller's user name 4 Caller's register 14 For ALLOCATE_PIPE requests: <ul style="list-style-type: none"> 1 Caller's parameter list 2 Call type 3 Caller's user name 4 CICS name 5 ALLOCATE options 6 Caller's register 14 For OPEN, CLOSE, and DEALLOCATE requests: <ul style="list-style-type: none"> 1 Caller's parameter list 2 Call type 3 Caller's user name 4 CICS name 5 Pipe token 6 Caller's register 14 For DPL requests: <ul style="list-style-type: none"> 1 Caller's parameter list 2 Call type

Table 10 (Page 6 of 8). External CICS interface trace entries

Point ID	Module	Lvl	Type	Data
EX 1001	DFHXCPRH	EX 1	Exit	<p>3 Caller's user name 4 CICS name 5 Pipe token 6 Program name 7 Caller's register 14 For INIT_USER, OPEN, CLOSE, and DEALLOCATE requests: 1 Caller's parameter list 2 Call type 3 Caller's user name 4 Return codes and message pointer 5 Caller's register 14 For ALLOCATE requests: 1 Caller's parameter list 2 Call type 3 Caller's user name 4 Return codes and message pointer 5 Pipe token 6 Caller's register 14 For DPL requests: 1 Caller's parameter list 2 Call type 3 Caller's user name 4 Return codes and message pointer 5 DPL return area 6 Caller's register 14</p>
EX 1010	DFHXCEIP	EX 1	Entry	<p>1 Program name 2 Target CICS applid 3 Transaction id 4 Caller's register 14 5 Up to first 100 bytes of commarea (if passed) 6 Commarea length, if commarea passed 7 Datalength, if commarea passed</p>
EX 1011	DFHXCEIP	EX 1	Exit	<p>1 EXEC retarea 2 Program name 3 Target CICS applid 4 Transaction id 5 Caller's register 14 6 Up to first 100 bytes of commarea (if passed) 7 Commarea length, if commarea passed</p>
EX 2000	DFHXCPRH	EX 2	IRP_LOGON	<p>1 Caller's parameter list 2 Call type 3 Caller's user name 4 Target CICS applid 5 IRP userid 6 SLCB address 7 Connection name</p>
EX 2001	DFHXCPRH	EX 2	IRP_CONN	<p>1 Caller's parameter list 2 Call type 3 Caller's user name 4 Target CICS applid 5 IRP userid 6 IRP threadid 7 SCCB address</p>
EX 2002	DFHXCPRH	EX 2	IRP_DISC	<p>1 Caller's parameter list 2 Call type 3 Caller's user name 4 Target CICS applid 5 Pipe token</p>

Table 10 (Page 7 of 8). External CICS interface trace entries

Point ID	Module	Lvl	Type	Data
EX 2003	DFHXCPRH	EX 2	IRP_LOGOFF	1 Caller's parameter list 2 Call type 3 Caller's user name 4 Pipe token 5 IRP userid
EX 2004	DFHXCPRH	EX 2	IRP_SWITCH	1 Caller's parameter list 2 Call type 3 Caller's user name 4 Target CICS applid 5 IRP userid 6 IRP threadid
EX 2005	DFHXCPRH	EX 2	IRP_SWITCH_DATA	1 User' appl name 2 Pipe token 3 Request header 4 Bind data 5 UOWID/USERID FMH 6 Transformed DPL request to CICS (up to 1000 bytes) 7 Final 1000 bytes of transformed DPL request
EX 2006	DFHXCPRH	EX 2	IRP_DATA	1 Caller's parameter list 2 Call type 3 Caller's user name 4 Target CICS applid 5 Length of data returned 6 Data (first 1000 bytes) 7 Data (final 1000 bytes)
EX 2007	DFHXCPRH	EX 2	PRE_URM	1 Caller's parameter list 2 Call type 3 Caller's user name 4 Parameters passed to DFHXCURM 5 URMINV, reason for calling URM 6 URM CICS, target CICS applid 7 URM ANCH, URM anchor point address
EX 2008	DFHXCPRH	EX 2	POST_URM	1 Caller's parameter list 2 Call type 3 Caller's user name 4 Parameters passed to DFHXCURM 5 URMINV, reason for calling URM 6 URM CICS, target CICS applid 7 URM ANCH, URM anchor point address
EX 3000	DFHXCEIP	Exc	ESTAEX_SETUP_ERROR	1 Return area (20 bytes) 2 VSE/ESA return code
EX 3001	DFHXCEIP	Exc	ESTAEX_INVOKED	1 Return area (20 bytes)
EX 3002	DFHXCEIP	Exc	INV_CTYPE_ON_INIT	1 Return area (20 bytes) 2 Call type
EX 3003	DFHXCEIP	Exc	INV_VNUM_ON_INIT	1 Return area (20 bytes) 2 Version number
EX 3004	DFHXCEIP	Exc	INV_APPL_NAME_ON_INIT	1 Return area (20 bytes) 2 User name
EX 3005	DFHXCEIP	Exc	INV_CTYPE_ON_ALLOC	1 Return area (20 bytes) 2 Call type
EX 3006	DFHXCEIP	Exc	INV_VNUM_ON_ALLOC	1 Return area (20 bytes) 2 Version number
EX 3007	DFHXCEIP	Exc	INV_UTOKEN_ON_ALLOC	1 Return area (20 bytes) 2 User token
EX 3008	DFHXCEIP	Exc	INV_CTYPE_ON_OPEN	1 Return area (20 bytes) 2 Call type
EX 3009	DFHXCEIP	Exc	INV_VNUM_ON_OPEN	1 Return area (20 bytes) 2 Version number
EX 3010	DFHXCEIP	Exc	INV_UTOKEN_ON_OPEN	1 Return area (20 bytes)

Table 10 (Page 8 of 8). External CICS interface trace entries

Point ID	Module	Lvl	Type	Data
EX 3011	DFHXCEIP	Exc	INV_PTOKEN_ON_OPEN	2 User token 1 Return area (20 bytes)
EX 3012	DFHXCEIP	Exc	INV_CTYPE_ON_DPL	2 Pipe token 1 Return area (20 bytes)
EX 3013	DFHXCEIP	Exc	INV_VNUM_ON_DPL	2 Call type 1 Return area (20 bytes)
EX 3014	DFHXCEIP	Exc	INV_UTOKEN_ON_DPL	2 Version number 1 Return area (20 bytes)
EX 3015	DFHXCEIP	Exc	INV_PTOKEN_ON_DPL	2 User token 1 Return area (20 bytes)
EX 3017	DFHXCEIP	Exc	INV_USERID_ON_DPL	2 Pipe token 1 Return area (20 bytes)
EX 3018	DFHXCEIP	Exc	PIPE_NOT_OPEN_ON_DPL	2 Userid 1 Return area (20 bytes)
EX 3019	DFHXCEIP	Exc	PIPE_MUST_CLOSE_ON_DPL	2 Pipe token 1 Return area (20 bytes)
EX 3020	DFHXCEIP	Exc	INV_CTYPE_ON_CLOSE	2 Pipe token 1 Return area (20 bytes)
EX 3021	DFHXCEIP	Exc	INV_VNUM_ON_CLOSE	2 Call type 1 Return area (20 bytes)
EX 3022	DFHXCEIP	Exc	INV_UTOKEN_ON_CLOSE	2 Version number 1 Return area (20 bytes)
EX 3023	DFHXCEIP	Exc	INV_PTOKEN_ON_CLOSE	2 User token 1 Return area (20 bytes)
EX 3024	DFHXCEIP	Exc	INV_CTYPE_ON_DEALL	2 Pipe token 1 Return area (20 bytes)
EX 3025	DFHXCEIP	Exc	INV_VNUM_ON_DEALL	2 Call type 1 Return area (20 bytes)
EX 3026	DFHXCEIP	Exc	INV_UTOKEN_ON_DEALL	2 Version number 1 Return area (20 bytes)
EX 3027	DFHXCEIP	Exc	INV_PTOKEN_ON_DEALL	2 User token 1 Return area (20 bytes)
EX 3028	DFHXCEIP	Exc	PIPE_NOT_CLOSED_ON_DEALL	2 Pipe token 1 Return area (20 bytes)
EX 3029	DFHXCEIP	Exc	XCEIP_RETRYING	2 Pipe token 1 Return area (20 bytes)

Chapter 10. Response and reason codes returned on EXCI calls

This chapter gives details of the reason codes for the responses returned on the EXCI call interface.

Note: All numeric response and reason code values shown are in decimal.

See also “OS/390 API Return Codes” in section “VSE/Advanced Functions & SVC Errors” of *VSE/ESA Messages and Codes - Volume 1* for OS/390 return codes.

Reason code for response:

OK

0 NORMAL

Explanation: Call completed normally.

Reason codes for response:

WARNING

1 PIPE_ALREADY_OPEN

Explanation: An Open_Pipe request has been issued for a pipe that is already open.

System Action: None. The pipe remains open.

User Response: If this response is unexpected, investigate whether an incorrect pipe token has been used on the Open_Pipe call.

2 PIPE_ALREADY_CLOSED

Explanation: A Close_Pipe request has been issued for a pipe that is already closed.

System Action: The external CICS interface ignores the request and the pipe remains closed.

User Response: If the response is unexpected, check that the Close_Pipe call is specifying the correct pipe token.

4 WS_FREEMAIN_ERROR

Explanation: An attempt to FREEMAIN working storage has resulted in a OS/390 FREEMAIN error.

System Action: The return code from the FREEMAIN is returned in the EXCI subreason field-1. The Initialize_User request continues unaffected.

User Response: If the problem persists, take a dump of the batch region and use the dump, together with the return code from the OS/390 FREEMAIN to determine why the FREEMAIN is failing.

5 XCPIPE_FREEMAIN_ERROR

Explanation: An attempt to FREEMAIN pipe storage has resulted in a OS/390 FREEMAIN error.

System Action: The return code from the FREEMAIN is returned in the EXCI subreason field-1. However, the external CICS interface continues processing the Deallocate_Pipe request. If the request fails with another error, this reason code is overwritten.

User Response: If the problem persists, take a dump of the client application program address space, and use the dump, with the return code from the OS/390 FREEMAIN to determine why the FREEMAIN is failing.

6 IRP_IOAREA_FM_FAILURE

Explanation: An attempt to FREEMAIN an MRO I/O area has resulted in a OS/390 FREEMAIN error.

System Action: The return code from the FREEMAIN is returned in the EXCI subreason field-1, but the DPL request continued to completion. Reason IRP_IOAREA_FM_FAILURE is returned to your application only if the DPL request completes, otherwise it is overwritten by subsequent response and reason codes.

User Response: If the problem persists, take a dump of the batch region and use it with the return code from the OS/390 FREEMAIN to determine why the FREEMAIN is failing.

7 SERVER_TERMINATED

Explanation: The CICS session, on which the server program has been executing, has been freed by CICS.

System Action: The CICS application server program has been detached at some point in its processing, and control is returned to the external CICS interface, which writes a trace entry for this error.

User Response: The most likely reason for this error is that the server program has caused CICS to terminate, perhaps by an EXEC CICS PERFORM SHUTDOWN command. During shutdown, CICS frees EXCI sessions so that shutdown can complete.

Reason codes for response:

RETRYABLE

Reviewers

We may be able to remove reason code 201

202 NO_PIPE

Explanation: An attempt has been made to open a pipe, but the target CICS system associated with the pipe has no free receive sessions.

System Action: The Open_pipe call fails, and the external CICS interface invokes the user-replaceable module, DFHXCURM.

User Response: This situation can occur even if the client application program has allocated (using Allocate_Pipe calls) no more pipes than the number of receive sessions defined on the target connection. This is because CICS can be in the process of cleaning up a pipe from a Close_Pipe request. For this reason, you are recommended to specify a larger RECEIVECOUNT value than is theoretically necessary when defining the SESSIONS resource definition to CICS. The application program can reissue the Open_Pipe request.

203 NO_CICS (on Open_Pipe call)

Explanation: An attempt has been made to open a pipe but the target CICS system is not available, or hasn't yet opened IRC, or the target connection is out of service.

System Action: The open pipe request fails, and the external CICS interface invokes the user-replaceable module, DFHXCURM.

User Response: If subreason field-1 is non-zero (the IRP response code (R15)), subreason field-2 contains the IRP reason code. For an explanation of the IRP return codes, see the interregion control blocks in the *CICS Data Areas* manual. The IRP return codes are in the DFHIRSDS copybook, listed under the heading IRC.

When you have corrected the problem, your client application program can reissue the Open_Pipe call. attempted.

203 NO_CICS (on DPL request)

Explanation: An attempt has been made to issue a DPL request but the target CICS system is not available, or the target connection is out-of-service.

System Action: The DPL request fails, and the external CICS interface invokes the user-replaceable module, DFHXCURM.

User Response: If subreason field-1 is non-zero (the IRP response code from R15), subreason field-2 contains the IRP reason code. For an explanation of the IRP return codes, see the interregion control blocks in the *CICS Data Areas* manual. The IRP return codes

are in the DFHIRSDS copybook, listed under the heading IRC. Having corrected the problem, you must close and re-open the pipe before attempting another DPL request.

Reason codes for response:

USER_ERROR

401 INVALID_CALL_TYPE

Explanation: An invalid *call-type* parameter value is specified on this EXCI request.

System Action: The request is rejected.

User Response: Check your EXCI client program and ensure the *call_type* parameter specifies the appropriate value for the EXCI call, as follows.

- 1 Initialize_User
- 2 Allocate_Pipe
- 3 Open_Pipe
- 4 Close_Pipe
- 5 Deallocate_Pipe
- 6 DPL

402 INVALID_VERSION_NUMBER

Explanation: The *version_number* parameter does not specify a value of 1.

System Action: The request is rejected.

User Response: Check the client application program and ensure that all EXCI calls specify the value of 1 for the version number.

403 INVALID_APPL_NAME

Explanation: The *user_name* parameter consists of all blank characters (X'40').

System Action: The call is rejected.

User Response: Change the application program to specify a valid, non-blank user name.

404 INVALID_USER_TOKEN

Explanation: The client application program has issued an EXCI request using a user token that is unknown to the external CICS interface.

System Action: The request is rejected.

User Response: The Initialize_User call returns a 4-byte token that must be used on *all* further requests for the user. Check the client application program and correct the error to ensure that the correct token is passed.

405 PIPE_NOT_CLOSED

Explanation: A Deallocate_Pipe request has been issued against a pipe that has not yet been closed.

System Action: The external CICS interface ignores the request and the pipe remains open.

User Response: Check the client application program, and ensure that the Deallocate_Pipe request is intended. If so, issue a Close_Pipe request for the pipe before issuing the Deallocate_Pipe request.

406 PIPE_NOT_OPEN

Explanation: A DPL call has been issued on a pipe that is not open.

System Action: The external CICS interface rejects the DPL request.

User Response: Check the client application program, and ensure that an Open_Pipe request is issued before using the pipe on a DPL request. If an Open_Pipe has been issued by the application program, check that it has not been closed inadvertently before all the DPL requests have been made.

407 INVALID_USERID

Explanation: A DPL request has been issued with a USERID parameter that consists of all blanks.

System Action: The DPL request is rejected.

User Response: Check the EXCI client program and ensure that the DPL request passes a valid USERID parameter. If you don't want to specify a userid, code the call parameter list with a null address for *userid*. If you pass a null address, the external CICS interface passes the userid under which the client application program is running (the batch region's userid).

408 INVALID_UOWID

Explanation: A DPL request has been issued with a *uowid* parameter that has invalid length fields.

System Action: The DPL request is rejected.

User Response: Check the client application program and ensure that the DPL request passes a valid *uowid* parameter. If you don't want to specify a unit of work id, code the call parameter list with a null address for *uowid*, in which case the external CICS interface generates a unit of work id for you.

409 INVALID_TRANSID

Explanation: A DPL request has been issued with a *transid* parameter that consists of all blanks.

System Action: The DPL request is rejected.

User Response: Check the client application program and ensure that the *transid* parameter is specified correctly or has not been overwritten in some way. If you don't want to specify your own *transid*, code the call parameter list with a null address for *transid*, in which case the external CICS interface uses the default CICS mirror transaction, CSMI.

410 DFHMEBM_LOAD_FAILED

Explanation: During Initialize_User processing, the external CICS interface attempted to load the main message module in preparation for issuing external CICS interface messages, and the load of this module failed.

System Action: The Initialize_User call is rejected. The return code from the OS/390 LOAD macro (R15) is returned in the subreason field-1. The external CICS interface handles the error, and returns the abend (R0) that would have occurred in the subreason field-2.

User Response: Using the OS/390 return code, determine why the load failed. The most likely reason is that the message module, DFHMEBM, is not in any library included in the LIBDEF PHASE,SEARCH library concatenation of the batch job. Ensure the PRD1.BASE library is included in the LIBDEF PHASE,SEARCH library concatenation, and restart the client application program.

411 DFHMET4E_LOAD_FAILED

Explanation: The load of message module, DFHMET4E, has failed. During Initialize_User processing, the external CICS interface attempted to load its message table in preparation for issuing messages. The load of this module failed.

System Action: The Initialize_User call is rejected. The return code from the OS/390 LOAD macro (R15) is returned in the subreason field-1. The external CICS interface handles the error, and returns the abend (R0) that would have occurred in the subreason field-2.

User Response: Using the OS/390 return code, determine why the load failed. The most likely reason is that the message table, DFHMET4E, is not in any library included in the LIBDEF PHASE,SEARCH library concatenation of the batch job. Ensure the PRD1.BASE library is included in the LIBDEF PHASE,SEARCH library concatenation, and restart the client application program.

412 DFHXCURM_LOAD_FAILED

Explanation: During Initialize_User processing, the external CICS interface attempted to load the user-replaceable module, DFHXCURM. The load of this module failed.

System Action: The Initialize_User call is rejected. The return code from the OS/390 LOAD macro (R15) is returned in the subreason field-1. The external CICS interface handles the error, and returns the abend (R0) that would have occurred in the subreason field-2.

User Response: Using the OS/390 reason code, determine why the load failed. The most likely reason is that module DFHXCURM is not in any library included in the LIBDEF PHASE,SEARCH library concatenation of the batch job. Ensure the library containing the module is included in the LIBDEF PHASE,SEARCH library concatenation, and restart the client application program.

413 DFHXCTRA_LOAD_FAILED

Explanation: During Initialize_User processing, the external CICS interface attempted to load the trap module (DFHXCTRA). The load of this module has failed.

System Action: The Initialize_User call is rejected. The return code from the OS/390 LOAD macro (R15) is returned in the subreason field-1. The external CICS interface handles the error, and returns the abend (R0) that would have occurred in the subreason field-2.

User Response: Using the OS/390 return code, determine why the load failed. The most likely reason is that DFHXCTRA is not in any library included in the LIBDEF PHASE,SEARCH library concatenation of the batch job. Ensure the library containing the module is included in the LIBDEF PHASE,SEARCH library concatenation, and restart the client application program.

414 IRP_ABORT_RECEIVED

Explanation: Whilst processing a DPL request, an error occurred in the CICS server region, resulting in an abort FMH7 flow being returned to the external CICS interface.

System Action: A message is returned to the client application program. This is the message that would have been issued to the terminal if the server program had been initiated from a terminal. A pointer to the message is returned to the client application program in the message pointer field of the EXCI return area. See the description of the EXCI return areas for the exact definition of the message format. The pipe is put into a "must close" state.

User Response: Use the message to determine the cause of the error. A typical example is where the

server transaction cannot be attached, either because is disabled, or it has not been defined, or because of a security failure. Correct the problem, close and reopen the pipe, and reissue the DPL request.

415 INVALID_CONNECTION_DEFN

Explanation: A DPL request has been rejected by CICS because the target connection is not defined for use by an external CICS client application program.

System Action: The DPL request is rejected and the pipe is put into a “must close” state.

User Response: The most likely reason for this is that the connection definition in the CICS server region has been defined incorrectly as a CICS-to-CICS MRO connection, instead of an EXCI connection. Ensure that PROTOCOL(EXCI) is specified on the appropriate CONNECTION and SESSIONS resource definitions. You must close and reopen the pipe before reissuing the DPL request.

416 INVALID_CICS_RELEASE

Explanation: A DPL request has been rejected by the target CICS server region because it doesn't recognize the request.

System Action: The DPL call is rejected and the pipe is put into a “must close” state.

User Response: The most likely reason for this is that the client application program has specified an incorrect target CICS server region that is not a CICS Transaction Server for VSE/ESA Release 1 region. CICS regions earlier than this do not recognize EXCI call requests, and reject them. Correct the problem, close and reopen the pipe and then reissue the DPL request.

417 PIPE_MUST_CLOSE

Explanation: A DPL request has been issued on a pipe that is in a “must close” state.

System Action: The DPL request is rejected.

User Response: Some EXCI errors are serious enough to require that the pipe be closed and reopened in order to restore the pipe to a point where it can be used for further DPL requests. Others, more minor errors, allow further calls without closing and reopening the pipe. A previous error on this pipe has been of the more serious variety and the pipe is now in a “must close” state. Close and reopen the pipe and reissue the DPL request.

418 INVALID_PIPE_TOKEN

Explanation: An Open_Pipe, Close_Pipe, Deallocate_Pipe, or DPL request has been issued, but the pipe token passed on the call is either not a valid pipe, or is not a valid pipe allocated for this user (that is, there is mismatch between the user token and the pipe token).

System Action: The call is rejected.

User Response: Ensure that the pipe token has not been overwritten and is being passed correctly on the call. Also ensure there is no mismatch between the user token and the pipe token.

419 CICS_AFCB_PRESENT

Explanation: An Initialize_User request has been issued on a VSE task that has already been used by CICS. The external CICS interface cannot share a VSE task with CICS, ensuring that a CICS application program cannot issue EXCI requests.

System Action: The Initialize_User request is rejected.

User Response: To use the external CICS interface, you must create a new VSE subtask (or daughter subtask), and issue the EXCI calls under that unique subtask.

420 DFHXCOPT_LOAD_FAILED

Explanation: During Initialize_User processing, the external CICS interface attempted to load its options module, DFHXCOPT. The load of this module failed.

System Action: The Initialize_User call is rejected. The return code from the OS/390 LOAD macro (R15) is returned in the subreason field-1. The external CICS interface handles the error, and returns the abend (R0) that would have occurred in the subreason field-2.

User Response: Using the OS/390 return code, determine why the load failed. The most likely reason is that DFHXCOPT is not in any library included in the LIBDEF PHASE,SEARCH library concatenation of the batch job. Correct the problem and restart the client application program.

422 SERVER_ABENDED

Explanation: Whilst processing a DPL request, the CICS server application program abended without handling the error.

System Action: The server application program is abended and backout out. The abend code is returned in the abend code field of the EXCI return area.

User Response: Determine why the server program abended and fix the problem.

Reason codes for response:

SYSTEM_ERROR

601 WS_GETMAIN_ERROR

Explanation: During Initialize_User processing, a GETMAIN for working storage failed.

System Action: Processing cannot continue without working storage, so the request is terminated. At this point the external CICS interface trace and dump services are not available to provide diagnostic information, therefore EXCI issues an OS/390 ABEND (U0408) to force a SYSDUMP. The return code from the OS/390 GETMAIN request is returned in the return area.

User Response: Locate the GETMAIN return code in the dump, and use this and the rest of the dump to determine why the GETMAIN failed. Possible reasons are:

- The ALLOC size specified for the partition is too small.
- The SIZE parameter is too large, restricting the amount of storage available for the OS/390 GETMAIN.

Correct the ALLOC or SIZE parameters and restart the client application.

602 XCGLOBAL_GETMAIN_ERROR

Explanation: During Initialize_User processing, a GETMAIN failed for a critical control block (XCGLOBAL).

System Action: Processing cannot continue without this control block, and the request is terminated. At this point the external CICS interface trace and dump services are not available to provide diagnostic information, therefore EXCI issues an OS/390 ABEND (U0403) to force a SYSDUMP. The return code from the OS/390 GETMAIN request is returned in the return area.

User Response: Locate the GETMAIN return code in the dump, and use this and the rest of the dump to determine why the GETMAIN failed. Possible reasons are:

- The ALLOC size specified for the partition is too small.
- The SIZE parameter is too large, restricting the amount of storage available for the OS/390 GETMAIN.

Correct the ALLOC or SIZE parameters and restart the client application.

603 XCUSER_GETMAIN_ERROR

Explanation: During Initialize_User processing, a GETMAIN request failed for the user control block (XCUSER).

System Action: Initialize_User processing is terminated. The return code from the GETMAIN is returned in subreason field-1 of the return area. The external CICS interface issues message DFHEX0003 and issues a OS/390 ABEND (0410) to force a SYSDUMP.

User Response: Use the return code from the GETMAIN, with the dump, to determine why the GETMAIN failed. Possible reasons are:

- The ALLOC size specified for the partition is too small.
- The SIZE parameter is too large, restricting the amount of storage available for the OS/390 GETMAIN.

Correct the ALLOC or SIZE parameters and restart the client application.

604 XCIPIPE_GETMAIN_ERROR

Explanation: During Allocate_Pipe processing, a GETMAIN request for the pipe control block (XCIPIPE) failed.

System Action: Allocate_Pipe processing is terminated. The return code from the GETMAIN is returned in subreason field-1 of the EXCI return area. The external CICS interface issues message DFHEX0003, and takes a system dump.

User Response: Use the return code from the GETMAIN, and the dump, to determine why the GETMAIN failed. Possible reasons are:

- The ALLOC size specified for the partition is too small.
- The SIZE parameter is too large, restricting the amount of storage available for the OS/390 GETMAIN.

Correct the ALLOC or SIZE parameters and restart the client application.

605 VERIFY_BLOCK_GM_ERROR

Explanation: During Initialize_User processing, a GETMAIN failed for an EXCI internal control block.

System Action: Initialize_User processing is terminated. The return code from the GETMAIN is returned in the subreason field-1 of the EXCI return area. This error occurs before EXCI dumping services are initialized. Therefore, EXCI issues an OS/390 ABEND (U0409) to force a SYSDUMP. The return code from the OS/390 GETMAIN request is returned in the return area.

User Response: Locate the GETMAIN return code in the dump, and use this and the rest of the dump to determine why the GETMAIN failed. Possible reasons are:

- The ALLOC size specified for the partition is too small.
- The SIZE parameter is too large, restricting the amount of storage available for the OS/390 GETMAIN.

Correct the ALLOC or SIZE parameters and restart the client application.

607 CICS_SVC_CALL_FAILURE

Explanation: During Initialize_User processing, a call to the CICS SVC failed.

System Action: The return code from the CICS SVC is returned in the subreason field-1 of the EXCI return area. This error occurs before the external CICS interface dump services are initialized, therefore EXCI issues an OS/390 user abend (0406) to force a SYSDUMP.

User Response: Contact your IBM support center for assistance, with the return code and the dump available.

608 IRC_LOGON_FAILURE

Explanation: During Allocate_Pipe processing, an attempt by the external CICS interface to LOGON to DFHIRP failed.

System Action: The two return codes returned from DFHIRP are returned in the subreason fields. The IRP response code (R15) is in subreason field-1, and the IRP reason code, if any, is given in subreason field-2. The Allocate_Pipe request fails.

User Response: For an explanation of the IRP return codes, see the interregion control blocks in the *CICS Data Areas* manual. The IRP return codes are in the DFHIRSDS copybook, listed under the heading IRC. Use the return codes to determine why the logon failed, or contact your IBM support personal with details of the failure.

609 IRC_CONNECT_FAILURE

Explanation: During Open_Pipe processing, an attempt to connect to the target CICS system failed.

System Action: The Open_Pipe request fails. The DFHIRP return code is returned in the EXCI subreason field-1.

User Response: For an explanation of the IRP return codes, see the interregion control blocks in the *CICS Data Areas* manual. The IRP return codes are in the DFHIRSDS copybook, listed under the heading IRC.

Use the return code to determine why the logon failed, and reissue the open pipe request.

Note: This error is not caused by the target CICS being unavailable, which is returned as a RETRYABLE condition (NO_CICS).

610 IRC_DISCONNECT_FAILURE

Explanation: During Close_Pipe processing, CICS issued a DFHIRP disconnect call to terminate the connection to CICS. This request has failed.

System Action: The call fails and the pipe is left open. The IRP return code (R15) and any IRP reason code (R0) are returned in the EXCI subreason field-1 and field-2 respectively. The external CICS interface takes a system dump.

Although the disconnect failed, it is possible that the pipe is still connected to CICS. However, all connections are automatically disconnected at the end of the batch program.

User Response: For an explanation of the IRP return codes, see the interregion control blocks in the *CICS Data Areas* manual. The IRP return codes are in the DFHIRSDS copybook, listed under the heading IRC. Use the return code and the dump to determine the cause of the error.

611 IRC_LOGOFF_FAILURE

Explanation: During Deallocate_Pipe processing, CICS issued a DFHIRP logoff call. This request failed.

System Action: The Deallocate_Pipe call fails and the pipe remains allocated. The IRP return code (R15) and any IRP reason code (R0) are returned in the EXCI subreason field-1 and field-2 respectively. The external CICS interface takes a system dump.

Note: Because it remains allocated, the pipe is available for further calls. Any storage associated with the pipe is not freed. However, this storage is freed at the end of the client application program.

User Response: For an explanation of the IRP return codes, see the interregion control blocks in the *CICS Data Areas* manual. The IRP return codes are in the DFHIRSDS copybook, listed under the heading IRC. Use the return code and the dump to determine the cause of the error.

612 TRANSFORM_1_ERROR

Explanation: During DPL processing, whilst processing the data in preparation for sending to CICS, an internal call to program DFHXFQ resulted in an error.

System Action: The DPL request is terminated.

User Response: The return code from the call is

returned in the EXCI subreason field-1, and the external CICS interface takes a system dump.

This is an external CICS interface error. Contact your IBM support center with details of the return code and the dump.

613 TRANSFORM_4_ERROR

Explanation: During DPL processing, whilst processing the data returned by the CICS server region, an internal call to module DFHXFQ resulted in an error.

System Action: The DPL request is terminated. Note that the server application program has executed. The return code from the call to DFHXFQ is returned in the EXCI subreason field-1. This return code corresponds to any EIBRCODE information that was available. The external CICS interface takes a system dump.

User Response: This is an external CICS interface error. Contact your IBM support center with details of the return code and the dump.

614 IRP_NULL_DATA_RECEIVED

Explanation: During DPL processing, a request has been sent to the target CICS and this target CICS has replied without returning any data.

System Action: The DPL processing is terminated and the external CICS interface takes a system dump.

User Response: This is an internal protocol error. Contact your IBM support center with details of the dump.

615 IRP_NEGATIVE_RESPONSE

Explanation: An internal protocol error has occurred whilst trying to communicate with the target CICS region.

System Action: The DPL request fails, the pipe is put into a “must close” state, and the external CICS interface takes a system dump.

User Response: This is an external CICS interface error. Keep the dump and contact your IBM support center.

Note: The pipe is in a “must close” state. Before attempting further calls, the pipe must first be closed and reopened.

616 IRP_SWITCH_PULL_FAILURE

Explanation: An internal protocol error has occurred whilst trying to communicate with the target CICS region.

System Action: The DPL request fails, the pipe is put into a “must close” state, and the external CICS interface takes a system dump. The IRP return code

(R15) and reason code if any (R0) are returned in the EXCI subreason field-1 and subreason field-2.

User Response: This is an external CICS interface error. Keep the dump and contact your IBM support center.

Note: The pipe is in a “must close” state, and before attempting further DPL calls, the pipe must first be closed and reopened.

617 IRP_IOAREA_GM_FAILURE

Explanation: During DPL processing, an OS/390 GETMAIN request for an internal control block failed.

System Action: The DPL request is terminated. The return code from the GETMAIN is returned in the EXCI subreason field-1.

Note: This error occurs whilst processing the data returned by CICS, after the server application program has completed execution. This error results in the pipe being put into a “must close” state.

User Response: Use the return code to determine why the GETMAIN failed. Possible reasons are:

- The ALLOC size specified for the partition is too small.
- The SIZE parameter is too large, restricting the amount of storage available for the OS/390 GETMAIN.

Correct the ALLOC or SIZE parameters and restart the client application.

619 IRP_BAD_IOAREA

Explanation: During a DPL request, an I/O area has been supplied to DFHIRP that could not be used.

System Action: The DPL request is terminated, the pipe is forced into a “must close” state, and the external CICS interface takes a system dump.

User Response: This is an external CICS interface error. Contact the IBM support center with details of the return code and the dump.

Note: The pipe is in a “must close” state after this error, and before attempting further calls must first be closed and reopened.

620 IRP_PROTOCOL_ERROR

Explanation: An internal protocol error has occurred whilst trying to communicate with the target CICS system.

System Action: The DPL request is terminated, the pipe is forced into a “must close” state, and the external CICS interface takes a system dump.

User Response: This is an external CICS interface error. Keep the dump and contact your IBM support center.

Note: The pipe is in a “must close” state after this error, and before attempting further calls must first be closed and reopened.

621 PIPE_RECOVERY_FAILURE

Explanation: An error has occurred during an open pipe request. The external CICS interface attempts to recover by disconnecting the pipe again. During this disconnection, further errors have occurred.

System Action: The Open_Pipe call is terminated and the pipe is placed in a “must close” state. The return code from DFHIRP is returned in the EXCI subreason field-1, and a system dump is taken.

User Response: For an explanation of the IRP return codes, see the interregion control blocks in the *CICS Data Areas* manual. The IRP return codes are in the DFHIRSDS copybook, listed under the heading IRC. Use the dump and IRP return codes to determine why the disconnect failed. You may also want to use the EXCI trace to determine the earlier error that caused the open pipe recovery routine to be invoked.

Note: The pipe is now in a “must close” state and if further calls are to be issued, the pipe must be closed and reopened again first.

622 ESTAEX_SETUP_FAILURE

Explanation: In order to protect itself from possible program checks the external CICS interface establishes an OS/390 ESTAEX. In this case, the ESTAEX macro has failed.

System Action: The call terminated, and the return code from the OS/390 ESTAEX command is returned in the EXCI subreason field-1. This error may occur before EXCI dump services are initialized, therefore an EXCI issues an OS/390 abend (U0402) to force a SYSDUMP.

User Response: Use the return code and the dump to determine why the ESTAEX command failed. This may be an internal EXCI error and if the problem persists, contact your IBM support center.

623 ESTAEX_INVOKED

Explanation: A program check is encountered during call processing, and the ESTAEX is invoked.

System Action: The program check is handled by the EXCI ESTAEX and an attempt is made to recover to a state that can support further EXCI calls. The OS/390 abend code is returned in the EXCI subreason field-1 of the return area. To aid further diagnosis, a SYSDUMP is taken.

User Response: Use the return code and the dump to determine why a program check occurred in the external CICS interface. The most likely reason for this is that the EXCI code abended whilst trying to access the client program's parameters. Use the EXCI trace to determine if any of the parameters might have caused this error. If this is not the case, this may be an error in the external CICS interface. Keep the dump and contact your IBM support center.

624 SERVER_TIMEDOUT

Explanation: A DPL request has been issued and the target server program has executed in the CICS server region. However, the server program has been executing for longer than the time-out value specified in the DFHXCOPT table.

System Action: The external CICS interface stops waiting for the server program to complete. Because the server program might complete some time after the time-out, and try to respond to the DPL call, the pipe is forced into a “must close” state.

User Response: Determine why the server application program timed out. Either there is a problem with the server program itself (for example, it might be in a loop), or the timeout value is too low.

625 STIMER_SETUP_FAILURE

Explanation: In order to provide a TIMEOUT mechanism, the external CICS interface issues an OS/390 STIMERM macro call. This call has failed.

System Action: The return code from the call is returned in the subreason field-1 of the EXCI return area. The DPL request is terminated and the external CICS interface takes a system dump. The pipe is placed in a “must close” state.

User Response: Use the OS/390 STIMERM return code and the dump to determine why the call failed. This could be an external CICS interface error. Contact your IBM support center with details of the dump.

Note: The pipe is in a “must close” state after this error, and before attempting further calls must first be closed and reopened.

626 STIMER_CANCEL_FAILURE

Explanation: On successful completion of a DPL request, the cancel of an STIMERM request issued to check the TIMEOUT value has failed with an error.

System Action: The return code from the STIMERM CANCEL is returned in the subreason field-1 of the EXCI return area. The pipe is placed in a “must close” state, and the external CICS interface takes a system dump.

User Response: Use the return code and the dump to determine why the OS/390 STIMERM CANCEL

command failed. This could be an external CICS interface error. Contact your IBM support center with details of the dump.

Note: The pipe is in a “must close” state after this error, and before attempting further calls must first be closed and reopened.

628 IRP_LEVEL_CHECK_FAILURE

Explanation: The release level of the module DFHIRP is not at the same, or higher, level than the release level of the external CICS interface.

System Action: The Allocate_pipe request is terminated. The IRP return code (R15) is returned in the EXCI subreason field-1, and the function level of DFHIRP being used is returned in the EXCI subreason field-2. Subreason field-2 is only meaningful if subreason field-1 is zero. The external CICS interface takes a system dump.

User Response: Check the level of the DFHIRP module installed in the SVA. Ensure that it is at least

the same as the external CICS interface. The installed level of DFHIRP must be the highest level of CICS or external CICS interface in use in the VSE/ESA image. For more details about installing DFHIRP, see the *CICS System Definition Guide*.

629 SERVER_PROTOCOL_ERROR

Explanation: A response to a DPL request has been returned by CICS but the external CICS interface does not understand the response.

System Action: The DPL request is terminated and the external CICS interface takes a system dump.

User Response: Use the dump to determine why the response was in error. The most likely reason for this is that the CICS application server program was not running under the control of a CICS mirror task. This can happen if the transaction definition named by the transid parameter on the DPL call does not specify DFHMIRS as the program name. This would cause unidentified responses being sent from the CICS server region.

Chapter 11. Messages and Codes

For details of all messages and abend codes for the external CICS interface, see the following manuals:

VSE/ESA Messages and Codes - Volume 1

VSE/ESA Messages and Codes - Volume 2

VSE/ESA Messages and Codes - Volume 3

Bibliography

CICS Transaction Server for VSE/ESA Release 1 library

Evaluation and planning	
<i>Release Guide</i>	GC33-1645
<i>Migration Guide</i>	GC33-1646
<i>Report Controller Planning Guide</i>	GC33-1941
General	
<i>Master Index</i>	SC33-1648
<i>Trace Entries</i>	SC34-5556
<i>User's Handbook</i>	SC34-5555
<i>Glossary (softcopy only)</i>	GC33-1649
Administration	
<i>System Definition Guide</i>	SC33-1651
<i>Customization Guide</i>	SC33-1652
<i>Resource Definition Guide</i>	SC33-1653
<i>Operations and Utilities Guide</i>	SC33-1654
<i>CICS-Supplied Transactions</i>	SC33-1655
Programming	
<i>Application Programming Guide</i>	SC33-1657
<i>Application Programming Reference</i>	SC33-1658
<i>Sample Applications Guide</i>	SC33-1713
<i>Application Migration Aid Guide</i>	SC33-1943
<i>System Programming Reference</i>	SC33-1659
<i>Distributed Transaction Programming Guide</i>	SC33-1661
<i>Front End Programming Interface User's Guide</i>	SC33-1662
Diagnosis	
<i>Problem Determination Guide</i>	GC33-1663
<i>Messages and Codes Vol 3 (softcopy only)</i>	SC33-6799
<i>Diagnosis Reference</i>	LY33-6085
<i>Data Areas</i>	LY33-6086
<i>Supplementary Data Areas</i>	LY33-6087
Communication	
<i>Intercommunication Guide</i>	SC33-1665
<i>CICS Family: Interproduct Communication</i>	SC33-0824
<i>CICS Family: Communicating from CICS on System/390</i>	SC33-1697
Special topics	
<i>Recovery and Restart Guide</i>	SC33-1666
<i>Performance Guide</i>	SC33-1667
<i>Shared Data Tables Guide</i>	SC33-1668
<i>Security Guide</i>	SC33-1942
<i>External CICS Interface</i>	SC33-1669
<i>XRF Guide</i>	SC33-1671
<i>Report Controller User's Guide</i>	GC33-1940
CICS Clients	
<i>CICS Clients: Administration</i>	SC33-1792
<i>CICS Universal Clients Version 3 for OS/2: Administration</i>	SC34-5450
<i>CICS Universal Clients Version 3 for Windows: Administration</i>	SC34-5449
<i>CICS Universal Clients Version 3 for AIX: Administration</i>	SC34-5348
<i>CICS Universal Clients Version 3 for Solaris: Administration</i>	SC34-5451
<i>CICS Family: OO programming in C++ for CICS Clients</i>	SC33-1923
<i>CICS Family: OO programming in BASIC for CICS Clients</i>	SC33-1671
<i>CICS Family: Client/Server Programming</i>	SC33-1435
<i>CICS Transaction Gateway Version 3: Administration</i>	SC34-5448

Books from VSE/ESA 2.4 base program libraries

VSE/ESA Version 2 Release 4

Book title	Order number
Administration	SC33-6705
Diagnosis Tools	SC33-6614
Extended Addressability	SC33-6621
Guide for Solving Problems	SC33-6710
Guide to System Functions	SC33-6711
Installation	SC33-6704
Licensed Program Specification	GC33-6700
Messages and Codes Volume 1	SC33-6796
Messages and Codes Volume 2	SC33-6798
Messages and Codes Volume 3	SC33-6799
Networking Support	SC33-6708
Operation	SC33-6706
Planning	SC33-6703
Programming and Workstation Guide	SC33-6709
System Control Statements	SC33-6713
System Macro Reference	SC33-6716
System Macro User's Guide	SC33-6715
System Upgrade and Service	SC33-6702
System Utilities	SC33-6717
TCP/IP User's Guide	SC33-6601
Turbo Dispatcher Guide and Reference	SC33-6797
Unattended Node Support	SC33-6712

High-Level Assembler Language (HLASM)

Book title	Order number
General Information	GC26-8261
Installation and Customization Guide	SC26-8263
Language Reference	SC26-8265
Programmer's Guide	SC26-8264

Language Environment for VSE/ESA (LE/VSE)

Book title	Order number
C Run-Time Library Reference	SC33-6689
C Run-Time Programming Guide	SC33-6688
Concepts Guide	GC33-6680
Debug Tool for VSE/ESA Fact Sheet	GC26-8925
Debug Tool for VSE/ESA Installation and Customization Guide	SC26-8798
Debug Tool for VSE/ESA User's Guide and Reference	SC26-8797
Debugging Guide and Run-Time Messages	SC33-6681
Diagnosis Guide	SC26-8060
Fact Sheet	GC33-6679
Installation and Customization Guide	SC33-6682
LE/VSE Enhancements	SC33-6778
Licensed Program Specification	GC33-6683
Programming Guide	SC33-6684
Programming Reference	SC33-6685
Run-Time Migration Guide	SC33-6687
Writing Interlanguage Communication Applications	SC33-6686

VSE/ICCF

Book title	Order number
Administration and Operations	SC33-6738
User's Guide	SC33-6739

VSE/POWER

Book title	Order number
Administration and Operation	SC33-6733
Application Programming	SC33-6736
Networking Guide	SC33-6735
Remote Job Entry User's Guide	SC33-6734

VSE/VSAM

Book title	Order number
Commands	SC33-6731
User's Guide and Application Programming	SC33-6732

VTAM for VSE/ESA

Book title	Order number
Customization	LY43-0063
Diagnosis	LY43-0065
Data Areas	LY43-0104
Messages and Codes	SC31-6493
Migration Guide	GC31-8072
Network Implementation Guide	SC31-6494
Operation	SC31-6495
Overview	GC31-8114
Programming	SC31-6496
Programming for LU6.2	SC31-6497
Release Guide	GC31-8090
Resource Definition Reference	SC31-6498

Books from VSE/ESA 2.4 optional program libraries

C for VSE/ESA (C/VSE)

Book title	Order number
C Run-Time Library Reference	SC33-6689
C Run-Time Programming Guide	SC33-6688
Diagnosis Guide	GC09-2426
Installation and Customization Guide	GC09-2422
Language Reference	SC09-2425
Licensed Program Specification	GC09-2421
Migration Guide	SC09-2423
User's Guide	SC09-2424

COBOL for VSE/ESA (COBOL/VSE)

Book title	Order number
Debug Tool for VSE/ESA Fact Sheet	GC26-8925
Debug Tool for VSE/ESA Installation and Customization Guide	SC26-8798
Debug Tool for VSE/ESA User's Guide and Reference	SC26-8797
Diagnosis Guide	SC26-8528
General Information	GC26-8068
Installation and Customization Guide	SC26-8071
Language Reference	SC26-8073
Licensed Program Specifications	GC26-8069
Migration Guide	GC26-8070
Migrating VSE Applications To Advanced COBOL	GC26-8349
Programming Guide	SC26-8072

DB2 Server for VSE

Book title	Order number
Application Programming	SC09-2393
Database Administration	GC09-2389
Installation	GC09-2391
Interactive SQL Guide and Reference	SC09-2410
Operation	SC09-2401
Overview	GC08-2386
System Administration	GC09-2406

DL/I VSE

Book title	Order number
Application and Database Design	SH24-5022
Application Programming: CALL and RQDLI Interface	SH12-5411
Application Programming: High-Level Programming Interface	SH24-5009
Database Administration	SH24-5011
Diagnostic Guide	SH24-5002
General Information	GH20-1246
Guide for New Users	SH24-5001
Interactive Resource Definition and Utilities	SH24-5029
Library Guide and Master Index	GH24-5008
Licensed Program Specifications	GH24-5031
Low-level Code and Continuity Check Feature	SH20-9046
Library Guide and Master Index	GH24-5008
Messages and Codes	SH12-5414
Recovery and Restart Guide	SH24-5030
Reference Summary: CALL Program Interface	SX24-5103
Reference Summary: System Programming	SX24-5104
Reference Summary: HLPI Interface	SX24-5120
Release Guide	SC33-6211

PL/I for VSE/ESA (PL/I VSE)

Book title	Order number
Compile Time Messages and Codes	SC26-8059
Debug Tool For VSE/ESA User's Guide and Reference	SC26-8797
Diagnosis Guide	SC26-8058
Installation and Customization Guide	SC26-8057
Language Reference	SC26-8054
Licensed Program Specifications	GC26-8055
Migration Guide	SC26-8056
Programming Guide	SC26-8053
Reference Summary	SX26-3836

Screen Definition Facility II (SDF II)

Book title	Order number
VSE Administrator's Guide	SH12-6311
VSE General Introduction	SH12-6315
VSE Primer for CICS/BMS Programs	SH12-6313
VSE Run-Time Services	SH12-6312

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. 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. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described 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:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply in the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore this statement may not apply to you.

This publication could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

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 IBM United Kingdom Laboratories, MP151, Hursley Park, Winchester, Hampshire, England, SO21 2JN. Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Programming License Agreement, or any equivalent agreement between us.

Trademarks and service marks

The following terms, used in this publication, are trademarks or service marks of IBM Corporation in the United States or other countries:

CICS
IBM
VSE/ESA

DATABASE 2
IMS
VTAM

DB2
Language Environment

Index

A

- abend codes 62
- abends 62
- addressing mode (AMODE)
 - assembler sample program 50
 - client program requirements 49
- allocate_opts, parameter of Allocate_Pipe command 12
- Allocate_Pipe command 11
 - invocation of DFHXCURM during 45
 - security check failure 57
- allocating a pipe 11
- AMODE (addressing mode)
 - See addressing mode (AMODE)
- application programming
 - commands 7
 - copybooks 27
 - DPL subset 2
 - exception conditions returned on LINK command 33
 - language considerations 49
 - RESP and RESP2 fields 33
 - restrictions for server programs 2
 - sample programs 50
 - stub 49
 - translation required for EXEC CICS LINK command 37
- applid, specifying on Allocate_Pipe command 11
- assembler
 - copybook 27
 - EXCI CALL interface 7
 - sample program 29, 50
- automatic retry of EXEC CICS LINK 34

B

- batch jobs, querying the status of 43
- benefits of external CICS interface 5
 - See also external CICS interface (EXCI)
- bind-time security 57

C

- C language
 - copybook 27
 - EXCI CALL interface 7
 - sample program 29, 50
 - special considerations for client program 49
- CALL interface
 - See external CICS interface (EXCI)
- call_type
 - parameter of Allocate_Pipe command 11

- call_type (*continued*)
 - parameter of Close_Pipe command 22
 - parameter of Deallocate_Pipe command 24
 - parameter of DPL_Request command 15
 - parameter of Initialize_User command 8
 - parameter of Open_Pipe command 13
- CEMT INQUIRE EXCI command 43
- CICS system definition file, EXCI sample definitions 52
- CICS_applid, parameter of Allocate_Pipe command 11
- client program
 - addressing mode 49
 - connection through DFHIRP 6
 - definition of 1
 - JCL needed
 - running an EXCI client 55
 - linking to server with EXEC CICS LINK 31
 - MRO logon and bind-time security 57
 - PL/I and C language considerations 49
 - sample job for starting 56
 - sample program 50
 - translating 37
 - use of multiple sessions 1
- Close_Pipe command 22
- closing a pipe 22
- COBOL II
 - copybook 27
 - example of EXCI DPL call 28
 - EXCI CALL interface 7
 - sample program 50
- codes, abend 62
- COMMAREA_len, parameter of DPL_Request command 16
- COMMAREA, parameter of DPL_Request command 16
- connecting an allocated pipe 13
- CONNECTION definition
 - CONNTYPE attribute 39
 - PROTOCOL attribute 40
- connection, generic
 - See generic connection
- connection, specific
 - See specific connection
- CONNTYPE attribute, CONNECTION definition 39
- copybooks for assembler, C language, COBOL, PL/I 27
- CSMI (CICS-supplied mirror transaction)
 - attached by CICS server 33
 - authorizing the link userid 58
 - default transid 17
 - security 58

D

data_len, parameter of DPL_Request command 16
Deallocate_Pipe command 24
deallocating a pipe 24
defining resources
 See resource definition
DFH\$AXCC, assembler sample program 29
 example of output 53
DFH\$DXCC, C sample program 29
DFH\$EXCI, sample server definitions 52
DFH\$FILA, sample file definitions 52
DFH\$PXCC, PL/I sample program 29
DFHIRP (interregion communication program)
 connection of client and server 6
 security checks performed by 57
DFHLIST, note about sample definitions 52
DFHXCIS, alias for DFHXCSTB stub 49
DFHXCOPT, options table 47
DFHXCPLD, return area and equate copybook for
 assembler 27
DFHXCPLH, return area and equate copybook for C
 language 27
DFHXCPLL, return area and equate copybook for
 PL/I 27
DFHXCPLP, return area and equate copybook for
 COBOL 27
DFHXCRCO, return code copybook for assembler 27
DFHXCRCR, return code copybook for C language 27
DFHXCRCCL, return code copybook for PL/I 27
DFHXCRCO, return code copybook for COBOL 27
DFHXCSTB, stub for client programs 49
DFHXCSTRA, EXCI service trap 65
DFHXCSTRD, parameter list 65
DFHXCURM, user-replaceable module 45
disconnecting a pipe 22
distributed program link (DPL)
 API subset for server programs 2
 example COBOL call without userid and uowid 28
 request program call 15
DPL (distributed program link)
 See distributed program link (DPL)
DPL_opts, parameter of DPL_Request command 19
DPL_Request call 15
dpl_retarea, parameter of DPL_Request command 19
dumps 61
 formatting 61

E

EQUATE copybooks 27
error codes
 See abend codes
 See exception conditions returned on LINK
 command
 See reason codes

error codes (*continued*)
 See RESP and RESP2 fields
 See response codes
ESM (external security manager)
 See external security manager (ESM)
exception conditions returned on LINK command 33
EXCI (external CICS interface)
 See external CICS interface (EXCI)
EXCI on CEMT INQUIRE command 43
EXEC CICS LINK command 31
 automatic retry 34
 choosing between EXEC CICS and CALL
 interface 2
 security checking 58
 translation 37
external CICS interface (EXCI) 1, 31
 benefits 5
 CALL interface 2, 7
 choosing between EXEC CICS and CALL
 interface 2
 return area 27
 syntax 7
CICS releases supported 6
compiling and link-editing client programs 49
defining connections 39
description of 1
inquiring on the state of connections 43
languages supported 7
messages 85
options table, DFHXCOPT 47
PL/I and C language considerations 49
problem determination 61
programming languages supported 7
reason codes 73
response codes 73
security 57
user-replaceable module (DFHXCURM) 45
external security manager (ESM) 57
 specifying userid on DPL_Request command 18

F

freeing storage associated with a pipe 24
function call EQUATE copybooks 27

G

generic connection
 definition of 40
 note about lack of security checks 57

I

Initialize_User command 8
interregion communication
 See DFHIRP (interregion communication program)

interregion communication (IRC)
opening after installation of sample definitions 52
interregion communication program (DFHIRP)
See DFHIRP (interregion communication program)

J

job control language (JCL)
for running an EXCI client program 55

L

languages
See application programming
LINK command 31
choosing between EXEC CICS and CALL
interface 2
link-editing
See also application programming
DFHXCOPT options table 47
for client program 49
translation required for EXEC CICS LINK
command 37
use of DFHXCSTB stub 7
logon security 57

M

messages 85
mirror transaction
See also CSMI (CICS-supplied mirror transaction)
authorizing the link userid 58
security 58
MSGCASE, parameter of DFHXCOPT 47
multiregion operation (MRO)
logon and bind time security 57

N

null parameters, example of EXCI CALLs with 28

O

Open_Pipe command 13
opening a pipe 13
options table, DFHXCOPT 47

P

parameters
null 28
specifying with options table 47
pgmname
parameter of DPL_Request command 15
pipe
See also pipe_token
allocating 11

pipe (*continued*)
closing 22
connecting 13
deallocating 24
definition of 1
disconnecting 22
freeing storage associated with 24
invocation of DFHXCURM during
ALLOCATE_PIPE 45
opening 13
restriction on leaving open 13
reusing a closed pipe 22
pipe_token
parameter of Allocate_Pipe command 11
parameter of Close_Pipe command 22
parameter of Deallocate_Pipe command 24
parameter of DPL_Request command 15
parameter of Open_Pipe command 14
PL/I
copybook 27
EXCI CALL interface 7
sample program 29, 50
special considerations for client program 49
problem determination 61
abends 62
dumps 61
service trap 65
trace 61
programming
See application programming
programming restrictions for server programs 2
PROTOCOL attribute
CONNECTION definition 40
SESSIONS definition 41

R

reason codes 73
Allocate_Pipe call 12
Close_Pipe call 22
Deallocate_Pipe call 25
DPL call 20
Intialize_User call 9
Open_Pipe call 14
RECEIVECOUNT attribute, SESSIONS definition 41
RECEIVEPFX attribute, SESSIONS definition 42
residence mode (RMODE)
assembler sample program 50
resource definition
CONNECTION definition 39
sample programs 52
SESSIONS definition 41
RESP and RESP2 fields 33
response codes 26, 73
Allocate_Pipe call 12
Close_Pipe call 22

response codes (*continued*)

- Deallocate_Pipe call 25
- DPL call 20
- Initialize_User call 9
- Open_Pipe call 14

retries on an EXEC CICS LINK command 34

return_area

- parameter of Allocate_Pipe command 11
- parameter of Close_Pipe command 22
- parameter of Deallocate_Pipe command 24
- parameter of DPL_Request command 15
- parameter of Initialize_User command 8
- parameter of Open_Pipe command 13

reusing a closed pipe 22

RMODE (residence mode)

- See residence mode (RMODE)

running the sample applications 52

S

sample programs 29, 50

- description 50

security 57

SENDCOUNT attribute, SESSIONS definition 42

SENDPFX attribute, SESSIONS definition 43

server program

- API restrictions 2
- connection through DFHIRP 6
- definition of 1
- DPL subset 2
- linking from client with EXEC CICS LINK 31
- programming restrictions 2
- sample program 50
- security considerations 58

service trap 65

session

- See pipe

SESSIONS definition

- PROTOCOL attribute 41
- RECEIVECOUNT attribute 41
- RECEIVEPFX attribute 42
- SENDCOUNT attribute 42
- SENDPFX attribute 43

shared virtual area (SVA), installation of DFHIRP 6

specific connection

- definition of 39
- MRO logon security checks 57

storage, freeing 24

stub for client programs

- DFHXCIS 49
- DFHXCSTB 49

SVA (shared virtual area), installation of DFHIRP 6

system definition

- See resource definition

system definition file (CSD), CICS 52

T

TIMEOUT, parameter of DFHXCOPT 47

trace 61

- TRACE parameter of DFHXCOPT 48
- trace points 65
- TRACESZE parameter of DFHXCOPT 48

transid, parameter of DPL_Request command 16

translation of EXEC CICS LINK command 37

trap, DFHXCTRA 65

- TRAP, parameter of DFHXCOPT 48

TYPE, parameter of DFHXCOPT 47

U

unit-of-work identifier, specifying on DPL_Request 17

uowid, parameter of DPL_Request 17

user environment, initializing 8

user security 58

user-replaceable module 45

- DFHXCURM 45

user_name, parameter of Initialize_User command 9

user_token

- parameter of Allocate_Pipe command 11
- parameter of Close_Pipe command 22
- parameter of Deallocate_Pipe command 24
- parameter of DPL_Request command 15
- parameter of Initialize_User command 8
- parameter of Open_Pipe command 13

userid, parameter of DPL_Request command 18

V

version_number

- parameter of Allocate_Pipe command 11
- parameter of Close_Pipe command 22
- parameter of Deallocate_Pipe command 24
- parameter of DPL_Request command 15
- parameter of Initialize_User command 8
- parameter of Open_Pipe command 13

VS COBOL II

- See COBOL II

W

writing programs

- See application programming

Sending your comments to IBM

CICS® Transaction Server for VSE/ESA™

External CICS Interface

SC33-1669-00

If you want to send to IBM any comments you have about this book, please use one of the methods listed below. Feel free to comment on anything you regard as a specific error or omission in the subject matter, and on the clarity, organization or completeness of the book itself.

To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

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

You can send your comments to IBM in any of the following ways:

- By mail:
 - IBM UK Laboratories
 - Information Development
 - Mail Point 095
 - Hursley Park
 - Winchester, SO21 2JN
 - England
- By fax:
 - From outside the U.K., after your international access code use 44 1962 870229
 - From within the U.K., use 01962 870229
- Electronically, use the appropriate network ID:
 - IBM Mail Exchange: GBIBM2Q9 at IBMMAIL
 - IBMLink: HURSLEY(IDRCF)
 - Email: idrcf@hursley.ibm.com

Whichever method you use, ensure that you include:

- The publication number and title
- The page number or topic to which your comment applies
- Your name and address/telephone number/fax number/network ID.



Program Number: 5648-054



Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC33-1669-00



Spine information:



CICS TS for VSE/ESA

External CICS Interface

Release 1