

CICS for MVS/ESA



External CICS Interface

Version 4 Release 1

CICS for MVS/ESA



External CICS Interface

Version 4 Release 1

Note!

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

Second edition (April 1997)

This edition applies to Version 4 Release 1 of the IBM licensed program Customer Information Control System/Enterprise Systems Architecture (CICS/ESA), program number 5655-018, and to all subsequent versions, releases, and modifications until otherwise indicated in new editions. Consult the latest edition of the applicable IBM system bibliography for current information on this product.

This is the second edition of the External CICS Interface for CICS/ESA 4.1. It is based on the first edition, SC33-1390-00, which is now obsolete. Changes from the first edition are marked by the '+' sign to the left of the changes.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

At the back of this publication is a page entitled "Sending your comments to IBM". If you want to make comments, but the methods described are not available to you, please address them to:

IBM United Kingdom Laboratories Limited, Information Development,
Mail Point 095, Hursley Park, Winchester, Hampshire, England, SO21 2JN.

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

© Copyright International Business Machines Corporation 1994, 1997. All rights reserved.

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

Contents

Notices	v
Programming interface information	v
Trademarks and service marks	vi
Preface	vii
What this book is about	vii
Who this book is for	vii
What you need to know to understand this book	vii
How to use this book	vii
Notes on terminology	vii
Determining if a publication is current	vii
Bibliography	ix
CICS/ESA 4.1 library	ix
Other CICS books	x
Summary of changes	xi
Changes for the second edition	xi
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
Chapter 3. The EXEC CICS interface	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	44
Chapter 5. The EXCI user-replaceable module	45
Chapter 6. External CICS interface options table, DFHXCPT	47
Chapter 7. Compiling and link-editing external CICS interface client programs	51
The external CICS interface stub, DFHXCSTB	51
The CICS-supplied procedures for the external CICS interface	51
Language considerations	52
Sample application programs	53
Job control language to run an EXCI client program	59
Chapter 8. Security	61
MRO logon and bind-time security	61
Link security	62
User security	63

+

Chapter 9. Problem determination	65
Trace	65
System dumps	66
MVS abends	67
04xx (external CICS interface) abend codes	67
The EXCI service trap, DFHXCTRA	70
EXCI trace entry points	70
Chapter 10. Response and reason codes returned on EXCI calls	79
Reason code for response: OK	79
Reason codes for response: WARNING	79
Reason codes for response: RETRYABLE	81
Reason codes for response: USER_ERROR	82
Reason codes for response: SYSTEM_ERROR	85
Chapter 11. Messages and Codes	91
Index	97

Notices

The following paragraph does not apply in any 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 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.

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

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 Laboratory Counsel, MP151, IBM United Kingdom Laboratories, 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.

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

Programming interface information

This book is intended to help you use the external CICS interface. This book primarily documents General-use Programming Interface and Associated Guidance Information provided by CICS.

General-use programming interfaces allow the customer to write programs that obtain the services of CICS.

However, this book also documents Product-sensitive Programming Interface and Associated Guidance Information and Diagnosis, Modification or Tuning Information provided by CICS.

Product-sensitive programming interfaces allow the customer installation to perform tasks such as diagnosing, modifying, monitoring, repairing, tailoring, or tuning of CICS. Use of such interfaces creates dependencies on the detailed design or implementation of the IBM software product. Product-sensitive programming

interfaces should be used only for these specialized purposes. Because of their dependencies on detailed design and implementation, it is to be expected that programs written to such interfaces may need to be changed in order to run with new product releases or versions, or as a result of service.

Product-sensitive Programming Interface and Associated Guidance Information is identified where it occurs, by an introductory statement to a chapter or section.

Diagnosis, Modification or Tuning Information is provided to help you diagnose problems associated with the external CICS interface.

Warning: Do not use this Diagnosis, Modification or Tuning Information as a programming interface.

Diagnosis, Modification or Tuning Information is identified where it occurs, by an introductory statement to a chapter or section.

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:

ACF/VTAM	CICS	CICS/ESA	CICS/MVS	CICS OS/2
CICS/VSE	C/370	DB2	IBM	MVS/ESA
OpenEdition	RACF	VTAM		

Preface

What this book is about

This book is about the external CICS interface—a programming interface that allows an MVS application program running in an MVS address space to link to a CICS application program that executes in a CICS address space.

It contains reference information to help you write an MVS 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 an MVS 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, VS COBOL II, 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 an MVS address space.

Notes on terminology

“CICS/ESA” is used for Customer Information Control System/Enterprise System Architecture.

The MVS/Enterprise System Architecture (MVS/ESA) operating system is generally referred to as “MVS.”

Determining if a publication is current

IBM regularly updates its publications with new and changed information. When first published, both hardcopy and BookManager softcopy versions of a publication are in step, but subsequent updates will probably be available in softcopy before they are available in hardcopy.

For CICS books, these softcopy updates appear regularly on the *Transaction Processing and Data Collection Kit* CD-ROM, SK2T-0730-xx. Each reissue of the collection kit is indicated by an updated order number suffix (the -xx part). For example, collection kit SK2T-0730-06 is more up-to-date than SK2T-0730-05. The collection kit is also clearly dated on the cover.

Here's how to determine if you are looking at the most current copy of a publication:

- A publication with a higher suffix number is more recent than one with a lower suffix number. For example, the publication with order number SC33-0667-02 is more recent than the publication with order number SC33-0667-01. (Note that suffix numbers are updated as a product moves from release to release, as well as for hardcopy updates within a given release.)
- When the softcopy version of a publication is updated for a new collection kit the order number it shares with the hardcopy version does not change. Also, the date in the edition notice remains that of the original publication. To compare softcopy with hardcopy, and softcopy with softcopy (on two editions of the collection kit, for example), check the last two characters of the publication's filename. The higher the number, the more recent the publication. For example, DFHPF104 is more recent than DFHPF103. Next to the publication titles in the CD-ROM booklet and the readme files, asterisks indicate publications that are new or changed.
- Updates to the softcopy are clearly marked by revision codes (usually a “#” character) to the left of the changes.

Bibliography

CICS/ESA 4.1 library

Evaluation and planning		
<i>Release Guide</i>	GC33-1161	April 1997
<i>Migration Guide</i>	GC33-1162	April 1997
General		
<i>CICS Family: Library Guide</i>	GC33-1226	April 1995
<i>Master Index</i>	SC33-1187	October 1994
<i>User's Handbook</i>	SX33-1188	April 1997
<i>Glossary (softcopy only)</i>	GC33-1189	n/a
Administration		
<i>Installation Guide</i>	GC33-1163	April 1997
<i>System Definition Guide</i>	SC33-1164	April 1997
<i>Customization Guide</i>	SC33-1165	April 1997
<i>Resource Definition Guide</i>	SC33-1166	April 1997
<i>Operations and Utilities Guide</i>	SC33-1167	April 1997
<i>CICS-Supplied Transactions</i>	SC33-1168	April 1997
Programming		
<i>Application Programming Guide</i>	SC33-1169	October 1994
<i>Application Programming Reference</i>	SC33-1170	April 1997
<i>System Programming Reference</i>	SC33-1171	April 1997
<i>Sample Applications Guide</i>	SC33-1173	October 1994
<i>Distributed Transaction Programming Guide</i>	SC33-1174	October 1994
<i>Front End Programming Interface User's Guide</i>	SC33-1175	October 1994
Diagnosis		
<i>Problem Determination Guide</i>	SC33-1176	October 1994
<i>Messages and Codes</i>	GC33-1177	April 1997
<i>Diagnosis Handbook</i>	LX33-6093	October 1994
<i>Diagnosis Reference</i>	LY33-6082	April 1997
<i>Data Areas</i>	LY33-6083	April 1997
<i>Supplementary Data Areas</i>	LY33-6081	October 1994
<i>Closely-Connected Program Interface</i>	LY33-6084	November 1996
Communication		
<i>Intercommunication Guide</i>	SC33-1181	April 1997
<i>Server Support for CICS Clients</i>	SC33-1591	February 1996
<i>CICS Family: Inter-product Communication</i>	SC33-0824	October 1996
<i>CICS Family: Communicating from CICS on System/390</i>	SC33-1697	October 1996
Special topics		
<i>Recovery and Restart Guide</i>	SC33-1182	October 1994
<i>Performance Guide</i>	SC33-1183	October 1994
<i>CICS-IMS Database Control Guide</i>	SC33-1184	October 1994
<i>CICS-RACF Security Guide</i>	SC33-1185	October 1994
<i>Shared Data Tables Guide</i>	SC33-1186	October 1994
<i>External CICS Interface</i>	SC33-1390	April 1997
<i>CICS ONC RPC Feature for MVS/ESA Guide</i>	SC33-1119	February 1996
<i>CICS Web Interface Guide</i>	SC33-1892	November 1996

The book that you are reading was republished in hardcopy format in April 1997 to incorporate updated information previously available only in softcopy. The right-hand column in the above table indicates the latest hardcopy editions of the CICS/ESA books available in April 1997. A book with a date earlier than April 1997 remains the current edition for CICS/ESA 4.1. Note that it is possible that other books in the library will be updated after April 1997.

When a new order is placed for the CICS/ESA 4.1 product, the books shipped with that order will be the latest hardcopy editions.

The style of IBM covers changes periodically. Books in this library have more than one style of cover.

For information about the softcopy books, see “Determining if a publication is current” on page vii. The softcopy books are regularly updated to include the latest information.

Other CICS books

- *CICS Application Migration Aid Guide*, SC33-0768
- *CICS Application Programming Primer (VS COBOL II)*, SC33-0674
- *CICS/ESA Facilities and Planning Guide* for CICS/ESA Version 3 Release 3, SC33-0654
- *CICS/ESA XRF Guide* for CICS/ESA Version 3 Release 3, SC33-0661
- *CICS Family: API Structure*, SC33-1007
- *CICS Family: General Information*, GC33-0155
- *IBM CICS Transaction Affinities Utility MVS/ESA*, SC33-1159

CICS Clients

- *CICS Clients: Administration*, SC33-1436
- *CICS Family: Client/Server Programming*, SC33-1435

Summary of changes

+ This book is a new edition of the *External CICS Interface* for CICS/ESA 4.1.

+ **Changes for the second edition**

+ This second edition contains changes made as a result of APARs and Reader's
+ Comment Forms. Changes made for this edition are indicated by the '+' symbol to
+ the left of the changes. Users of the first edition can therefore see what has
+ changed since the first edition was published. Softcopy versions of this book also
+ use this revision indicator and also use the '#' symbol to show further changes
+ since this second hardcopy edition of the book was published.

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 MVS to call a program (a server program) running in a CICS/ESA 4.1 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) facility of CICS interregion communication (IRC) facility supports these requests, and each pipe¹ maps onto one MRO session, with a limit of 25 pipes per EXCI address space.

Unless the CICS region is running in a sysplex under MVS/ESA 5.1 and therefore able to use cross-system MRO (XCF/MRO), the client program and the CICS server region (the region where the server program runs or is defined) must be in the same MVS image. Although the external CICS interface does not support the cross-memory access method, it can use the XCF access method provided by XCF/MRO in CICS/ESA 4.1. See the *CICS/ESA Intercommunication Guide* for information about XCF/MRO.

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

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

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 to a CICS system from non-CICS programs running under MVS/ESA

¹ pipe. A one-way 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.

- Issue DPL requests on these sessions from the non-CICS programs
- Close and deallocate the sessions on completion of the DPL requests.

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/ESA 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 53 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.

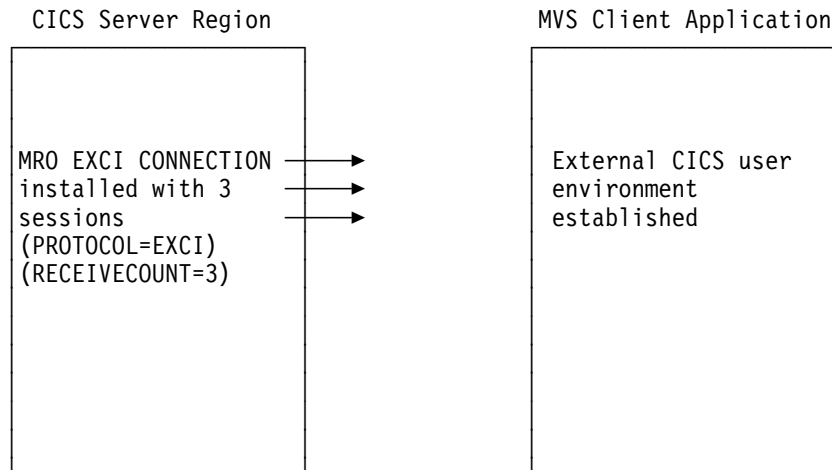


Figure 1. Stage 1: Status after an INITIALIZE_USER call

Notes to Figure 1:

1. In this diagram, the target CICS 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 address space is initialized with the EXCI user environment. There is no MRO activity at this stage, and no pipe exists.

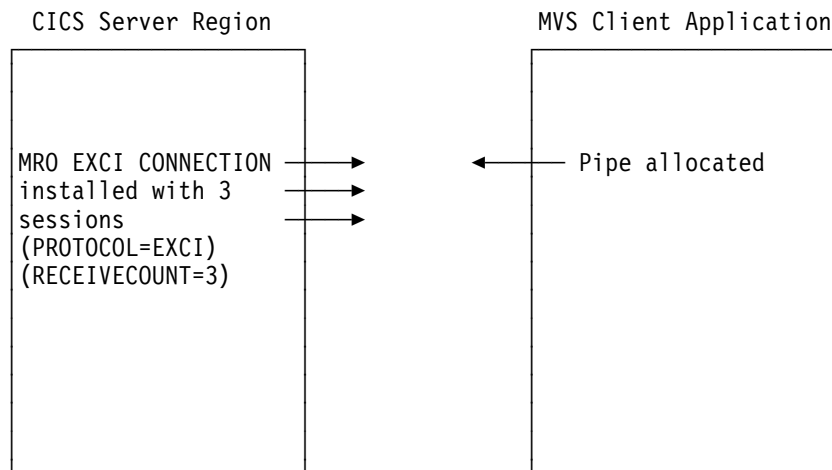


Figure 2. Stage 2: Status after the first ALLOCATE_PIPE call

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

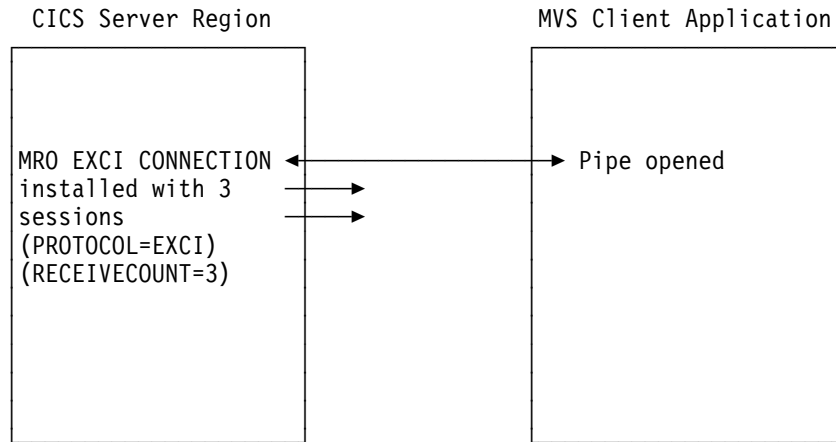


Figure 3. Stage 3: Status after the OPEN_PIPE call

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).

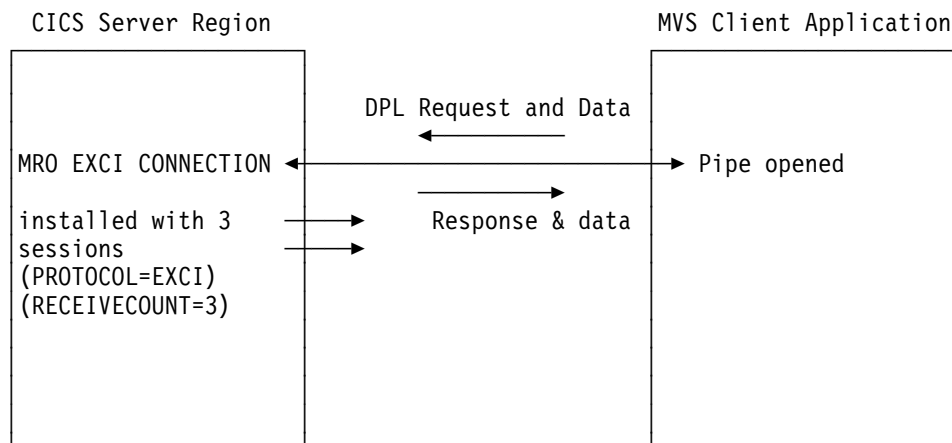


Figure 4. Stage 4: Status with one open pipe, processing a DPL call

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.

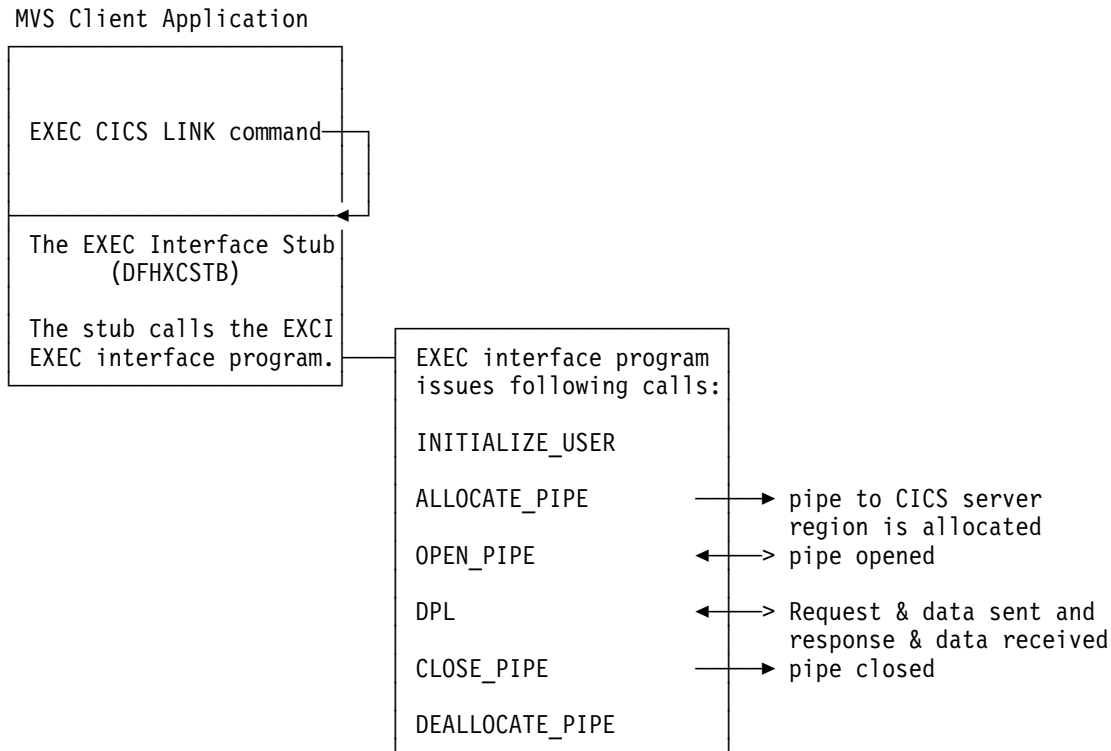


Figure 5. Illustration of the external CICS interface using the EXEC CICS command

Benefits of the external CICS interface

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

Programs running in MVS can issue an EXEC CICS LINK PROGRAM command to call a CICS application programs running in a CICS/ESA 4.1 region. Alternatively, the MVS programs can use the CALL interface when it is more appropriate to do so.

The provision of this programming interface means that, typically, MVS programs can:

- Update resources with integrity while CICS is accessing them.
- Take CICS resources offline, and back online, at the start and end of an MVS 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).

Distributed Computing Environment (DCE)

The external CICS interface opens up a new way to implement client/server applications, where the client program in a non-CICS environment calls a server program running in the CICS address space. The external CICS interface benefits not only TSO and batch applications, but allows you to extend the use of CICS application programs in an open client/server environment.

Although the CICS external interface operates over CICS MRO links, the client program can run on non-MVS platforms, and pass requests to CICS over an open system interface (OSI) using the IBM OpenEdition Distributed Computing Environment Application Support MVS/ESA CICS feature (OE DCE AS/CICS). In this way the external CICS interface provides an open interface to a wide variety of other application platforms.

Requirements for the external CICS interface

Client programs running in an MVS address space can communicate only with CICS server regions running under CICS/ESA 4.1 or a later, upward-compatible release. This is you need the parameters added to the MRO connection definition in CICS/ESA 4.1 to support the external CICS interface.

Also, the client program can connect to the server CICS region only through the CICS/ESA 4.1, or later, interregion communication program, DFHIRP.

For information about DFHIRP, and its requirement to be installed in the MVS extended link pack area (ELPA) see the *CICS/ESA Installation 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 system from non-CICS programs running under MVS/ESA, and
- 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 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, VS COBOL II, COBOL for MVS & VM, and PL/I programming languages, using the CALL conventions appropriate for the language.

There are examples of these CALLs, in all the supported languages, in the sample client programs provided by CICS. See "Sample application programs" on page 53 for information about these.

INITIALIZE_USER

Function

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

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 TCB. For example:

- A simple client program running under MVS can be a single user of the external CICS interface.
- A client program running under MVS 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 MVS can attach several TCBs, 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 TCB. Each is also a user, with a unique user name.

Syntax

```
▶▶—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 28 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 28.

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 DFHXCPLx (where x indicates the language) is INIT_USER. See page 28 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, then the value in *user_name* must match that of the NETNAME attribute of the CONNECTION definition for the specific pipe.

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

421 RUNNING_UNDER_AN_IRB

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 ESTAE_SETUP_FAILURE

INITIALIZE_USER

623 ESTAE_INVOKED

627 INCORRECT_SVC_LEVEL

For more information about response codes, see “Response code values” on page 27.

For information about the reason codes, see Chapter 10, “Response and reason codes returned on EXCI calls” on page 79.

ALLOCATE_PIPE

Function

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 address space.

Syntax

```
▶▶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 28 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 28.

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 28 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 system 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 URMAPPL parameter (DFHCURM is always invoked during Allocate_Pipe processing). You can also use the URMAPPL 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 URMAPPL parameter.

ALLOCATE_PIPE

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 29.

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 28 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
421 RUNNING_UNDER_AN_IRB

Response SYSTEM_ERROR:

604 XCPPIPE_GETMAIN_ERROR
608 IRC_LOGON_FAILURE
622 ESTAE_SETUP_FAILURE
623 ESTAE_INVOKED
628 IRP_LEVEL_CHECK_FAILURE

For information about response codes, see “Response code values” on page 27.

For information about the reason codes, see Chapter 10, “Response and reason codes returned on EXCI calls” on page 79.

OPEN_PIPE

Function

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 system. 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, step name, and procedure name of the client job that is using the session, and the MVSID of the MVS image on which the client program is running.

Syntax

```
▶▶—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 28 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 28.

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 28 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

421 RUNNING_UNDER_AN_IRB

Response SYSTEM_ERROR:

609 IRC_CONNECT_FAILURE

621 PIPE_RECOVERY_FAILURE

622 ESTAE_SETUP_FAILURE

623 ESTAE_INVOKED

For information about response codes, see "Response code values" on page 27.

For information about the reason codes, see Chapter 10, "Response and reason codes returned on EXCI calls" on page 79.

DPL_Request

Function

Issue a distributed program link request across an open pipe connected to the CICS system on which the server (or target) application program resides. The command is synchronous, and the TCB 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.

Syntax

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

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 28 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 28.

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 28 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 29.

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.

+ The DFHCICSA profile specifies the correct value for the INBFMH parameter,
+ which must be specified as INBFMH(ALL) for a mirror transaction.

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, CSML. 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 29.

uowid

An input area containing a unit-of-work identifier, using the APPC architected
format, that is passed on the DPL_Request for correlation purposes.

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 29.

If specified, the *uowid* parameter is passed to the CICS server region, which uses it as the 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 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. See the explanation about SYNCONRETURN on the *DPL_opts* parameter for more information about syncpoints in the CICS server region.

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 and BBBBBBBB are 1–8 byte names separated by a period
 - If AAAAAAAAA is omitted, the period must also be omitted
 - AAAAAAAAA and BBBBBBBB 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 MVSID, in characters, under which the client is running.
 - A 4-byte field containing the address space id (ASID) in which the MVS 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 RACF 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. See the *CICS/ESA CICS-RACF Security Guide* for information about link security on MRO connections.

See also Chapter 8, “Security” on page 61 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 an MVS batch job, the external CICS interface obtains and passes the *userid* specified on the *USER* parameter of the *JOB* statement.

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 29.

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:

Field 1 (fullword value)

This field is a fullword containing 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 CICS 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.

Field 2 :(fullword value)

This field is a fullword that 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 generally null
 + (CICS does not return RESP2 values across MRO links). However, if the
 + RESP field indicates SYSIDERR (value 53), this field provides a reason
 + code. See “Dpl_Retarea return codes” on page 29 for more information.

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/ESA 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.

Field 3 (4 bytes)

This field is a 4-character field containing:

- 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/ESA 4.1, 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 28 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.

+
+

#

Note that syncpointing and recovery in the CICS server take place independently of the client program. If the server program or the CICS server region fail for any reason, any updates of recoverable resources are backed out. Conversely, if CICS continues to execute, but the client program initiating the DPL request loses its connection to the CICS server (for example, because of a failure of the client region), CICS recoverable data that has been updated is committed. This is because updates of CICS recoverable resources are committed by any explicit syncpoint request issued by the server program and by the implicit syncpoint taken by the SYNCONRETURN function. The updates are committed before the transaction terminates with an AZI4 abend because of the lost connection, which is not detected until the syncpoints have been taken.

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
 421 RUNNING_UNDER_AN_IRB
 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 ESTAE_SETUP_FAILURE
 623 ESTAE_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 27.

DPL_Request

For information about the reason codes, see Chapter 10, “Response and reason codes returned on EXCI calls” on page 79.

CLOSE_PIPE

Function

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.

Syntax

```
▶▶—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 28 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 28.

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 28 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.

CLOSE_PIPE

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
421 RUNNING_UNDER_AN_IRB

Response SYSTEM_ERROR

610 IRC_DISCONNECT_FAILURE
622 ESTAE_SETUP_FAILURE
623 ESTAE_INVOKED

For information about response codes, see “Response code values” on page 27.

For information about the reason codes, see Chapter 10, “Response and reason codes returned on EXCI calls” on page 79.

DEALLOCATE_PIPE

Function

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.

Syntax

```
▶▶—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 28 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 28.

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 28 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

421 RUNNING_UNDER_AN_IRB

Response SYSTEM_ERROR:

611 IRC_LOGOFF_FAILURE

622 ESTAE_SETUP_FAILURE

623 ESTAE_INVOKED

For information about response codes, see "Response code values" on page 27.

For information about the reason codes, see Chapter 10, "Response and reason codes returned on EXCI calls" on page 79.

Response code values

The values that can be returned in the response field are shown in Table 1 (all values are specified in decimal):

Table 1. 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 an MVS re-IPL. Each reason code value for a RETRYABLE response explains 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 can be 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 2.

Copybook name	Language	Library
DFHXCPLD	Assembler	CICS410.SDFHMAC
DFHXCPLH	C	CICS410.SDFHC370
DFHXCPLD	COBOL	CICS410.SDFHCOB
DFHXCPLL	PL/I	CICS410.SDFHPL1

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 3.

Copybook name	Language	Library
DFHXCRCRCD	Assembler	CICS410.SDFHMAC
DFHXCRCRCH	C	CICS410.SDFHC370
DFHXCRCRCO	COBOL	CICS410.SDFHCOB
DFHXCRCRCL	PL/I	CICS410.SDFHPL1

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 4. 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

+ SYSDERR also can be returned on an EXCI DPL_Request, if the DPL_Request specifies a program defined in the CICS server region as a remote program, and the link between the server and the remote CICS region is not open. In this situation, SYSDERR is returned in the first word of the DPL_Retarea (return code 53). The reason code qualifying SYSDERR is placed in the second word of this area (the equivalent of a RESP2 value). For SYSDERR, the information stored in this field is derived from bytes 1 and 2 of the CICS EIBRCODE field. For example, if a remote link is not available, the EIBRCODE value stored in bytes 2 and 3 of the DPL_Retarea RESP2 field is X'0800'. For a list of the SYSDERR reason codes that can be returned in the RESP2 field, see the SYSDERR section of the notes on EIBRCODE in the *CICS/ESA 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 system. For example, suppose client program BATCH1 issues an EXCI DPL request to CICS A for program PROG1, which is defined as remote, and the request is function-shipped to CICS B where the program resides. If the session between CICS A and CICS B fails, or CICS B itself fails whilst executing the program PROG1, then TERMERR is returned to CICS A, 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.

<i>DPL parameter</i>	<i>COBOL variable</i>	
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_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  EXCI-RETURN-CODE  USER-TOKEN
                                DPL-REQUEST  PIPE-TOKEN  TARGET-PROGRAM
                                COMMAREA    COMM-LENGTH DATA-LENGTH
                                TARGET-TRANSID NULL-PTR   NULL-PTR
+   EXCI-DPL-RETAREA  SYNCONRETURN.
```

This example is taken from the CICS-supplied sample external CICS interface program, DFH0CXCC, which is supplied in CICS410.SDFHSAMP. 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.

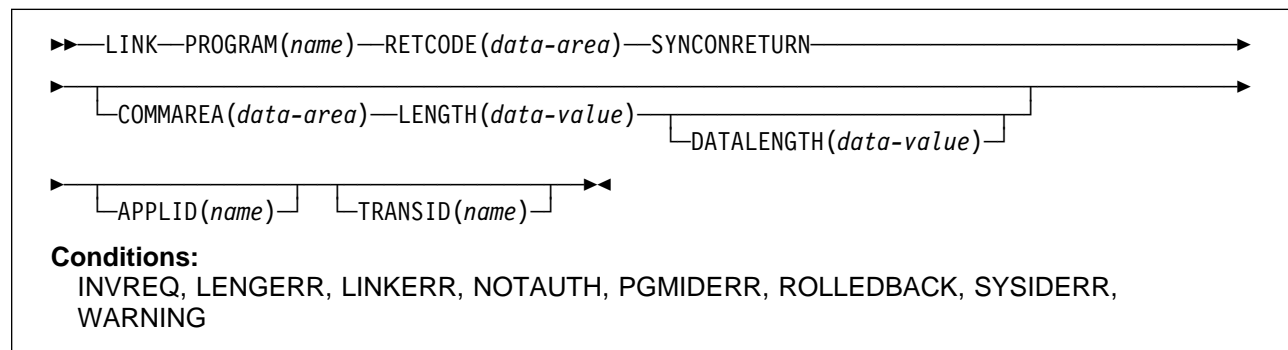
The external CICS interface provides this as a single, composite command, to invoke all the calls of the EXCI CALL interface. Each time you issue an EXEC CICS LINK PROGRAM command in a client application program, the external CICS interface invokes on your behalf each of the six EXCI calls.

EXEC CICS LINK command

Function

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

Syntax



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/ESA 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/ESA 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.

#

See the description of the *DPL_opts* parameter on the CALL interface for more information about syncpointing in the CICS application server region.

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 17.

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.

+

APAR PN85955

+

Documentation for PN85955 added on 12 November 1996

+

There are also two EXCI-specific exceptions, WARNING and LINKERR, that can be returned on the EXEC CICS LINK command. These exceptions, and their corresponding numeric values, are as follows:

+

+

+

WARNING	4
INVREQ	16
LENGERR	22
PGMIDERR	27
SYSIDERR	53
NOTAUTH	70
TERMERR	81
ROLLEDBACK	82
LINKERR	88

+

+ These exception condition codes are returned in the RESP field. The RESP value
+ can be tested using the CICS built-in function, DFHRESP, as shown in the
+ following example of the LINK command:

```
+ DCL LINK_RETAREA          CHAR(20);  
+ EXCI_EXEC_RETURN-CODE_PTR = ADDR(LINK_RETAREA);  
  
+ EXEC CICS LINK PROGRAM('CICSPROG')  
+       RETCODE(LINK_RETAREA);  
  
+ IF EXEC_RESP = DFHRESP(PGMIDERR) THEN ...
```

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 79 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 5 lists all the exception conditions and RESP2 values that are specific to the EXEC CICS LINK command for the external CICS interface.

Table 5 (Page 1 of 3). Exceptional conditions. RESP and RESP2 values returned from the EXEC API

Condition (RESP)	RESP2	Meaning
INVREQ (16)	21	SYNCONRETURN has not been specified
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
	421	A Close_Pipe or Deallocate_Pipe command has been issued under an IRB
	610	There has been a CICS IRP logoff failure on a Deallocate_Pipe call
LINKERR (88)	611	There has been a CICS IRC disconnect failure on a Close_Pipe call
	622	There has been an MVS ESTAE 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 ESTAE to be invoked
LINKERR (88)	201	Command has been issued on an MVS 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

Table 5 (Page 2 of 3). Exceptional conditions. RESP and RESP2 values returned from the EXEC API

Condition (RESP)	RESP2	Meaning
LINKERR (88)	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
	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/ESA 4.1 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 TCB that this command is attempting to use. This is not permitted and the command fails
	420	Load of DFHXCOPT failed
	421	The command has been issued under an MVS IRB, which is not permitted
	422	The server has abended
	601	A GETMAIN of working storage failed. This error leads to user abend 408
	602	A GETMAIN failed. This error leads to user abend 403
	603	A GETMAIN failed. This error leads to user abend 410
	604	A GETMAIN failed
	605	A GETMAIN for the VERIFY block failed. This error leads to user abend 409
	606	An SSI verify request (to obtain CICS SVC instruction) failed. This error leads to user abend 405

Table 5 (Page 3 of 3). Exceptional conditions. RESP and RESP2 values returned from the EXEC API

Condition (RESP)	RESP2	Meaning
LINKERR (88)	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
	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 ESTAE set-up command failed This error leads to user abend 402.
	623	XCPRH ESTAE invoked due to program check during the processing of this command. ESTAE attempts backout and takes a SYSMDUMP. 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	An MVS STIMERM macro call failed
	626	An MVS STIMERM 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 ESTAE set-up command failed	
904	The server program abended with the abend code in the ABCODE field of the RETCODE area	
905	An XCEIP ESTAE invoked	

See "Return codes" on page 28 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/ESA Application Programming Guide*.

For information about compiling and link-editing external CICS interface client programs, see page 51.

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 an MVS 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 TCB. For example:

- A simple client program running under MVS can be a single user of the external CICS interface.
- A client program running under MVS 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 MVS can attach several TCBs, 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 TCB. 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 must 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, you must 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. You must leave the PROTOCOL blank for MRO, and on the SESSIONS definition you must 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, you must 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.

+
+
+
+
+
+
+
Note: Although you can specify up to the 999 RECEIVECOUNT limit for an EXCI connection, IRP enforces a limit on the number of sessions (pipes) that any one EXCI address can use. An EXCI address space is limited to 25 pipes. When this limit is reached, IRP rejects further requests for a session with SYSTEM_ERROR, reason code 608. See the *CICS/ESA External CICS Interface* for more information.

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, you must 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.

You must 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 the SENDCOUNT in this system should equal RECEIVEDCOUNT 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, you must 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 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). There are two types of these batch jobs:

- Jobs that are sharing CICS local DL/I data through the CICS batch shared database interface.
- Jobs that are running a client application program that are using the external CICS interface to link to a server program in CICS.

To obtain this information about batch jobs linked to CICS through MRO, you use the CEMT INQUIRE IRBATCH 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.stepname.procname - mvsid
```

Either `stepname`, or `procname`, or both may not be present, indicated by the periods (.) being adjacent to one another.

The `mvsid` identifies the MVS system on which the job is running. If XCF/MRO is in use, the job can reside on a different MVS image to that on which CICS is running.

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/ESA 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 is supplied in CICS410.SDFHEXCI, and the source in CICS410.SDFHSAMP.

DFHXCURM is invoked in the non-CICS region during the processing of Allocate_Pipe commands, and after the occurrence of any re-tryable error. The re-tryable 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 MVS 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 re-tryable 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 MVS 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 MVS 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 DFHXCO 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 is supplied in CICS410.SDFHSAMP, and the load module is in CICS410.SDFHEXCI.

You assemble and link-edit the modified DFHXCOPT table into a suitable library in the STEPLIB concatenation of the job that runs the MVS 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 STEPLIB concatenation.

Table 6 shows the format of the DFHXCO macro and its parameters.

Table 6. The DFHXCO macro parameters		
	DFHXCO	TYPE={ CSECT DSECT} [,CICSSVC={0 number}] [,CONFDATA={SHOW HIDETC}] [,DURETRY={30 number-of-seconds}] [,GTF={OFF ON}] [,MSGCASE={MIXED UPPER}] [,TIMEOUT={0 number}] [,TRACE={OFF 1 2}] [,TRACESZE={16 number-of-kilobytes}] [,TRAP={OFF ON}]
	END	You must terminate 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.

CICSSVC={0|number}

Specifies the CICS type 3 SVC number being used for MRO communication.

The external CICS interface must use the same SVC number that is in use by the CICS MRO regions that reside in the MVS image in which the client program is running.

If you do not specify a specific CICS SVC number, the external CICS interface determines the SVC in use for MRO by means of an MVS VERIFY command.

0 Specify zero to indicate that the external CICS interface is to obtain the CICS SVC number from MVS. This is the default.

You should only specify 0 when you are sure that at least one CICS region has logged on to DFHIRP during the life of the MVS IPL.

number

Specify the CICS SVC number, in the range 200—255, that is in use for CICS interregion communications. This must be the SVC number that is installed in the MVS image in which the client program is running (the local MVS).

If no MRO CICS regions have ever logged on to DFHIRP in the local MVS during the life of the IPL, you must specify the SVC number. If you allow this parameter to default, and the external CICS interface requests the SVC from MVS, the request will fail if no CICS region has logged on to DFHIRP.

This parameter is required in those MVS images that do not run any CICS regions, and the client program is issuing DPL requests to a server CICS region that resides in another MVS. In these circumstances, the client program logs on to the local DFHIRP using the locally defined SVC, and communicates with the remote CICS region using XCF/MRO.

Note: All CICS regions using MRO within the same MVS image must use the highest level of both DFHIRP and the CICS SVC, DFHCSVC. If your MRO CICSplex consists of CICS regions at different release levels, the DFHIRP and DFHCSVC installed in the LPA must be from highest release level of CICS within the CICSplex.

MVS client programs using the external CICS interface can communicate only with server regions running under CICS/ESA 4.1 or later.

+ **CONFDATA={SHOW|HIDETC}**

+ Code this parameter to indicate whether the external CICS interface is to
+ suppress (hide) user data that might otherwise appear in EXCI trace entries
+ output to GTF or in EXCI dumps. This option applies to the tracing of the
+ COMMAREA flowing between the EXCI client program and the CICS server
+ program.

+ **SHOW**

+ Data suppression is not in effect. User data is traced.

+ **HIDETC**

+ This specifies that you want EXCI to 'hide' user COMMAREA data from
+ trace entries. Instead of the COMMAREA data, the trace entry contains a
+ character string stating that the data has been suppressed.

DURETRY={30|number-of-seconds|0}

Specifies the total time, in seconds, that the external CICS interface is to continue trying to obtain an MVS system dump using the SDUMP macro.

DURETRY allows you to control whether, and for how long, the external CICS interface is to reissue the SDUMP if another address space in the same MVS system is already taking an SDUMP when the external CICS interface issues an SDUMP request.

In the event of an SDUMP failure, the external CICS interface reacts as follows:

- If MVS is already taking an SDUMP for another address space, and the DURETRY parameter is nonzero, the external CICS interface issues an MVS STIMERM macro to wait for five seconds, before retrying the SDUMP macro. The external CICS interface issues a message to say that it will retry the SDUMP every five seconds until the DURETRY time limit.
- If the SDUMP fails for any other reason such as:
 - There are no SYS1.DUMP data sets available, or
 - There are I/O errors preventing completion of the dump, or
 - The DURETRY limit expires while retrying SDUMP
 the external CICS interface issues a message to inform you that the SDUMP has failed, giving the reason why.

30 30 seconds allows the external CICS interface to retry up to six times (once every five seconds).

number-of-seconds

Code the total number of seconds (up to 32767 seconds) during which you want the external CICS interface to continue retrying the SDUMP macro. The external CICS interface retries the SDUMP, once every five seconds, until successful or until retries have been made over a period equal to or greater than the DURETRY value.

0 Code a zero value if you do not want CICS to retry the SDUMP.

GTF={OFF|ON}

Specifies whether all trace entries normally written to the external CICS interface internal trace table are also to be written to an MVS generalized trace facility (GTF) data set (if GTF tracing is active).

OFF

Code this if trace entries are not to be written to GTF.

ON

Code this if trace entries are to be written to GTF.

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 the 16MB line. You should ensure that there is enough virtual storage for the trace table by specifying a large enough region size on the MVS REGION parameter.

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. Subpool 1 is used for the trace table storage, which exists for the duration of the jobstep TCB. 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 CICS410.SDFHEXCI library.

Note: The CICS410.SDFHEXCI also contains entries for DFHXCIE and DFHXCIS, which are aliases for DFHXCSTB.

To help you ensure that the stub is included, CICS provides a number of procedures, one for each language, which you can use for translating, compiling, and link-editing.

The required linkage editor modes

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

The CICS-supplied procedures for compiling and link-editing client programs include the following parameters on the PARM statement of the linkage editor job step:

```
LNKPARM=' AMODE(31),LIST,XREF'
```

The CICS-supplied procedures for the external CICS interface

CICS provides 7 procedures to enable you to translate, compile, and link-edit your client programs. Four of these are for use with specific language compilers or assembler, the other 3 being for use with Language Environment/370. These procedures, with the 4 language-specific procedures shown first, are:

DFHEXTAL The assembler procedure for assembler versions of client programs

DFHEXTDL The C procedure for C versions of client programs

DFHEXTPL The PL/I procedure for PL/I versions of client programs

DFHEXTVL The COBOL procedure for VS COBOL II versions of client programs.

- DFHYXTDL** The procedure for C versions of client programs running under Language Environment/370
- DFHYXTPL** The procedure for PL/I versions of client programs running under Language Environment/370
- DFHYXTVL** The procedure for VS COBOL II versions of client programs running under Language Environment/370.

To ensure that the EXCI stub is included with your client program, all these procedures include a step, COPYLINK, that unloads the stub into a temporary data set defined with a block length suitable for the linkage-editor. This temporary data set is then concatenated with the temporary data set containing your object program on the SYSLIN DD statement in the LKED step.

These procedures are supplied in the CICS410.SDFHPROC library. You are recommended to copy these to SYS1.PROCLIB or another suitable procedure library.

Language considerations

#

There are some language requirements that apply to writing an MVS client program that uses the external CICS interface. These affect programs written in PL/I and C. Also, for all languages, consider how you handle return codes before terminating your MVS client program.

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,OS)
```

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.

Setting the return code (R15) at termination

The external CICS interface does not clear register 15 at termination, regardless of
whether your client program executes normally or not. Therefore, even if your MVS
client program terminates normally after successfully using the external CICS
interface, the job step could end with an undefined return code.

To ensure a meaningful return code is given at termination, set the job step return
code before terminating your program. The sample client programs illustrate how
you can do this, using the saved response code from last call to the external CICS
interface. For example, the COBOL sample DFH0CXCC program moves
`SAVED-RESPONSE` to special register `RETURN-CODE` before terminating.

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 MVS 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 on the CICS/ESA 4.1 distribution tape.

The sample MVS client program is provided in assembler language, VS COBOL II, C/370, and PL/I. The sample CICS server program is provided in assembler only. Assembler language programs are in source and executable form. COBOL, PL/I, and C/370 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 7 on page 54, are supplied in source form in CICS410.SDFHSAMP. The sample assembler server program is also supplied in executable form in CICS410.SDFHLOAD. The assembler client program is supplied in CICS410.SDFHEXCI.

Note: The assembler version of the client program uses BSAM, 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 7. 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/370	DFH\$DXCC	Client program

The sample client programs show you how to code a simple MVS client application using the EXCI CALL interface and the EXEC CICS LINK command.

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 8.

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.

Using the COMMAREA in the sample programs

Data is passed between the sample client and server programs using a standard CICS communications area (COMMAREA) for passing data between programs. The definitions of this COMMAREA are identical on each side of the EXCI link to ensure that data is mapped correctly.

The sample client program minimizes data transmission by specifying a specific data length to avoid sending the whole COMMAREA over the link. The data length option specifies a binary value that is the length of a contiguous area of storage, from the start of the COMMAREA, to be passed to the server program. This is because the amount of data being passed to the server is small, but the COMMAREA itself is much larger so that the server can return the requested records from FILEA. Specifying a data length when the amount of data being passed is smaller than the COMMAREA improves performance.

The first and third sections of the sample client program define the COMMAREA as only 18 bytes (no data is requested from the server in these sections). For the second section, the sample client program defines the COMMAREA as 98 bytes, the extra 80 bytes being required for the sample server program to return a record from FILEA. In all sections, the data length is defined as 18 bytes. The COMMAREA structure is defined in the sample programs as follows:

Bytes	Data type	Field description
0-3	Fullword	Call type code.
4-11	Char(8)	Target file name.
12-17	Char(6)	Ridfield identifier.
18-97	Char(80)	FILEA record data area.

Note that, although the COMMAREA structure is described in both the client and server programs, the actual size of the COMMAREA made available by CICS to the server is determined by the client program. If you modify the sample programs to work with one of your own application programs, make sure you specify a COMMAREA large enough to handle the maximum amount of data the server is

expected to return to the client. The server must not attempt to return data that is
larger than the COMMAREA specified by the client.

For more information about using a COMMAREA for passing data between CICS
programs, see the *CICS/ESA Application Programming Guide*

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: Both the generic and specific connection definitions supplied in the sample group DFH\$EXCI specify 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 RACF, or an equivalent 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 connection resource definitions 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/370 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 60 as a basis for creating your own batch job to run the client program.

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

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

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 58.

If an error occurs while running the application, then, assuming the error is not severe, messages are written to the SYSPRINT output log displaying the reasons and/or return codes that cause processing to be aborted. Several examples of error-invoked output are shown in Figure 9, Figure 10, and Figure 11 on page 59.

```

***** 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 an MVS address space, for example, as a batch job.

You should note the following requirements when writing the JCL for your client program:

- You must include in the STEPLIB concatenation those libraries that contain the CICS-supplied external CICS interface modules and also the client program.

The external CICS interface modules are supplied in CICS410.SDFHEXCI. The contents of this library are as follows:

```
DFH$AXCC
DFHMEBM
DFHMET4E
DFHXCEIX
DFHXCIE   (alias of DFHXCSTB)
DFHXCIS   (alias of DFHXCTSB)
DFHXCOPT
DFHXCPRX
DFHXCSTB
DFHXCTRA
DFHXCURM
```

- You are recommended to include a DD statement for SYSMDUMP. The external CICS interface uses SYSMDUMP for some error conditions.
- The REGION parameter must specify a large enough region size to allow for the size of the internal trace table specified by the TRACESZE parameter in the DFHXCOPT options table.
- Include a SYSPRINT or equivalent DD statement for any output from the client program.

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

```
//EXCI    JOB (accounting_information),CLASS=A,TIME=1440,
//        USER=userid,PASSWORD=pswd,REGION=100M
//*****
//*      JCL to execute an external CICS interface client program  *
//*****
//        EXEC  PGM=pgmname
//STEPLIB DD  DSN=CICS410.EXCI.LOADLIB,DISP=SHR
//        DD   DSN=CICS410.SDFHEXCI,DSIP=SHR
//SYSPRINT DD  SYSOUT=A
//SYSMDUMP DD  DSN=SYS1.SYSMDP00,VOL=SER=valid,SPACE=(CYL,(1,1)),
//              DISP=OLD,UNIT=3390
```

Figure 12. Sample job for starting an EXCI client program

Notes:

1. The job userid, specified on the USER parameter, must be defined to RACF, or an equivalent external security manager (ESM).
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 61 for information about security in when using the external CICS interface.
3. See “Installing the EXCI sample definitions” on page 56 for information about modifying the sample connection definitions before you run the sample application programs in an environment that does not have RACF, or an equivalent external security manager (ESM), installed and active.

+
+
+
+

Chapter 8. Security

CICS applies security checks in a number of ways against requests received from an MVS client program. These 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

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 MVS 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 RACF. When you have defined the batch region's userid to RACF, 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 the DFHAPPL.*user_name* RACF FACILITY class profile(s), with UPDATE authority. The *user_name* part of the profile is the name defined on the INITIALIZE_USER command.

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

See "Defining DFHAPPL FACILITY class profiles for an EXCI region" on page 62 for information about FACILITY class profiles for an EXCI client program.

2. Bind-time authorization

Authorize the batch region's userid to the DFHAPPL.*applid* RACF FACILITY class profile of the target CICS server region, with READ authority.

Failure to authorize the batch region's userid to the CICS server region's DFHAPPL.*applid* profile causes Open_Pipe processing to fail with

```

#           RESPONSE(SYSTEM_ERROR) REASON(IRC_CONNECT_FAILURE). The
#           subreason field-1 for a bind-time security check failure returns decimal 176.

#           See the CICS/ESA CICS-RACF Security Guide for information about the MRO
#           logon and bind-time security checks, and for examples of how to define the RACF
#           DFHAPPL profiles.

# Defining DFHAPPL FACILITY class profiles for an EXCI region
#           Define the user_name part of the DFHAPPL profile name as follows:

#           • For the EXCI CALL interface, the user_name must be the name you specify on
#           the user_name parameter of the INITIALIZE_USER command.

#           Define FACILITY class profiles, with appropriate authorizations, for each user
#           name specified in a client program if the program has INITIALIZE_USER
#           commands for more than one user name.

#           For example, if the user_name defined on an INITIALIZE_USER command is
#           DCEUSER1, define the DFHAPPL profile in the FACILITY class as follows:

#           RDEFINE FACILITY (DFHAPPL.DCEUSER1) UACC(NONE)

#           If the batch region's userid is CLIENTA, authorize the batch region to logon to
#           IRP as follows:

#           PERMIT DFHAPPL.DCEUSER1 CLASS(FACILITY) ID(CLIENTA)
#           ACCESS(UPDATE)

#           • For the EXEC CICS LINK command, the user_name is preset by the external
#           CICS interface as DFHXCEIP. This does not require authorization for IRP
#           logon, because the EXEC CICS LINK interface uses a generic connection to
#           which the logon security check does not apply.

```

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.

See the *CICS/ESA CICS-RACF Security Guide* for information about MRO link 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 RACF, or an equivalent 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).

Further information

For more information about CICS security, see the *CICS/ESA CICS-RACF Security Guide*.

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
- MVS Abends
- The EXCI service trap, DFHXCTRA
- EXCI trace entry points

Details of the external CICS interface messages and abend codes are given in Chapter 11, "Messages and Codes" on page 91.

Trace

The external CICS interface writes trace data to two destinations: an internal trace table and an external MVS GTF data set. The internal trace table resides in the non-CICS MVS address space. 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 and/or an MVS GTF data set. They are listed in "EXCI trace entry points" on page 70.

To use GTF for external CICS interface tracing, GTF user tracing must be active, GTF must be started in the MVS image, and you must specify GTF=ON in the DFHXCOPT options table.

If you use GTF trace for both the CICS server region and the external CICS interface region, the trace entries are interleaved, which can help you with problem determination in the CICS-EXCI environment.

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

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

Formatting GTF trace

To format external CICS interface trace entries written to GTF, you can use the standard CICS DFHTR410 trace formatting routine.

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 MVS SYSMDUMPs for some error conditions and MVS SDUMPs for other, more serious conditions. These dumps contain all the external CICS interface control blocks, as well as trace entries.

Formatting system dumps

You can use the CICS IPCS verb 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 MVS 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 TCBs

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

Capturing SYSMDUMPs

To capture SYSMDUMPs produced by the external CICS interface, ensure you always include a DD statement for the SYSMDUMP data set in the client application program's JCL.

Using the MVS console for dumps

In addition to the dumps taken automatically by the external CICS interface, you can also force a dump of an address space running a client application program by means of an MVS dump command entered at the console. You can use the CICS IPCS verb exit routine DFHPD410 to format dumps taken in this way.

You can also use the console to dump the CICS server address space as well as the client address space, and use the CICS IPCS verb exit routine DFHPD410 to format both address space dumps together.

MVS abends

The following MVS abends can occur when you are running an external CICS interface job:

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 an MVS ESTAE macro to establish a recovery environment, but a nonzero return code was returned from MVS.

System Action: The application terminates abnormally with a dump.

User Response: Examine the dump and any associated MVS messages produced to determine why the MVS ESTAE 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 IPCS 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 an MVS GETMAIN request to obtain storage for its XCGLOBAL block, but a nonzero return code was returned from MVS.

System Action: Module DFHXCPRH issues an MVS abend with abend code 0403 which invokes its ESTAE routine to clear up its environment. A SYSMDUMP 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 MVS indicating why the GETMAIN failed. An application using the EXCI EXEC API receives RESP(LINKERR) RESP2(602).

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

An attempt to format the SYSMDUMP produced with the CICS IPCS 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 MVS SDUMP for an earlier reported problem. However the error has occurred too early in EXCI initialization for EXCI dump services to be available.

System Action: Module DFHXCPRH issues an MVS abend with abend code 0404 which invokes its ESTAE routine from which a SYSMDUMP is taken instead of an SDUMP to capture the earlier reported problem.

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

An attempt to format the SYSMDUMP produced with the CICS IPCS 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

0405

Explanation: The external CICS interface module DFHXCPRH issued an IEFSSREQ SSI verify request to MVS to determine the number of the CICS SVC type 3 SVC to use. The SSI VERIFY request failed.

System Action: Module DFHXCPRH issues an MVS abend with abend code 0405 which invokes its ESTAE routine to clear up its environment. A SYSMDUMP is taken before returning control to the application. An application using the EXCI CALL API receives RESPONSE(SYSTEM_ERROR) REASON(SSI_VERIFY_FAILED) in its return area. The subreason1 field of the return area contains the R15 return code from MVS indicating why the SSI

verify failed. An application using the EXCI EXEC API receives RESP(LINKERR) RESP2(606).

User Response: Use the MVS R15 return code obtained from the application or from the dump to determine why the SSI VERIFY request failed.

An attempt to format the SYSMDUMP produced with the CICS IPCS 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 an MVS abend with abend code 0406 which invokes its ESTAE routine to clear up its environment. A SYSMDUMP 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 MVS R15 return code obtained from the application or from the dump to determine why the CICS SVC call failed.

An attempt to format the SYSMDUMP produced with the CICS IPCS 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 an MVS abend with abend code 0407 which invokes its ESTAE routine to clear up its environment. A SYSMDUMP 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 SYSMDUMP produced with the CICS IPCS 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 an MVS GETMAIN request for its working storage but a nonzero return code was returned from MVS.

System Action: Module DFHXCPRH issues an MVS abend with abend code 0408 which invokes its ESTAE routine to clear up its environment. A SYSMDUMP 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 MVS indicating why the GETMAIN failed. An application using the EXCI EXEC API receives RESP(LINKERR) RESP2(601).

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

An attempt to format the SYSMDUMP produced with the CICS IPCS 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

0409

Explanation: The external CICS interface module DFHXCPRH issued an MVS GETMAIN request for storage required for its SSI VERIFY request, but a nonzero return code was returned from MVS.

System Action: Module DFHXCPRH issues an MVS abend with abend code 0409 which invokes its ESTAE routine to clear up its environment. A SYSMDUMP is taken before returning control to the application. An application using the EXCI CALL API receives RESPONSE(SYSTEM_ERROR) REASON(VERIFY_BLOCK_GM_ERROR) in its return area. The subreason1 field of the return area contains the R15 return code from MVS indicating why the GETMAIN failed. An application using the EXCI EXEC API receives RESP(LINKERR) RESP2(605).

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

An attempt to format the SYSMDUMP produced with the CICS IPCS 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 an MVS GETMAIN request for an XCUSER block but a nonzero return code was returned from MVS.

System Action: Module DFHXCPRH issues an MVS abend with abend code 0410 which invokes its ESTAE routine to clear up its environment. A SYSMDUMP 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 MVS indicating why the GETMAIN failed. An application using the EXCI EXEC API receives RESP(LINKERR) RESP2(603).

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

Module: DFHXCPRH

0411

Explanation: The external CICS interface dump module DFHXCDMP was attempting to call the CICS SVC in order for an MVS SDUMP to be taken to capture an earlier problem. DFHXCDMP was unable to call the SVC as no SVC number was available. DFHXCDMP issued an 0411 abend in order that the callers ESTAE routine is invoked which takes a SYSMDUMP instead.

System Action: A SYSMDUMP is taken instead of an SDUMP for an earlier reported problem.

User Response: Use the SYSMDUMP produced to diagnose the earlier reported problem.

An attempt to format the SYSMDUMP produced with the CICS IPCS 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: DFHXCDMP

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 SYSMDUMP produced with the CICS IPCS dump formatter may not produce any formatted output for the job if this was the first EXCI request for this TCB.

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 SYSMDUMP produced with the CICS IPCS dump formatter may not produce any formatted output for the job if this was the first EXCI request for this TCB.

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 an MVS ESTAE macro to establish a recovery environment but a nonzero return code was returned from MVS.

System Action: The application terminates abnormally with a dump.

User Response: Examine the dump and any associated MVS messages to determine why the MVS ESTAE request failed.

An attempt to format the SYSMDUMP produced with the CICS IPCS dump formatter may not produce any

formatted output for the job if this was the first EXCI request for this TCB.

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 ESTAE routine is invoked from where an SYSMDUMP is taken instead to capture the error.

System Action: The application terminates abnormally with a dump.

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

An attempt to format the SYSMDUMP produced with the CICS IPCS 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 skip writing the current trace entry to GTF
4. 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 CICS410.SDFHSRC. The parameter list passed to DFHXCTRA is defined in the copybook DFHXCTRD, which is supplied in CICS410.SDFHMAC. DFHXCTRD also defines all the external CICS interface trace points for use by DFHXCTRA.

EXCI trace entry points

Table 9 (Page 1 of 8). External CICS interface trace entries

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

Table 9 (Page 2 of 8). External CICS interface trace entries

Point ID	Module	Lvl	Type	Data
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 0008	DFHXCPRH	Exc	XFRSTG1_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 0201	DFHXCPRH	Exc	NO_CICS_IRC_STARTED	1 Caller's parameter list 2 Call type 3 Caller's user name 4 Return codes and message pointer
EX 0202	DFHXCPRH	Exc	NO_PIPE	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

Table 9 (Page 3 of 8). External CICS interface trace entries

Point ID	Module	Lvl	Type	Data
EX 0408	DFHXCPRH	Exc	INVALID_UOWID	2 Call type 3 Caller's user name 1 Caller's parameter list
EX 0409	DFHXCPRH	Exc	INVALID_TRANSID	2 Call type 3 Caller's user name 4 UOWID 1 Caller's parameter list
EX 0414	DFHXCPRH	Exc	ABORT_RECEIVED	2 Call type 3 Caller's user name 1 Caller's parameter list
EX 0415	DFHXCPRH	Exc	INVALID_CONNECTION	2 Call type 3 Caller's user name 4 Target CICS applid 5 Message to be returned 1 Caller's parameter list
EX 0416	DFHXCPRH	Exc	INVALID_CICS_RELEASE	2 Call type 3 Caller's user name 4 Connection name 5 Target CICS applid 1 Caller's parameter list
EX 0417	DFHXCPRH	Exc	PIPE_MUST_CLOSE	2 Call type 3 Caller's user name 4 Target CICS applid 1 Caller's parameter list
EX 0418	DFHXCPRH	Exc	INVALID_PIPE_TOKEN	2 Call type 3 Caller's user name 4 Pipe token 1 Caller's parameter list
EX 0422	DFHXCPRH	Exc	SERVER_ABENDED	2 Call type 3 Caller's user name 4 Return codes and message pointer 5 DPL return area 1 Caller's parameter list
EX 0603	DFHXCPRH	Exc	XCUSER_GM_ERROR	2 Call type 3 Caller's user name 4 Return codes and message pointer 1 Caller's parameter list
EX 0604	DFHXCPRH	Exc	XCPPIPE_GM_ERROR	2 Call type 3 Caller's user name 4 Return codes and message pointer 1 Caller's parameter list
EX 0605	DFHXCPRH	Exc	VERIFY_BLOCK_GM_ERROR	2 Call type 3 Caller's user name 4 Return codes and message pointer 1 Caller's parameter list
EX 0606	DFHXCPRH	Exc	SSI_VERIFY_FAILED	2 Call type 3 Caller's user name 4 Return codes and message pointer 1 Caller's parameter list
EX 0607	DFHXCPRH	Exc	SVC_CALL_FAILED	2 Call type 3 Caller's user name 4 Return codes and message pointer 1 Caller's parameter list
EX 0608	DFHXCPRH	Exc	IRP_LOGON_FAILURE	2 Call type 3 Caller's user name 4 Return codes and message pointer 5 Target CICS applid 6 Logon name 1 Caller's parameter list
EX 0609	DFHXCPRH	Exc	IRP_CONNECT_FAIL	1 Caller's parameter list

Table 9 (Page 4 of 8). External CICS interface trace entries

Point ID	Module	Lvl	Type	Data
EX 0610	DFHXCPRH	Exc	IRP_DISC_FAIL	2 Call type 3 Caller's user name 4 Return codes and message pointer 5 Pipe token 6 Target CICS applid 1 Caller's parameter list
EX 0611	DFHXCPRH	Exc	IRP_LOGOFF_FAILED	2 Call type 3 Caller's user name 4 Return codes and message pointer 5 Target CICS applid 6 Pipe token 1 Caller's parameter list
EX 0612	DFHXCPRH	Exc	TRANSFORM_1_ERROR	2 Call type 3 Caller's user name 4 Return codes and message pointer 5 Target CICS applid 6 Pipe token 1 Caller's parameter list
EX 0613	DFHXCPRH	Exc	TRANSFORM_4_ERROR	2 Call type 3 Caller's user name 4 Return codes and message pointer 5 Target CICS applid 6 Pipe token 1 Caller's parameter list
EX 0614	DFHXCPRH	Exc	IRP_NULL_DATA	2 Call type 3 Caller's user name 4 Return codes and message pointer 5 Target CICS applid 1 Caller's parameter list
EX 0615	DFHXCPRH	Exc	IRP_NEG_RESPONSE	2 Call type 3 Caller's user name 4 Return codes and message pointer 5 Target CICS applid 1 Caller's parameter list
EX 0616	DFHXCPRH	Exc	IRP_SWITCH_PULL_ERR	2 Call type 3 Caller's user name 4 Return codes and message pointer 5 Target CICS applid 1 Caller's parameter list
EX 0617	DFHXCPRH	Exc	IRP_IOAREA_GM_ERR	2 Call type 3 Caller's user name 4 Return codes and message pointer 5 Target CICS applid 6 Pipe token 1 Caller's parameter list
EX 0619	DFHXCPRH	Exc	IRP_BAD_IOAREA	2 Call type 3 Caller's user name 4 IOAREA address 1 Caller's parameter list
EX 0620	DFHXCPRH	Exc	IRP_PROTOCOL_ERR	2 Call type 3 Caller's user name 4 Target CICS applid 5 Pipe token 1 Caller's parameter list
EX 0621	DFHXCPRH	Exc	PIPE_RECOVERY_FAILURE	2 Call type 3 Caller's user name 4 Target CICS applid 5 Pipe token 1 Caller's parameter list
EX 0622	DFHXCPRH	Exc	ESTAE_SETUP_FAIL	2 Call type 3 Caller's user name 4 Return codes and message pointer 1 Caller's parameter list

Table 9 (Page 5 of 8). External CICS interface trace entries

Point ID	Module	Lvl	Type	Data
EX 0623	DFHXCPRH	Exc	ESTAE_INVOKED	1 Caller's parameter list 2 Call type 3 Caller's user name 4 Return codes and message pointer 5 MVS abend code
EX 0624	DFHXCPRH	Exc	TIMEDOUT	1 Caller's parameter list 2 Call type 3 Caller's user name 4 Server program name 5 Target CICS applid
EX 0625	DFHXCPRH	Exc	STIMER_SETUP_FAIL	1 Caller's parameter list 2 Call type 3 Caller's user name 4 Return codes and message pointer
EX 0626	DFHXCPRH	Exc	STIMER_CANCEL_FAIL	1 Caller's parameter list 2 Call type 3 Caller's user name 4 Return codes and message pointer
EX 0627	DFHXCPRH	Exc	INCORRECT_SVC_LEVEL	1 Caller's parameter list 2 Call type 3 Caller's user name 4 SVC instruction
EX 0800	DFHXCPRH	Exc	RESP shows LENGERR	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	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	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	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	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	1 Caller's parameter list 2 Call type 3 Caller's user name 4 Program name 5 Target CICS applid 6 DPL_Retarea
EX 0806	DFHXCPRH	Exc	RESP shows TERMERR	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: 1 Caller's parameter list

Table 9 (Page 6 of 8). External CICS interface trace entries

Point ID	Module	Lvl	Type	Data
				<ul style="list-style-type: none"> 2 Call type 3 Caller's user name 4 Caller's register 14 For ALLOCATE_PIPE requests: 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: 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: 1 Caller's parameter list 2 Call type 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
EX 1001	DFHXCPRH	EX 1	Exit	
EX 1010	DFHXCEIP	EX 1	Entry	<ul style="list-style-type: none"> 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
EX 1011	DFHXCEIP	EX 1	Exit	<ul style="list-style-type: none"> 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)

Table 9 (Page 7 of 8). External CICS interface trace entries

Point ID	Module	Lvl	Type	Data
EX 2000	DFHXCPRH	EX 2	IRP_LOGON	7 Commarea length, if commarea passed 1 Caller's parameter list 2 Call type 3 Caller's user name 4 Target CICS applid 5 IRP userid 6 SLCB address
EX 2001	DFHXCPRH	EX 2	IRP_CONN	7 Connection name 1 Caller's parameter list 2 Call type 3 Caller's user name 4 Target CICS applid 5 IRP userid 6 IRP threadid
EX 2002	DFHXCPRH	EX 2	IRP_DISC	7 SCCB address 1 Caller's parameter list 2 Call type 3 Caller's user name 4 Target CICS applid
EX 2003	DFHXCPRH	EX 2	IRP_LOGOFF	5 Pipe token 1 Caller's parameter list 2 Call type 3 Caller's user name 4 Pipe token
EX 2004	DFHXCPRH	EX 2	IRP_SWITCH	5 IRP userid 1 Caller's parameter list 2 Call type 3 Caller's user name 4 Target CICS applid
EX 2005	DFHXCPRH	EX 2	IRP_SWITCH_DATA	6 IRP threadid 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

Table 9 (Page 8 of 8). External CICS interface trace entries

Point ID	Module	Lvl	Type	Data
				7 URMANCH, URM anchor point address
EX 3000	DFHXCEIP	Exc	ESTAE_SETUP_ERROR	1 Return area (20 bytes) 2 MVS return code
EX 3001	DFHXCEIP	Exc	ESTAE_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) 2 User token
EX 3011	DFHXCEIP	Exc	INV_PTOKEN_ON_OPEN	1 Return area (20 bytes) 2 Pipe token
EX 3012	DFHXCEIP	Exc	INV_CTYPE_ON_DPL	1 Return area (20 bytes) 2 Call type
EX 3013	DFHXCEIP	Exc	INV_VNUM_ON_DPL	1 Return area (20 bytes) 2 Version number
EX 3014	DFHXCEIP	Exc	INV_UTOKEN_ON_DPL	1 Return area (20 bytes) 2 User token
EX 3015	DFHXCEIP	Exc	INV_PTOKEN_ON_DPL	1 Return area (20 bytes) 2 Pipe token
EX 3017	DFHXCEIP	Exc	INV_USERID_ON_DPL	1 Return area (20 bytes) 2 Userid
EX 3018	DFHXCEIP	Exc	PIPE_NOT_OPEN_ON_DPL	1 Return area (20 bytes) 2 Pipe token
EX 3019	DFHXCEIP	Exc	PIPE_MUST_CLOSE_ON_DPL	1 Return area (20 bytes) 2 Pipe token
EX 3020	DFHXCEIP	Exc	INV_CTYPE_ON_CLOSE	1 Return area (20 bytes) 2 Call type
EX 3021	DFHXCEIP	Exc	INV_VNUM_ON_CLOSE	1 Return area (20 bytes) 2 Version number
EX 3022	DFHXCEIP	Exc	INV_UTOKEN_ON_CLOSE	1 Return area (20 bytes) 2 User token
EX 3023	DFHXCEIP	Exc	INV_PTOKEN_ON_CLOSE	1 Return area (20 bytes) 2 Pipe token
EX 3024	DFHXCEIP	Exc	INV_CTYPE_ON_DEALL	1 Return area (20 bytes) 2 Call type
EX 3025	DFHXCEIP	Exc	INV_VNUM_ON_DEALL	1 Return area (20 bytes) 2 Version number
EX 3026	DFHXCEIP	Exc	INV_UTOKEN_ON_DEALL	1 Return area (20 bytes) 2 User token
EX 3027	DFHXCEIP	Exc	INV_PTOKEN_ON_DEALL	1 Return area (20 bytes) 2 Pipe token
EX 3028	DFHXCEIP	Exc	PIPE_NOT_CLOSED_ON_DEALL	1 Return area (20 bytes) 2 Pipe token
EX 3029	DFHXCEIP	Exc	XCEIP_RETRYING	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.

Notes:

1. All numeric response and reason code values shown are in decimal.
- + 2. See "Setting the return code (R15) at termination" on page 53 for information about setting the job step return code in your EXCI client program.

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.

3 VERIFY_BLOCK_FM_ERROR

Explanation: Initialize_User processing requires storage below 16MB to build the parameter list for the SSI Verify call, and an error has occurred during the FREEMAIN for this area.

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 MVS FREEMAIN, to determine why the FREEMAIN is failing.

4 WS_FREEMAIN_ERROR

Explanation: An attempt to FREEMAIN working storage has resulted in an MVS 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 MVS FREEMAIN to determine why the FREEMAIN is failing.

5 XCIPIPE_FREEMAIN_ERROR

Explanation: An attempt to FREEMAIN pipe storage has resulted in an MVS 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 MVS 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 an MVS 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 MVS FREEMAIN to determine why the FREEMAIN is failing.

APAR PN84275

Documentation for PN84275 added on 2 August 1996

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.

+ 8 XFRSTG1_FM_FAILURE

+ **Explanation:** An attempt to FREEMAIN the transmission area has resulted in an MVS 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 XFRSTG1_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 MVS FREEMAIN to determine why the FREEMAIN is failing.

Reason codes for response: RETRYABLE

201 NO_CICS_IRC_STARTED

Explanation: An Initialize_User command has been issued on an MVS image that has had no IRC activity since the previous IPL, and the external CICS interface cannot determine the CICS SVC number.

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

User Response: Ensure that a CICS region in the MVS image has logged on to IRC (that is, has started up with the system initialization parameter IRCSTRT=YES or has started IRC dynamically with an OPEN IRC command). Alternatively, if there is no local CICS region in the MVS image, you must specify the SVC parameter that the external CICS interface is to use, by coding a CICSSVC parameter in the DFHXCOPT table. This situation can occur if you are using XCF to communicate to a CICS region in another MVS image. Once the problem has been resolved, re-issue the Initialize_User request.

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 (on Open_Pipe call) NO_CICS

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/ESA Data Areas* manual. The IRP return codes are in the DFHIRSPS copybook, listed under the heading IRC.

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

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/ESA Data Areas* manual. The IRP return codes are in the DFHIRSPS 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, CSML.

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 MVS 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 MVS 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 STEPLIB of the batch job. Ensure the CICS410.SDFHEXCI library is included in the STEPLIB 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 MVS 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 MVS reason code, determine why the load failed. The most likely reason is that the message table, DFHMET4E, is not in any library included in the STEPLIB of the batch job. Ensure the CICS410.SDFHEXCI library is included in the STEPLIB 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 MVS 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 MVS reason code, determine why the load failed. The most likely reason is that module DFHXCURM is not in any library included in the STEPLIB of the batch job. Ensure the library containing the module is included in the STEPLIB 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 MVS 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 MVS reason code, determine why the load failed. The most likely reason is that DFHXCTRA is not in any library included in the STEPLIB of the batch job. Ensure the library containing the module is included in the STEPLIB 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 it 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/ESA 4.1 region. CICS regions earlier than 4.1 do not recognize an EXCI call request, 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 TCB that has already been used by CICS or by CICS batch shared database. The external CICS interface cannot share a TCB 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 TCB (or daughter TCB), and issue the EXCI calls under that unique TCB.

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 MVS 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 MVS reason code, determine why the load failed. The most likely reason is that DFHXCOPT is not in any library included in the STEPLIB of the batch job. Correct the problem and restart the client application program.

421 RUNNING_UNDER_AN_IRB

Explanation: The EXCI call is issued under an MVS IRB, which is not permitted.

System Action: The call is rejected.

User Response: Determine why the call was issued under an IRB and change 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 MVS abend (U0408) to force a SYSMDUMP. The return code from the MVS 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. A possible reason for this is that the region size specified for the job is too small. If this is the case, increase the region size and restart the client application program.

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 MVS abend (U0403) to force a SYSMDUMP. The return code from the MVS 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. A possible reason for this is that the region size specified for the job is too small. If this is the case, increase the region size and restart the client application program.

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 an MVS user abend (0410) to force a SYSMDUMP.

User Response: Use the return code from the GETMAIN, with the dump, to determine why the

GETMAIN failed. A possible reason for this is that the region size of the job is too small. If this is the case, increase the region size and restart the client application program.

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. A possible reason for this is that the region size for the job is too small. If this is the case, increase the region size and restart the client application program.

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 MVS abend (U0409) to force a SYSMDUMP. The return code from the MVS 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. A possible reason for this is that the region size specified for the job is too small. If this is the case, increase the region size and restart the client application program.

606 SSI_VERIFY_FAILED

Explanation: A VERIFY call to the MVS subsystem interface (SSI) to obtain the current CICS SVC number failed.

System Action: The Initialize_User request is terminated. The return code from the SSI call is returned in subreason field-1 of the return area. This error occurs before the external CICS interface dump services are initialized, therefore EXCI issues an MVS user abend (0405) to force a SYSMDUMP.

User Response: Locate the return code in the dump, and use this with the rest of the dump and SSI documentation to determine why the VERIFY request failed. When the problem is resolved, restart the client application program.

607 CICS_SVC_CALL_FAILURE

Explanation: During Initialize_User processing, a call to the currently installed 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 MVS user abend (0406) to force a SYSMDUMP.

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/ESA Data Areas* manual. The IRP return codes are in the DFHIRSPS copybook, listed under the heading IRC. For example, subreason field-1 could show X'38' (decimal 56), indicating that there are no free logon access control blocks available (see constant IRERRNFL in the IRP control block). This indicates that the IRP session limit has been reached, which is 25 per EXCI address space. 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/ESA Data Areas* manual. The IRP return codes are in the DFHIRSPS 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/ESA Data Areas* manual. The IRP return codes are in the DFHIRSPS 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/ESA Data Areas* manual. The IRP return codes are in the DFHIRSPS 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 MVS 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. A possible reason for this is that the region size of the job is too small. If this is the case, increase the region size and restart the batch job.

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/ESA Data Areas* manual. The IRP return codes are in the DFHIRSPS 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 ESTAE_SETUP_FAILURE

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

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

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

623 ESTAE_INVOKED

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

System Action: The program check is handled by the EXCI ESTAE and an attempt is made to recover to a state that can support further EXCI calls. The MVS abend code is returned in the EXCI subreason field-1 of the return area. To aid further diagnosis, a SYSMDUMP 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 MVS STIMER 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 MVS 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 STIMER request issued to check the TIMEOUT value has failed with an error.

System Action: The return code from the STIMER 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 MVS STIMER 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.

627 INCORRECT_SVC_LEVEL

Explanation: The release level of the CICS SVC (DFHCSVC) is not the same (or higher) than the release level of the external CICS interface.

System Action: The Initialize_User request is terminated. This error occurs before the external CICS interface SDUMP facilities are initialized, therefore EXCI issues an MVS abend (U0407) to force a SYSMDUMP.

User Response: Determine the level of the CICS SVC being used and ensure it is the same release level as the external CICS interface, or higher. If the SVC number is allowed to default (CICSSVC=0 in DFHXCOPT), the SVC number being used is the SVC first used by a CICS region on the MVS image. That is, the SVC used by the first CICS region to open the CICS interregion communications (IRC). If the SVC number is specified on CICSSVC in DFHXCOPT, the SVC number specified is at an incorrect level. For more information, see the description of the CICSSVC parameter in Chapter 6, "External CICS interface options table, DFHXCOPT" on page 47.

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 LPA. 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 MVS image. For more details about installing DFHIRP, see the *CICS/ESA Installation 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

This chapter lists messages and abend codes for the external CICS interface.

DFHEX0001 An abend (code *aaa/bbbb*) has occurred in module *modname*.

Explanation: An unexpected program check or abend *aaa/bbbb* has occurred in module *modname*. This implies that there may be an error in external CICS interface code.

Alternatively, unexpected data has been passed on an external CICS interface call or storage has been overwritten.

The code *aaa/bbbb* is, if applicable, a 3-digit hexadecimal MVS system completion code *aaa* (for example, 0C1 or D37). If an MVS code is not applicable, this field is filled with three hyphens. The 4-digit code *bbbb*, which follows *aaa* is, if applicable, a user abend code produced by the external CICS interface. If the user abend code is not applicable, this field is filled with four hyphens.

System Action: An exception entry is made in the external CICS interface internal trace table, and to the GTF trace dataset (if GTF is active), and a SYSMDUMP is taken.

The external CICS interface terminates the current request, and attempts to recover to a consistent state so that further EXCI requests can be serviced. For an application using the EXCI CALL API, a response of EXCI_SYSTEM_ERROR with a REASON of ESTAE_INVOKED is returned to the application. For an application using the EXCI EXEC API, an EXEC_RESP of LINKERR is returned to the application, together with an EXEC_RESP2 of ESTAE_INVOKED or EXEC_ESTAE_INVOKED, depending on whether the call level ESTAE routine, or the EXEC level ESTAE routine was invoked.

User Response: Look up the MVS code *aaa*, if there is one, in the relevant MVS codes manual which is detailed in the book list in the front of this manual.

If applicable, see the description of abend code *bbbb* for further guidance.

You may need further assistance from IBM to resolve this problem. See Part 4 of the *CICS/ESA Problem Determination Guide* for guidance on how to proceed.

Destination: Console

Modules: DFHXCPRH, DFHXCEIP

DFHEX0002 A severe error (code *X'code'*) has occurred in module *modname*.

Explanation: An error has been detected in module *modname*. The code *X'code'* is the exception trace point ID which uniquely identifies what the error is and where the error was detected.

System Action: An exception entry is made in the EXCI internal trace table and to GTF if it is active, (*X'code'* in the message). A system dump is taken.

This is a critical error and the EXCI request is terminated. The external CICS interface attempts to recover to a consistent state so that further EXCI requests can be issued. For applications using the EXCI CALL API, the EXCI_REASON returned to the application indicates the reason for the error. For applications using the EXCI EXEC API, the reason is returned in the EXEC_RESP2 field of the RETCODE area.

User Response: This failure indicates a serious error in the external CICS interface code. For further information about the EXCI exception trace entries, refer to the *CICS/ESA Problem Determination Guide*.

You need further assistance from IBM to resolve this problem. See Part 4 of the *CICS/ESA Problem Determination Guide* for guidance on how to proceed.

Destination: Console

Modules: DFHXCPRH, DFHXCEIP

DFHEX0003 A GETMAIN request in module *modname* (code *X'code'*) has failed. Reason *X'rc'*.

Explanation: An MVS GETMAIN was issued by module *modname*, but it failed with return code *rc*.

The code *X'code'* is the exception trace point ID which uniquely identifies the place where the MVS GETMAIN was issued.

System Action: An exception entry is made in the EXCI internal trace table (code *X'code'* in the message). This is a critical error and the EXCI request is terminated. The external CICS interface attempts to recover to a consistent state so that further EXCI requests can be issued.

For applications using the EXCI CALL API, the EXCI_REASON returned to the application indicates the point of failure.

For applications using the EXCI EXEC API, the point of failure is returned in the EXEC_RESP2 field of the RETCODE area.

For EXCI_REASON and EXCI_RESP of 603, the EXCI module DFHXCPRH also issues abend 0410 which drives the ESTAE exit. Message DFHEX0001 is issued and a SYSMDUMP is taken

User Response: Look up the MVS GETMAIN return code *rc* in the relevant MVS codes manual.

If the reason is insufficient storage, try increasing the size of the region for the batch EXCI job.

You may need further assistance from IBM to resolve this problem. See Part 4 of the *CICS/ESA Problem Determination Guide* for guidance on how to proceed.

Destination: Console

Modules: DFHXCPRH, DFHXCTRI

**DFHEX0004 JOBNAME: *jobname*, STEPNAME: *stepname*, PROCNAME: *procname*,
SYSID IN SMF: *sysid*, APPLID: *applid***

Explanation: This message accompanies message DFHEX0001 and will provide the *jobname*, *stepname*, *procname*, *sysid* (in SMF) and *applid* to which the EXCI job is connecting to. If an insert value is unknown or not specified then the message insert will read Unknown. For example, *procname* and *stepname* are not mandatory in an EXCI job, and if they are omitted and DFHEX0004 is issued, the inserts for *procname* and *stepname* read "Unknown."

System Action: Follow system action for DFHEX0001.

User Response: Follow user response for DFHEX0001.

Destination: Console

Modules: : DFHXCPRH, DFHXCEIP

DFHEX0100 The installed level of CICS SVC does not support the EXCI call.

Explanation: The external CICS interface module DFHXCPRH detected that the level of CICS SVC (DFHCSVC) in use does not support the external CICS interface.

System Action: The EXCI request is terminated. An exception trace is made in the EXCI internal trace table, and if GTF is active, in the GTF trace data set. The external CICS interface module DFHXCPRH issues abend 0407 which drives the ESTAE exit. Message DFHEX0001 is issued, and a SYSMDUMP is taken.

User Response: Check the level of DFHCSVC installed in the LPA. A CICS/ESA 4.1 level of DFHCSVC is required for the external CICS interface. Generally, the latest level of DFHCSVC must be used when running CICS and the external CICS interface. For more information about installing DFHCSVC see the *CICS/ESA Installation Guide*.

Destination: Console

Module: DFHXCPRH

DFHEX0101 Unable to start interregion communication because DFHIRP services are down level.

Explanation: The version of DFHIRP being used is at a lower level than that of the External CICS Interface (EXCI) module DFHXCPRH.

System Action: The EXCI allocate pipe request is rejected, and a return code passed back to the batch application.

User Response: Update the level of the DFHIRP module in the LPA such that it matches the level of the latest CICS version in use.

Destination: Console

Module: DFHXCPRH

DFHEX0110 EXCI SDUMP has been taken. Dumpcode: *dumpcode*, Dumpid: *dumpid*.

Explanation: This message is issued on successful completion of a MVS SDUMP issued by external CICS interface module DFHXCDMP. An error, signalled by a previous message, caused a call to be made to DFHXCDMP to take a system dump.

The dump code *dumpcode* is an 8-character system dump code identifying the external CICS interface problem. A system dump code is the EXCI message number with the DFH prefix removed.

dumpid is the unique 9-character string identifying this dump.

System Action: The EXCI request is terminated.

User Response: See the EXCI message indicated by *dumpcode* for further guidance.

You may need further assistance from IBM to resolve this problem. See Part 4 of the *CICS/ESA Problem Determination Guide* for guidance on how to proceed.

Destination: Console

Module: DFHXCDMP.

DFHEX0111 EXCI SDUMP attempted but SDUMP is busy - will retry every five seconds for *n* seconds.

Explanation: At the time of the MVS SDUMP request issued by DFHXCDMP another address space in the same MVS system was in the process of taking an SDUMP. This causes MVS to reject the new request. A nonzero value for the dump retry parameter in the DFHXCOPT table means that the external CICS interface waits five seconds before retrying the SDUMP request. If necessary, the external CICS interface retries every five seconds for the total time specified on the retry parameter.

System Action: The external CICS interface issues an MVS STIMERM macro which causes it to wait for five seconds. The request is reissued when the delay interval has expired.

User Response: None.

Destination: Console

Module: DFHXCDMP.

DFHEX0112 SDUMP request failed - *reason X'nn'*.

Explanation: An MVS SDUMP request issued from the external CICS interface has failed to complete successfully. The possible reasons, (*reason*) for the failure are as follows:

ONLY PARTIAL DUMP

The SYS1.DUMP data set to which the dump is written is not large enough to contain all of the dumped storage.

SDUMP BUSY

At the time of the MVS SDUMP request issued by the EXCI, another address space in the same MVS system was in the process of taking an SDUMP. This causes MVS to reject the new request. If a nonzero value is specified for the dump retry parameter in DFHXOPTS table, the EXCI has retried the SDUMP request every five seconds for the specified period. This message is only issued if SDUMP is still busy after the final retry.

DFHEX0113

STIMERM FAILED

In order to delay for five seconds before retrying SDUMP after an SDUMP BUSY condition, the EXCI issues an MVS STIMERM macro request. MVS has indicated that the STIMERM request has failed.

NO DATA SET AVAILABLE

No SYS1.DUMP data sets were available at the time the SDUMP request was issued.

REJECTED BY MVS, REASON = X'nn'

MVS has rejected the SDUMP request because of user action (for example, specifying DUMP=NO in the MVS IPL) or because of an I/O error or terminating error in the SDUMP routine. X'nn' is the SDUMP reason code.

NOT AUTHORIZED FOR EXCI

SDUMP is not authorized for the external CICS interface.

INSUFFICIENT STORAGE

The EXCI issued an MVS GETMAIN for subpool 253 storage during the processing of the SDUMP request. The GETMAIN has been rejected by MVS.

System Action: The EXCI proceeds as if the dump had been successful.

User Response: The user response depends on the reasons, (*reason*), for the failure.

ONLY PARTIAL DUMP

Increase the size of the SYS1.DUMP data sets and cause the SDUMP request to be reissued.

SDUMP BUSY

Cause the SDUMP to be reissued after, if appropriate, increasing the dump retry time in DFHXCOPT.

STIMERM FAILED

Use MVS problem determination methods to fix the STIMERM failure and then cause the SDUMP request to be reissued.

NO DATA SET AVAILABLE

Clear a SYS1.DUMP data set and then cause the SDUMP request to be reissued.

REJECTED BY MVS, REASON = X'nn'

No action is required if the dump is suppressed deliberately. If the dump has failed because of an error in the MVS SDUMP routine, use MVS problem determination methods to fix the error and then cause the SDUMP request to be reissued. See the *MVS/ESA Application Development Macro Reference* for an explanation of the SDUMP reason code X'nn'.

NOT AUTHORIZED FOR EXCI

This reason is unlikely because SDUMP is unconditionally authorized during EXCI initialization, and should be authorized throughout the EXCI run. If you do get this reason, the EXCI AFCB (authorized function control block) has probably been accidentally overwritten.

INSUFFICIENT STORAGE

Ensure sufficient storage is available to MVS for subpool 253 requests.

Destination: Console

Module: DFHXCDMP

DFHEX0113 EXCI trace Initialization has failed.

Explanation: An attempt to initialize external CICS interface (EXCI) trace facilities during EXCI initialization has failed.

System Action: The EXCI request continues without trace facilities. An earlier message identifies the cause of the failure.

User Response: Refer to the earlier message to determine the cause of the failure.

Destination: Console

Module: DFHXCTRI

DFHEX0114 Incorrect data has been passed for EXCI tracing causing a program check in DFHXCTRP.

Explanation: Some data passed to the external CICS interface (EXCI) trace module DFHXCTRP for addition to the EXCI internal trace table, or GTF trace, caused a program check to occur when an attempt was made to access it.

The most likely cause of this error is incorrect data passed on an EXCI CALL API request that the trace program DFHXCTRP is attempting to access.

System Action: The EXCI request is terminated and a SYSMDUMP is taken.

User Response: Examine the dump to determine the source of the incorrect data.

You may need further assistance from IBM to resolve this problem. See Part 4 of the *CICS/ESA Problem Determination Guide* for guidance on how to proceed.

Destination: Console

Module: DFHXCTRI

DFHEX0115 EXCI trace services have been disabled due to a previous error.

Explanation: An error occurred in the external CICS interface (EXCI) trace module DFHXCTRP indicated by message DFHEX0001. In trying to recover from the error, module DFHXCTRI determined that the error was not caused by accessing incorrect data passed to DFHXCTRP, but was due to a program check in DFHXCTRP.

System Action: The EXCI trace facilities are disabled to prevent further errors. A SYSMDUMP is taken.

User Response: See the DFHEX0001 message and the SYSMDUMP to determine the cause of the error.

You may need further assistance from IBM to resolve this problem. See Part 4 of the *CICS/ESA Problem Determination Guide* for guidance on how to proceed.

Destination: Console

Module: DFHXCTRI

DFHEX0116 Program check occurred within global trap exit - DFHXCTRA now marked unusable.

Explanation: After making a trace entry, the external CICS interface (EXCI) trace program DFHXCTRP called the EXCI field engineering global trap program DFHXCTRA. A program check occurred during execution of DFHXCTRA.

System Action: The EXCI marks the currently active version of DFHXCTRA as unusable and ignores it on subsequent calls to DFHXCTRP for all subsequent calls made under this TCB. The EXCI request is terminated, and a SYSMDUMP is taken.

User Response: Use the dump to find the cause of the program check.

You may need further assistance from IBM to resolve this problem. See Part 4 of the *CICS/ESA Problem Determination Guide* for guidance on how to proceed.

You should use the global trap exit only in consultation with an IBM support representative.

Destination: Console

Module: DFHXCTRI

Index

A

- abend codes 67
- addressing mode (AMODE)
 - assembler sample program 53
 - client program requirements 51
- allocate_opts, parameter of ALLOCATE_PIPE command 12
- ALLOCATE_PIPE command 11
 - invocation of DFHXCURM during 45
 - security check failure 61
- allocating a pipe 11
- AMODE (addressing mode)
 - See addressing mode (AMODE)
- application programming
 - commands 7
 - copybooks 28
 - DPL subset 2
 - exception conditions returned on LINK command 33
 - language considerations 52
 - RESP and RESP2 fields 33
 - restrictions for server programs 2
 - sample programs 53
 - stub 51
 - translation required for EXEC CICS LINK command 38
- applid, specifying on ALLOCATE_PIPE command 11
- assembler
 - CICS-supplied procedure, DFHEXTAL 51
 - copybook 28
 - EXCI CALL interface 7
 - sample program 30, 53
- automatic retry of EXEC CICS LINK 35

B

- basic sequential access method (BSAM)
 - use of by assembler sample client program 53
- batch jobs, querying the status of 44
- benefits of external CICS interface 5
 - See *also* external CICS interface (EXCI)
- bind-time security 61
- BSAM (basic sequential access method)
 - See basic sequential access method (BSAM)

C

- C language
 - CICS-supplied procedure, DFHEXTDL 51
 - CICS-supplied procedure, DFHYXTDL 52
 - copybook 28
 - EXCI CALL interface 7

- C language (*continued*)
 - sample program 30, 53
 - special considerations for client program 52
- CALL interface
 - See external CICS interface (EXCI)
- call_type
 - parameter of ALLOCATE_PIPE command 11
 - parameter of CLOSE_PIPE command 23
 - parameter of DEALLOCATE_PIPE command 25
 - parameter of DPL_Request command 15
 - parameter of INITIALIZE_USER command 9
 - parameter of OPEN_PIPE command 13
- capturing dumps 66
- CEMT INQUIRE IRBATCH command 44
- CICS system definition file, EXCI sample definitions 56
- CICS_applid, parameter of ALLOCATE_PIPE command 11
- CICSSVC, parameter of DFHXC macro 47
- client program
 - addressing mode 51
 - BSAM, use of by assembler sample program 53
 - compiling 51
 - connection through DFHIRP 6
 - definition of 1
 - JCL needed
 - capturing SYSMDUMPs 66
 - running an EXCI client 59
 - link-editing 51
 - linking to server with EXEC CICS LINK 31
 - MRO logon and bind-time security 61
 - PL/I and C language considerations 52
 - sample job for starting 60
 - sample program 53
 - translating 38, 51
 - use of multiple sessions 1
- CLOSE_PIPE command 23
- closing a pipe 23
- COBOL II
 - CICS-supplied procedure, DFHEXTVL 51
 - CICS-supplied procedure, DFHYXTVL 52
 - copybook 28
 - example of EXCI DPL call 29
 - EXCI CALL interface 7
 - sample program 53
- codes, abend 67
- COMMAREA_len, parameter of DPL_Request command 16
- COMMAREA, parameter of DPL_Request command 16
- CONFDATA, parameter of DFHXCO macro 48
- 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 28

cross-system multiregion operation (XCF/MRO) 1

CSMI (CICS-supplied mirror transaction)
 attached by CICS server 33
 authorizing the link userid 62
 default transid 17
 security 62

D

data_len, parameter of DPL_Request command 16

DCE (distributed computing environment) 6

DEALLOCATE_PIPE command 25

deallocating a pipe 25

defining resources
 See resource definition

DFH\$AXCC, assembler sample program 30
 example of output 57

DFH\$DXCC, C sample program 30

DFH\$EXCI, sample server definitions 56

DFH\$FILA, sample file definitions 56

DFH\$PXCC, PL/I sample program 30

DFHAPPL FACILITY class profiles, defining 62

DFHEXTAL, procedure for assembler client programs 51

DFHEXTDL, procedure for C client programs 51

DFHEXTPL, procedure for PL/I client programs 51

DFHEXTVL, procedure for VS COBOL II client programs 51

DFHIRP (interregion communication program)
 connection of client and server 6
 security checks performed by 61

DFHLIST, note about sample definitions 56

DFHXCIE, alias for DFHXCSTB stub 51

DFHXCIS, alias for DFHXCSTB stub 51

DFHXCLOPT, options table 47

DFHXCPLD, return area and equate copybook for assembler 28

DFHXCPLH, return area and equate copybook for C language 28

DFHXCPLL, return area and equate copybook for PL/I 28

DFHXCPLP, return area and equate copybook for COBOL 28

DFHXCRCO, return code copybook for assembler 28

DFHXCRCRCH, return code copybook for C language 28

DFHXCRCRCL, return code copybook for PL/I 28

DFHXCRCO, return code copybook for COBOL 28

DFHXCSTB, stub for client programs 51

DFHXCSTRA, EXCI service trap 70

DFHXCSTRD, parameter list 70

DFHXCURM, user-replaceable module 45

DFHYXTDL, LE/370 procedure for C client programs 52

DFHYXTPL, LE/370 procedure for PL/I client programs 52

DFHYXTVL, LE/370 procedure for VS COBOL II client programs 52

disconnecting a pipe 23

distributed computing environment (DCE) 6

distributed program link (DPL)
 API subset for server programs 2
 example COBOL call without userid and uowid 29
 request program call 15

DPL (distributed program link)
 See distributed program link (DPL)

DPL_opts, parameter of DPL_Request command 20

DPL_Request call 15

dpl_retarea, parameter of DPL_Request command 19

dumps 66
 formatting 66
 SYSMDUMP 66

DURETRY, parameter of DFHXC macro 48

E

ELPA (extended link pack area), installation of DFHIRP 6

EQUATE copybooks 28

error codes
 See abend codes
 See exception conditions returned on LINK command
 See reason codes
 See RESP and RESP2 fields
 See response codes

exception conditions returned on LINK command 33

EXCI (external CICS interface)
 See external CICS interface (EXCI)

EXEC CICS LINK command 31
 automatic retry 35
 choosing between EXEC CICS and CALL interface 2
 DFHAPPL profile definition 62
 security checking 62, 63
 translation 38

extended link pack area (ELPA), installation of DFHIRP 6

external CICS interface (EXCI) 1, 31
 benefits 5
 CALL interface 2, 7
 choosing between EXEC CICS and CALL interface 2

external CICS interface (EXCI) (*continued*)
 CALL interface (*continued*)
 DFHAPPL profile definition 62
 return area 28
 syntax 7
 CICS releases supported 6
 compiling and link-editing client programs 51
 defining connections 39
 description of 1
 inquiring on the state of connections 44
 languages supported 7
 messages 91
 options table, DFHXCOPT 47
 PL/I and C language considerations 52
 problem determination 65
 programming languages supported 7
 reason codes 79
 response codes 79
 security 61
 user-replaceable module (DFHXCURM) 45

F

FACILITY class profiles, defining 62
 freeing storage associated with a pipe 25
 function call EQUATE copybooks 28

G

generic connection
 definition of 40
 note about lack of security checks 61
 GTF, parameter of DFHXCO macro 49

I

INITIALIZE_USER command 8
 interregion communication
 See DFHIRP (interregion communication program)
 interregion communication (IRC)
 opening after installation of sample definitions 56
 interregion communication program (DFHIRP)
 See DFHIRP (interregion communication program)
 IRBATCH on CEMT INQUIRE command 44

J

job control language (JCL)
 for capturing SYSMDUMPs 66
 for running an EXCI client program 59

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 51
 translation required for EXEC CICS LINK
 command 38
 use of DFHXCSTB stub 7
 logon security 61

M

messages 91
 mirror transaction
 See *also* CSMI (CICS-supplied mirror transaction)
 authorizing the link userid 62
 security 62
 MRO (multiregion operation)
 See multiregion operation (MRO)
 MSGCASE, parameter of DFHXCO macro 49
 multiregion operation (MRO)
 cross-system (XCF/MRO) 1
 logon and bind time security 61
 MVS abends 67

N

null parameters, example of EXCI CALLs with 29

O

open system interface (OSI) 6
 OPEN_PIPE command 13
 opening a pipe 13
 options table, DFHXCOPT 47
 OSI (open system interface) 6

P

parameters
 null 29
 specifying with options table 47
 pgmname
 parameter of DPL_Request command 15
 pipe
 See *also* pipe_token
 allocating 11
 closing 23
 connecting 13
 deallocating 25
 definition of 1
 disconnecting 23
 freeing storage associated with 25
 invocation of DFHXCURM during
 ALLOCATE_PIPE 45
 opening 13
 restriction on leaving open 13
 reusing a closed pipe 23

- pipe_token
 - parameter of ALLOCATE_PIPE command 11
 - parameter of CLOSE_PIPE command 23
 - parameter of DEALLOCATE_PIPE command 25
 - parameter of DPL_Request command 15
 - parameter of OPEN_PIPE command 14
- PL/I
 - CICS-supplied procedure, DFHEXTPL 51
 - CICS-supplied procedure, DFHYXTPL 52
 - copybook 28
 - EXCI CALL interface 7
 - sample program 30, 53
 - special considerations for client program 52
- problem determination 65
 - dumps 66
 - MVS abends 67
 - service trap 70
 - trace 65
- programming
 - See application programming
- programming restrictions for server programs 2
- PROTOCOL attribute
 - CONNECTION definition 40
 - SESSIONS definition 41

R

- RACF (resource access control facility)
 - See resource access control facility (RACF)
- reason codes 79
 - Allocate_Pipe call 12
 - Close_Pipe call 23
 - Deallocate_Pipe call 26
 - DPL call 21
 - Intialize_User call 9
 - Open_Pipe call 14
- RECEIVECOUNT attribute, SESSIONS definition 41
- RECEIVEPFX attribute, SESSIONS definition 42
- residence mode (RMODE)
 - assembler sample program 53
- resource access control facility (RACF) 61
 - specifying userid on DPL_Request command 18
- resource definition
 - CONNECTION definition 39
 - sample programs 56
 - SESSIONS definition 41
- RESP and RESP2 fields 33
- response codes 27, 79
 - Allocate_Pipe call 12
 - Close_Pipe call 23
 - Deallocate_Pipe call 26
 - DPL call 21
 - Intialize_User call 9
 - Open_Pipe call 14
- retries on an EXEC CICS LINK command 35

- return code 53
 - clearing R15 53
- return_area
 - parameter of ALLOCATE_PIPE command 11
 - parameter of CLOSE_PIPE command 23
 - parameter of DEALLOCATE_PIPE command 25
 - parameter of DPL_Request command 15
 - parameter of INITIALIZE_USER command 8
 - parameter of OPEN_PIPE command 13
- reusing a closed pipe 23
- RMODE (residence mode)
 - See residence mode (RMODE)
- running the sample applications 57

S

- sample programs 30, 53
 - description 53
- security 61
- 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 53
 - security considerations 62
- service trap 70
- session
 - See pipe
- SESSIONS definition
 - PROTOCOL attribute 41
 - RECEIVECOUNT attribute 41
 - RECEIVEPFX attribute 42
 - SENDCOUNT attribute 42
 - SENDPFX attribute 43
- specific connection
 - definition of 39
 - MRO logon security checks 61
- storage, freeing 25
- stub for client programs
 - DFHXCIE 51
 - DFHXCIS 51
 - DFHXCSTB 51
- suppressing user data in trace
 - CONFDATA option 48
- SYSDUMPS, capturing 66
- sysplex, use of cross-system MRO 1
- system definition
 - See resource definition
- system definition file (CSD), CICS 56

T

TIMEOUT, parameter of DFHXCO macro 49
trace 65
TRACE parameter of DFHXCO macro 50
trace points 70
TRACESZE parameter of DFHXCO macro 50
transid, parameter of DPL_Request command 17
translation of EXEC CICS LINK command 38
trap, DFHXCTRA 70
TRAP, parameter of DFHXCO macro 50
TYPE, parameter of DFHXCO macro 47

U

unit-of-work identifier, specifying on DPL_Request 17
uowid, parameter of DPL_Request 17
user environment, initializing 8
user security 63
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 23
parameter of DEALLOCATE_PIPE command 25
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 23
parameter of DEALLOCATE_PIPE command 25
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

X

XCF/MRO (cross-system multiregion operation)
See cross-system multiregion operation (XCF/MRO)

Sending your comments to IBM

CICS for MVS/ESA

External CICS Interface

SC33-1390-03

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM.

Feel free to comment on what you regard as specific errors or omissions, and on the accuracy, organization, subject matter, or completeness of this book. Please limit your comments to the information in this book and the way in which the information is presented.

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, use the Readers' Comment Form
- By fax:
 - From outside the U.K., after your international access code use 44 962 870229 (after 16 April 1995, use 44 1962 870229)
 - From within the U.K., use 0962 870229 (after 16 April 1995, use 01962 870229)
- Electronically, use the appropriate network ID:
 - IBM Mail Exchange: GBIBM2Q9 at IBMAIL
 - IBMLink: HURSLEY(IDRCF)
 - Internet: idrcf@hursley.vnet.ibm.com

Whichever 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.

Readers' Comments

CICS for MVS/ESA

External CICS Interface

SC33-1390-03

Use this form to tell us what you think about this manual. If you have found errors in it, or if you want to express your opinion about it (such as organization, subject matter, appearance) or make suggestions for improvement, this is the form to use.

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. This form is provided for comments about the information in this manual and the way it is presented.

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.

Be sure to print your name and address below if you would like a reply.

Name

Address

Company or Organization

Telephone

Email



You can send your comments POST FREE on this form from any one of these countries:

Australia	Finland	Iceland	Netherlands	Singapore	United States
Belgium	France	Israel	New Zealand	Spain	of America
Bermuda	Germany	Italy	Norway	Sweden	
Cyprus	Greece	Luxembourg	Portugal	Switzerland	
Denmark	Hong Kong	Monaco	Republic of Ireland	United Arab Emirates	

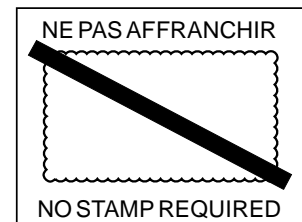
If your country is not listed here, your local IBM representative will be pleased to forward your comments to us. Or you can pay the postage and send the form direct to IBM (this includes mailing in the U.K.).

1 Cut along this line

2 Fold along this line

By air mail
Par avion

IBRS/CCRINUMBER: PHQ - D/1348/SO



**REPONSE PAYEE
GRANDE-BRETAGNE**

IBM United Kingdom Laboratories
Information Development Department (MP095)
Hursley Park,
WINCHESTER, Hants
SO21 2ZZ United Kingdom

3 Fold along this line

From: Name _____
Company or Organization _____
Address _____

EMAIL _____
Telephone _____

1 Cut along this line

4 Fasten here with adhesive tape



Program Number: 5655-018



Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC33-1390-03

