

CICS[®] Family:



Communicating from CICS on System/390[®]

CICS[®] Family:



Communicating from CICS on System/390[®]

Note!

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

Third Edition (March 1999)

This edition applies to the following IBM Customer Information Control System (CICS) licensed programs, and to all subsequent releases and modifications of these programs, until otherwise indicated in new editions:

CICS Transaction Server for OS/390, program number 5655-147
CICS/ESA Version 4, program number 5655-018
CICS/ESA Version 3, program number 5685-083
CICS Transaction Server for VSE/ESA, program number 5648-054
CICS/VSE Version 2, program number 5686-026

Consult the latest edition of the applicable IBM system bibliography for current information on these products.

This book is based on the second edition of the *CICS Family: Communicating from CICS on System/390* manual, SC33-1697-01. Changes from that edition are marked by vertical lines 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 1993, 1999. All rights reserved.

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

Contents

Notices	ix
Programming interface information	x
Trademarks and service marks	xi
Preface	xiii
What this book is about	xiii
Who this book is for	xiii
What is not covered by this book	xiii
What you need to know to understand this book	xiv
Terminology	xiv
Macro syntax notation	xv
Determining if a publication is current	xvi
Book structure	xvii
Bibliography	xix
CICS Family intercommunication books	xix
CICS on System/390 intercommunication books	xix
CICS Transaction Server for OS/390 Release 3	xix
CICS Transaction Server for OS/390 Release 2	xix
CICS Transaction Server for OS/390 Release 1	xix
CICS/ESA Version 4	xix
CICS/ESA Version 3	xix
CICS Transaction Server for VSE/ESA Release 1	xx
CICS/VSE Version 2	xx
CICS non-System/390 intercommunication books	xx
CICS Clients	xx
Summary of changes	xxi
Changes for the second edition	xxi
Changes for the first edition	xxii

Part 1. Communicating with non-System/390 CICS systems 1

Chapter 1. Overview	3
Function shipping	3
Restrictions on function shipping	4
Data conversion	4
Transaction routing	4
CICS on System/390→CICS non-System/390	5
CICS non-System/390→CICS on System/390	5
Data conversion	5
Transaction routing restrictions	6
Distributed program link (DPL)	6
Restrictions on programs linked by DPL	6
Note for DB2®	6
Abends when using DPL	7
Performance optimization for DPL	7
Asynchronous processing	8
Distributed transaction processing (DTP)	8
Summary of CICS System/390–non-System/390 intercommunication	8
Chapter 2. Planning	11
Path length and resource definition tradeoffs	11

Assumptions	11
Possible approaches	12
Summary	14
Syncpointing (LU 6.2)	15
Function shipping from CICS on System/390 to CICS for OS/2	15
DPL or function shipping from CICS for OS/2 to CICS on System/390	16
Chapter 3. Resource definition	17
System generation and initialization	18
LU 6.2 link	19
All communication with CICS non-System/390 systems	19
Function shipping and DPL from CICS non-System/390 systems	19
Function shipping and DPL to CICS non-System/390 systems	19
Transaction routing from CICS non-System/390 systems	20
Transaction routing to CICS non-System/390 systems	20
Defining remote resources	20
CICS on System/390 mirror transaction definition	20
CICS on System/390 mirror program definition	20
Data conversion program definition	22
Data conversion table definition	22
User-replaceable conversion program definition	22
Remote system TERMINAL definition (single session)	23
Remote system TYPETERM definition	25
CONNECTION definition for parallel sessions	26
SESSIONS definition for parallel sessions	27
Remote terminal TERMINAL definition	28
Remote terminal TYPETERM definition	30
Chapter 4. CICS internal security	31
Introduction to CICS/VSE internal security	31
Security profile	31
Link profile	31
SNT entry for link	32
Required specifications in remote systems	32
CICS/VSE specifications	33
Specifying LU 6.2 security requirements	34
Mirror transaction	34
Mirror program and data conversion modules	34
Connection	35
Session	35
File control table (FCT)	36
Signon table (SNT)	36
Chapter 5. VTAM/NCP definitions	39
VTAM application definition	39
Single sessions	39
Parallel sessions	40
Chapter 6. Data conversion	43
Where data conversion takes place	43
Function shipping and DPL	43
Distributed transaction processing	44
Transaction routing	44
Avoiding data conversion	44
Types of conversion	44
Character data	45

CICS-supported conversions	45
Arabic	47
Baltic Rim.	47
Cyrillic	47
Estonia.	48
Greek	48
Hebrew.	48
Latin-1	49
Latin-1 including euro support	50
Latin-2	50
Latin-5	51
Latin-9	51
Japanese	52
Korean	53
Simplified Chinese	53
Traditional Chinese	54
Unicode data	54
Binary data	54
The conversion process	55
Components	55
Process	55
Standard and nonstandard conversion	56
Sequence of conversion processing	57
Resource definition to enable data conversion	59
Defining the conversion table.	59
DFHCNV macro types	59
DFHCNV TYPE=INITIAL	61
DFHCNV TYPE=ENTRY	63
DFHCNV TYPE=KEY	64
DFHCNV TYPE=SELECT	65
DFHCNV TYPE=FIELD.	66
DFHCNV TYPE=FINAL.	68
Hints on coding the macros	68
User-defined conversion tables	68
SRVERCP=USR	69
SRVERCP=USRD.	69
Invalid and undefined DBCS characters.	71
Example macros	72
Assembling and link-editing the conversion programs.	76
User-replaceable conversion program	76
Input to DFHUCNV	76
Supplied user-replaceable conversion program	83

Part 2. Server Support for CICS Clients 89

Chapter 7. Introduction to CICS Clients	91
What is a CICS Client?	91
What functions do CICS Clients provide?	91
The External Call Interface	91
The External Presentation Interface	92
Terminal emulation	93
What protocols are supported?	93
APPC	93
TCP/IP support.	93
Benefits of Client support	93
Further information about Clients	94

Chapter 8. Installation of server support for Clients	95
DFHCLNT resource group	95
CSCC transient data queue	95
Connections to Clients	95
Static APPC definitions	95
Autoinstalled APPC connections	96
Using TCP/IP	96
Client virtual terminals	98
Using static definitions	98
Using autoinstall	100
How CICS installs Client terminals	102
Defining a Client-attached printer	103
Example static definition of a Client-attached printer	103
Security	106
Bind security	106
Link security	106
User security	106
CICS-supplied transactions	107
System initialization parameters	107
Chapter 9. Data conversion for Clients	109
The client code page	109
The ECI	109
The EPI and terminal emulator	109
The server code page and character set	110
The ECI	110
The EPI and terminal emulator	110
Binary data conversion	111
Defining code pages to CICS on System/390	111
Chapter 10. Application programming for Clients	113
ECI server programs	113
EPI server programs	113
Client-attached printers	114
Chapter 11. Problem determination for Clients	117
Trace points	117
Messages	117
Abend codes	117
Chapter 12. Recovery after a restart of CICS	119
The Client terminal emulator	119
Client EPI and ECI programs	119
Chapter 13. Restrictions on Client support	121
Chapter 14. Migration considerations	123
Moving to a client/server environment	123
Server applications	123
Data conversion	123
The External Call Interface	123
The External Presentation Interface	123

Part 3. Appendixes 125

Glossary of Client-related terms	127
---	-----

Index	131
Sending your comments to IBM	135

Notices

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

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

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

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

The following paragraph does not apply in 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.

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

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact: IBM United Kingdom Laboratories, Mail Point 151, Hursley Park, Winchester, Hampshire, England, SO21 2JN. Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

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

Programming interface information

This book is intended to help you to set up a CICS System/390 product to communicate with CICS for OS/2, CICS for Windows NT, CICS on Open Systems, CICS/400, or the CICS Clients workstation products.

This book also documents Product-sensitive Programming Interface and Associated Guidance 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, either by an introductory statement to a chapter or section or by the following marking:

┌ **Product-sensitive programming interface** _____

Product-sensitive Programming Interface and Associated Guidance Information...

└ **End of Product-sensitive programming interface** _____

This book contains sample programs. Permission is hereby granted to copy and store the sample programs into a data processing machine and to use the stored copies for study and instruction only. No permission is granted to use the sample programs for any other purpose.

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:

AIX	IBMLink
AnyNet	Language Environment
AS/400	MVS/ESA
BookManager	MVS/XA
C/370	OS/2
CICS	OS/390
CICS/400	Presentation Manager
CICS/6000	RACF
CICS/ESA	RISC System/6000
CICS/MVS	S/370
CICS OS/2	System/370
CICS/VSE	System/390
COBOL/2	VSE/ESA
DB2	VTAM
IBM	

INTEL is a registered trademark of Intel Corporation, in the United States, or other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems Inc, in the United States, or other countries, or both.

Microsoft, Windows, and Windows NT are trademarks of Microsoft Corporation in the United States, or other countries, or both.

UNIX is a trademark of X/Open Company Limited, in the United States, or other countries, or both.

Preface

What this book is about

This book is about setting up a CICS® System/390® product—CICS Transaction Server for OS/390®, CICS/ESA®, CICS Transaction Server for VSE/ESA®, or CICS/VSE®—to communicate with a non-System/390 CICS product—CICS for OS/2®, CICS for Windows™ NT™, CICS on Open Systems, CICS/400®, or one of the CICS Clients workstation products. This type of communication is called CICS interproduct communication.

For an overview of CICS interproduct communication, see the *CICS Family: Interproduct Communication* manual, which explains the documentation scheme of which this book is a part. Note that Chapter s 6-11 of the *CICS Family: Interproduct Communication* manual duplicate material in the *Intercommunication Guide* for your CICS System/390 product. These chapters are primarily intended for users of CICS non-System/390 products.

In general, the present book does not duplicate information in a CICS System/390 product library. It may, however, duplicate material in the CICS non-System/390 product libraries, if the information is relevant to users of CICS System/390 products.

Who this book is for

This book is intended for those responsible for planning and implementing the **System/390 side** of intercommunication between a System/390 CICS system and any of the following non-System/390 CICS products—CICS for OS/2, CICS for Windows NT, CICS on Open Systems, CICS/400, or CICS Clients.

What is not covered by this book

This book describes communication between System/390 and non-System/390 CICS products. It does not describe:

- CICS on System/390-CICS on System/390 communication. See the *Intercommunication Guides* for the relevant CICS on System/390 products.
- Distributed transaction processing. See the *Distributed Transaction Programming* manual for your CICS on System/390 product.
- The Front End Programming Interface feature of some CICS on System/390 products. See the *Front End Programming Interface User's Guide* for your CICS on System/390 product.
- CICS/VSE Version 2.3's support for the CICS Clients workstation products. This is described in a separate manual—the *CICS/VSE 2.3 Server Support for CICS Clients* manual, SC33-1712.
- Access to CICS programs and transactions from non-CICS environments. Some of the CICS on System/390 products offer more support for this than others. The books that describe this support are:
 - The *CICS External CICS Interface* manual
 - The *CICS External Interfaces Guide*
 - The *CICS Internet and External Interfaces Guide*
 - The *CICS Internet Guide*
 - The *CICS ONC RPC Guide*
 - The *CICS Web Interface Guide*.

To discover which of these books apply to your CICS on System/390 product, and their order numbers, see “CICS on System/390 intercommunication books” on page xix.

What you need to know to understand this book

This book assumes a conceptual understanding of CICS intercommunication, recovery and restart, resource definition, customization, and security.

Terminology

The following CICS products run on computers of the System/370™ or System/390 series, and support communication with CICS products that run on other hardware platforms:

CICS Transaction Server for OS/390, program number 5655-147
CICS/ESA Version 4, program number 5655-018
CICS/ESA Version 3, program number 5685-083
CICS Transaction Server for VSE/ESA, program number 5648-054
CICS/VSE Version 2, program number 5686-026

The term **System/390** is used to refer to any 370/390 computer on which one of the above products can run. The term **non-System/390** refers to the hardware platforms used by other CICS products—AS/400® (used by CICS/400), IBM®-compatible personal computers (used by CICS for OS/2 and CICS for Windows NT), and RISC System/6000® (used by CICS on Open Systems).

In statements that apply to each of the CICS products that runs on a System/390 hardware platform, the generic term **CICS on System/390** is used to represent all of them. One of these CICS products is referred to by name only if there is a difference in its interface to non-System/390 products as compared with the interface from other System/390 products. Subject to explicitly-stated exceptions, interpret all references to CICS as applying to your CICS System/390 product.

A reference to a CICS product that does not quote a Version number applies to all current versions of the product. For example, the term *CICS/ESA*, without a qualifying Version number, means both CICS/ESA Version 3 and CICS/ESA Version 4. Where it is necessary to distinguish between these products, the full product names and Version numbers are quoted.

The term *CICS for OS/2*, without a qualifier, is used as a generic name for:

- Transaction Server for OS/2 Warp Version 4.0 (which contains CICS for OS/2 Version 3)
- Transaction Server for OS/2 Warp Version 4.1 (which contains CICS for OS/2 Version 3.1)

Where it is necessary to distinguish between these products, the full product names and Version numbers are quoted.

The term *CICS for Windows NT*, without a qualifier, is used as a generic name for:

- Transaction Server for Windows NT Version 4.0 (which contains CICS for Windows NT Version 4.0)
- TX Series Version 4.2 for Windows NT (which contains CICS for Windows NT Version 4.2)
- TX Series Version 4.2.1 for Windows NT (which contains CICS for Windows NT Version 4.2.1)

Where it is necessary to distinguish between these products, the full product names and Version numbers are quoted.

The term *CICS on Open Systems* is used as a generic name for:

- Transaction Server for AIX[®] Version 4.1 (which contains CICS for AIX Version 2.1.1)
- TX Series Version 4.2 for AIX (which contains CICS for AIX Version 4.2)
- CICS for HP-UX Version 2.1.1
- TX Series Version 4.2 for HP-UX (which contains CICS for HP-UX Version 4.2)
- CICS for Solaris Version 2.1.1
- TX Series Version 4.2 for Solaris (which contains CICS for Solaris Version 4.2)

Where it is necessary to distinguish between these products, the full product names are quoted.

The term *CICS/VSE* means CICS for VSE/ESA Version 2 Release 3. The term *CICS Transaction Server for VSE/ESA* means CICS Transaction Server for VSE/ESA Release 1.

The term *CICS/400* means CICS/400 Version 3.

The notation **CICS–CICS for OS/2** is used to refer to communication in either direction. To specify communication in only one direction, an arrow is added. For example, CICS–CICS on Open Systems function shipping refers to function shipping from CICS to CICS on Open Systems or from CICS on Open Systems to CICS. CICS/400→CICS function shipping refers only to function shipping from CICS/400 to CICS.

Macro syntax notation

This section explains the syntax of the DFHCNV resource definition macros described in “Defining the conversion table” on page 59.

The operand field is used to specify the services and options to be generated. Operands are always in a keyword format and any parameters are specified according to the following general rules:

- If the parameter associated with the operand is shown entirely in capital letters (for example, TYPE=INITIAL), the operand and parameter must be specified exactly as shown.
- If the parameter associated with the operand is shown in lowercase letters (for example, CLINTCP=dddd), the operand must be specified exactly as shown and a value, address, or name must be substituted for the lowercase letters.
- Commas must be specified exactly as shown, except that a comma following the last operand specified must be omitted.
- Because a blank character indicates the end of the operand field, the operand field must not contain blanks except within quotes, after a comma on a continued line, or after the last operand of the macro. The first operand on a continuation line must begin in column 16.
- When a CICS macro is written on more than one line, each line containing part of the macro (except the last line) must contain a character (for example, an asterisk) in column 72, indicating that the macro is continued on the next line.

The symbols [], { }, |, and , ... are used in this book to show the operands for the macros as clearly as possible. *Do not use these symbols in your specifications.* They act only to indicate how a command or macro can be written; their definitions are given below:

- [] indicates optional operands. You may or may not need to specify the operand enclosed in the brackets (for example, [SRVERCP=nnnn]), depending on whether the associated option is desired.
- { } indicates that a choice must be made. One of the parameters from the list within braces separated by a | symbol (for example, {YES|NO}) must be specified, depending on which of the associated services is desired. Any default value is indicated like this: **default**.
- | indicates that a choice must be made between the operands or values that are separated by this symbol.
- , ... indicates that more than one set of values can be designated.

The default value of an operand is shown like this: **default**. For example:

```
PARAM={A|B|C}
```

means that you can code A, B, or C as the value for this operand. If you do not code the operand at all, the value A is assumed.

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.

Book structure

“Part 1. Communicating with non-System/390 CICS systems” on page 1
describes how to set up a CICS on System/390 system to communicate with non-System/390 CICS systems.

It contains the following chapters:

Chapter 1. Overview

summarizes intercommunication between System/390 and non-System/390 CICS products, listing the supported facilities, with a brief description of each.

Chapter 2. Planning

describes the areas where you have a choice to make when designing your System/390—non-System/390 communication links and applications.

Chapter 3. Resource definition

describes the CICS on System/390 resource definitions needed to support communication with a non-System/390 CICS system.

Chapter 4. CICS internal security

describes the security available on CICS on System/390—non-System/390 links, and the resource definitions needed to specify the level of security you want.

Chapter 5. VTAM/NCP definitions

describes the VTAM[®]/NCP definitions necessary to support intercommunication between System/390 and non-System/390 CICS products.

Chapter 6. Data conversion

describes the CICS on System/390 resource definitions needed to enable data conversion for communication with CICS for OS/2 or CICS on Open Systems. This chapter also lists the supplied user-replaceable data conversion program, which you can replace or modify if the supplied standard conversion program does not meet your needs.

“Part 2. Server Support for CICS Clients” on page 89

describes how to set up a CICS on System/390 system to act as a server to the CICS Clients workstation products.

It contains the following chapters:

Chapter 7. Introduction to CICS Clients

Chapter 8. Installation of server support for Clients

Chapter 9. Data conversion for Clients

Chapter 10. Application programming for Clients

Chapter 11. Problem determination for Clients

Chapter 12. Recovery after a restart of CICS

Chapter 13. Restrictions on Client support

Chapter 14. Migration considerations

Glossary of Client-related terms

Bibliography

This section lists those books in the System/390 and non-System/390 CICS libraries that are related to intercommunication.

Note: To help you find the information you need, some books are listed in more than one category.

CICS Family intercommunication books

CICS Family: Client/Server Programming, SC33-1435
CICS Family: Communicating from CICS on System/390, SC33-1697
CICS Family: Interproduct Communication, SC33-0824

CICS on System/390 intercommunication books

CICS Transaction Server for OS/390 Release 3

CICS Distributed Transaction Programming Guide, SC33-1691-02
CICS External Interfaces Guide, SC33-1944-01
CICS Front End Programming Interface User's Guide, SC33-1692-02
CICS Intercommunication Guide, SC33-1695-02
CICS Internet Guide, SC34-5445-00

CICS Transaction Server for OS/390 Release 2

CICS Distributed Transaction Programming Guide, SC33-1691-01
CICS Front End Programming Interface User's Guide, SC33-1692-01
CICS Intercommunication Guide, SC33-1695-01
CICS Internet and External Interfaces Guide, SC33-1944-00

CICS Transaction Server for OS/390 Release 1

CICS Distributed Transaction Programming Guide, SC33-1691-00
CICS External CICS Interface, SC33-1703-00
CICS Front End Programming Interface User's Guide, SC33-1692-00
CICS Intercommunication Guide, SC33-1695-00
CICS ONC RPC Guide, SC33-1778-02
CICS Web Interface Guide, SC33-1825-00

CICS/ESA Version 4

CICS ONC RPC Feature for MVS/ESA Guide, SC33-1119
CICS Web Interface Guide, SC33-1892
Distributed Transaction Programming Guide, SC33-1174
External CICS Interface, SC33-1390
Front End Programming Interface User's Guide, SC33-1175
Intercommunication Guide, SC33-1181

CICS/ESA Version 3

Distributed Transaction Programming Guide, SC33-0783
Front End Programming Interface User's Guide, SC33-0804
Intercommunication Guide, SC33-0657

CICS Transaction Server for VSE/ESA Release 1

Distributed Transaction Programming Guide, SC33-1661

External CICS Interface, SC33-1669

Front End Programming Interface User's Guide, SC33-1662

Intercommunication Guide, SC33-1665

CICS/VSE Version 2

Distributed Transaction Programming Guide, SC33-0898

Intercommunication Guide, SC33-0701

Server Support for CICS Clients, SC33-1712

CICS non-System/390 intercommunication books

CICS for OS/2 Intercommunication, SC33-1583

CICS for Windows NT Intercommunication, SC33-1423

CICS on Open Systems Intercommunication Guide, SC33-1564

CICS/400 Intercommunication, SC33-1388

CICS Clients

CICS Clients: Administration, SC33-1436

CICS Family: Client/Server Programming, SC33-1435

CICS/VSE Version 2 Release 3 Server Support for CICS Clients, SC33-1712

Summary of changes

This book is based on the second edition of the *Communicating from CICS on System/390* manual, SC33-1697-01. Changes from the second edition are marked by vertical bars in the left margin.

The major changes for this edition are:

- The book has been revised to take account of the following new products:
 - CICS for HP-UX Version 2.1.1
 - CICS for Solaris Version 2.1.1
 - CICS Transaction Server for OS/2 Warp Version 4.1
 - **CICS Transaction Server for OS/390 Release 3**
 - **CICS Transaction Server for VSE/ESA Release 1**
 - Transaction Server for AIX Version 4.1
 - Transaction Server for OS/2 Warp Version 4.0
 - Transaction Server for Windows NT Version 4.0
 - TX Series Version 4.2 for AIX
 - TX Series Version 4.2 for HP-UX
 - TX Series Version 4.2 for Solaris
 - TX Series Version 4.2 for Windows NT
 - TX Series Version 4.2.1 for Windows NT
- The tables of client and server code pages in “Chapter 6. Data conversion” on page 43 have been updated.
- The tables of client and server code pages in “Chapter 9. Data conversion for Clients” on page 109 have been updated.
- References to the following products have been removed, because they are no longer supported:
 - CICS/MVS® Version 2 (all Releases)
 - CICS/VSE Version 2 Release 2
 - CICS/6000® Version 1 (all Releases)
 - CICS/400 Version 2
 - CICS OS/2™ Version 2
 - CICS OS/2 Version 2.0.1
 - CICS for Windows NT Version 2.

Changes for the second edition

- The book was revised to take account of the CICS Transaction Server for OS/390 Release 2 product.
- “Part 2. Server Support for CICS Clients” on page 89 was added. This part of the book amalgamates information previously published separately, in the CICS/ESA Version 4.1 and CICS Transaction Server for OS/390 Release 1 editions of the *Server Support for CICS Clients* manual. It describes how to set up a CICS on System/390 system to act as a server to the CICS Clients workstation products.
- “TCP/IP support” on page 93 and “Using TCP/IP” on page 96 were added; these sections describe how some CICS on System/390 products can communicate with some types of CICS Client over a TCP/IP network.
- References to CICS/ESA Version 3 Release 2.1 were removed, because this product is no longer supported.

Changes for the first edition

- The book was based on the second edition of the *Communicating from CICS/ESA and CICS/VSE* manual, SC33-0825-01. It was renamed to the more generic *Communicating from CICS on System/390*, to reflect the multiplicity of CICS products that run on System/390.
- The book was revised to take account of the following new CICS products:
 - CICS Transaction Server for OS/390 1.1
 - CICS/VSE Version 2 Release 3
 - CICS for OS/2 Version 2.0.1
 - CICS for OS/2 Version 3
 - CICS for Windows NT Version 2
 - CICS on Open Systems. This comprises:
 - IBM AIX CICS/6000 Version 1 Release 2
 - IBM CICS for AIX Version 2 Release 1
 - CICS for Digital UNIX™
 - CICS for HP 9000.
- The description of the data conversion process in “Chapter 6. Data conversion” on page 43 was rewritten.

Part 1. Communicating with non-System/390 CICS systems

This part of the book describes how to set up a CICS on System/390 system to communicate with non-System/390 CICS systems.

Note: CICS on System/390's support for the CICS Clients workstation products is described in "Part 2. Server Support for CICS Clients" on page 89.

Chapter 1. Overview

Note

In this chapter and throughout this book, the generic term **CICS on System/390** represents all CICS System/390 products.

Interproduct communication between System/390 and non-System/390 CICS systems uses the Systems Network Architecture (SNA) LU 6.2 protocol. The descriptions in this chapter assume LU 6.2 links.

Below is a list of the supported facilities, with a page reference to a discussion of each. Every CICS on System/390 product supports all these facilities.

Function shipping (see page 3)

File control, temporary storage, transient data, syncpoint, and interval control requests can be shipped in either direction between System/390 and non-System/390 CICS systems.

Transaction routing (see page 4)

Transactions can be routed over an LU 6.2 link in either direction between System/390 and non-System/390 CICS systems.

Distributed program link (DPL) (see page 6)

DPL enables CICS application programs to issue EXEC CICS LINK commands in either direction between System/390 and non-System/390 CICS systems.

Asynchronous processing (see page 8)

Asynchronous processing is supported between System/390 and non-System/390 CICS systems. The initiating request can flow in either direction.

Distributed transaction processing (see page 8)

Distributed transaction processing, using mapped conversations, is supported between System/390 and non-System/390 CICS systems. The initiating request can flow in either direction.

Function shipping

Non-System/390 CICS application programs can access resources (data or transactions) owned by a CICS on System/390 system, and a CICS on System/390 application can access resources owned by a non-System/390 CICS system, in each case provided that the resources are defined as remote in the function shipping system.

A function shipping request takes the form of a normal EXEC CICS command. If either of the following conditions applies, the application-owning system recognizes that function shipping is required and ships the request to the remote resource-owning system.

1. The EXEC command specifies a remote system in the SYSID option.
2. The resource is defined as remote.

The mirror program in each CICS product (DFHMIRS in CICS on System/390) handles inbound function shipping.

function shipping

As already noted, the LU 6.2 protocol is used for all communication between System/390 and non-System/390 CICS systems. Synchronization level 2 on LU 6.2 links is supported by CICS on Open Systems, CICS/400 Version 3, and CICS for Windows NT.¹ It is *not* supported by CICS for OS/2. Synchronization level 1 is supported for function shipping between all non-System/390 and System/390 CICS systems. See “Syncpointing (LU 6.2)” on page 15.

Restrictions on function shipping

There are some restrictions on function shipping between System/390 and non-System/390 CICS systems.

CICS non-System/390→CICS on System/390

Function shipping the following sequence of commands from a non-System/390 CICS to a CICS on System/390 system causes the System/390 mirror transaction to abend:

```
DELETEQ TS Q(RFRED)
WRITEQ TS Q(RFRED) FROM()
SYNCPOINT
```

This is because, on CICS on System/390, you cannot delete a temporary storage queue and then write to it, without issuing a syncpoint between the two commands.

CICS for OS/2

Function shipping to and from CICS for OS/2 over LU 6.2 links requires CICS for OS/2 to be running with Communications Manager/2.

DL/I database access

Non-System/390 CICS systems cannot function-ship requests to DL/I databases accessed through CICS on System/390 systems. To access DL/I databases from CICS for OS/2, CICS for Windows NT, CICS on Open Systems, or CICS/400, use distributed transaction processing or the distributed program link function.

Data conversion

CICS for OS/2, CICS for Windows NT, and CICS on Open Systems use ASCII² data representation and CICS/400 and CICS on System/390 systems use EBCDIC³. When conversion is necessary, the ASCII-based system always converts system data such as resource names. Conversion of user data is performed as necessary in the resource-owning system. For example, for CICS for OS/2→CICS function shipping, CICS converts the user data (see Table 2 on page 43). For CICS→CICS for OS/2 function shipping, CICS for OS/2 converts the user data.

Transaction routing

Transaction routing enables a terminal in one CICS system to run with a transaction in another CICS system. The typical way to initiate transaction routing is by entering a remote transaction ID at a local terminal. For other ways, see the *CICS Family: Interproduct Communication* manual.

Transactions can be routed in either direction over an LU 6.2 link between any CICS non-System/390 system and any CICS on System/390 system.

1. For details of how to set up CICS for Windows NT support, see your *CICS for Windows NT Intercommunication* manual.

2. American National Standard Code for Information Interchange

3. Extended Binary-Coded Decimal Interchange Code

Transaction routing to and from CICS for OS/2 requires CICS for OS/2 to be running under OS/2 Version 2 with Extended Services or Communications Manager/2.

CICS on System/390→CICS non-System/390

For transaction routing from a CICS on System/390 system, CICS on System/390 requires a remote definition of the non-System/390 transaction. The REMOTESYSTEM name must be the name of the connection to the non-System/390 system.

CICS on System/390 requires a local definition of the terminal from which the transaction is routed. If this definition is not shippable, the non-System/390 CICS requires a remote definition of the terminal (see the *Intercommunication Guide* or equivalent for the non-System/390 system).

CICS non-System/390→CICS on System/390

For transaction routing from CICS for OS/2, CICS for Windows NT, CICS on Open Systems, or CICS/400, the non-System/390 CICS requires a remote definition of the CICS on System/390 transaction (see the *Intercommunication Guide* or equivalent for the non-System/390 system).

The non-System/390 CICS requires a local definition of the terminal from which the transaction is routed. Most terminal definitions can be shipped from the non-System/390 CICS to CICS on System/390. However, definitions of non-VTAM terminals cannot be shipped. CICS on System/390 requires remote definitions of such terminals.

The remote definition of a non-System/390 terminal has the following characteristics:

- REMOTESYSTEM is the System/390 name of the connection to the non-System/390 system.
- A subset of the 3270 extended data stream is supported.
- The color, highlight, programmable symbols (PS), and outline extended attributes are supported but only to the extent that they are generated by BMS. (A field may have these attributes but an individual character within a field cannot have separate attributes.)
- TCTUALENG must be the same as TCTUAL in the non-System/390 system TCT definition for the terminal.
- REMOTENAME is the terminal ID in the non-System/390 TCT definition for the terminal. ⁴

Data conversion

CICS on System/390 systems do no data conversion for transaction routing. Screen data always flows as 3270 data streams. COMMAREAs and TCTUAs (which are relevant to pseudoconversational transactions) are converted by the ASCII system.

4. In a network of CICS for OS/2 or CICS for Windows NT systems, they can all use the same terminal names. A transaction routing exit allows the changing of shipped terminal names to unique values, called terminal shipping aliases. If a terminal shipping alias is used, it must be the name in the CICS remote definition of the terminal.

transaction routing

Transaction routing restrictions

There are some restrictions on transaction routing support, as follows:

- BMS paging is not supported.
- The fully qualified network name of CICS for OS/2 or CICS for Windows NT is not available in the CICS monitoring record for LU 6.2 links.
- CICS for OS/2 should not invoke CICS on System/390 transactions defined with message protection options that cannot be honored on synclevel 1 links—that is, MSGINTEG(YES) or PROTECT(YES).

Distributed program link (DPL)

Distributed program link (DPL) enables an application program in a local CICS system to issue an EXEC CICS LINK command to link to a program in a remote CICS system, which returns control to the calling program.

CICS on System/390 supports both inbound and outbound DPL with all current non-System/390 CICS systems.

Distributed program link:

- Provides a way for non-System/390 CICS systems to access DL/I and SQL databases and BDAM files owned by a System/390 CICS system, and allows existing System/390 programs to be used on the data. (Another way of accessing this data is to use distributed transaction processing—see page 8.)
- Provides improved performance for a distributed system consisting of multiple CICS systems. For example, a single link can achieve a data set browse that would require multiple flows if function shipping were used.
- Gives the non-System/390 CICS programmer access to programs that cannot be ported from the System/390 system—for example, macro-level programs.
- Allows a CICS programmer to use an LU 6.2 link without needing to know the protocol.

Restrictions on programs linked by DPL

In CICS on System/390, the linked program runs under the mirror transaction, using that transaction's attributes, for example, task priority, security attributes, and keys.

A CICS program linked by a program in a remote CICS system cannot issue:

- Terminal control commands to the initiating CICS system
- Commands that inquire on terminal attributes
- BMS commands
- Commands that address the TWA or the TCTUA (use the COMMAREA to pass data between systems)
- SIGNON and SIGNOFF (CICS Transaction Server for OS/390 and CICS/ESA only).

A DPL server program on CICS on System/390 terminates with transaction abend code ADPL if it issues one of the restricted commands listed above.

Note for DB2®

When DB2 data is accessed from CICS for OS/2, CICS for Windows NT, CICS 400, or CICS on Open Systems, security access is based on the TRANSID passed to the System/390 mirror transaction.

The System/390 EIBTRNID field is set to the transaction ID passed by the remote CICS system, and this is used for the duration of the link. This mechanism allows greater selectivity for DB2 plans.

Abends when using DPL

If the linked CICS program terminates abnormally and doesn't handle the abend itself, the mirror program returns an abend code. The code returned is that which would have been returned by an ASSIGN ABCODE command. Note that the abend code returned to the linking CICS system is the last abend to occur in the mirror program, which may have handled other abends before terminating.

Performance optimization for DPL

The performance of DPL may be affected by the amount of data transmitted, which includes the optional COMMAREA specified on an EXEC CICS LINK command. The length of the COMMAREA is in the range 1–32 767.

CICS on System/390 and the other CICS products contain algorithms designed to reduce the number of bytes to be transmitted. The algorithms remove some trailing binary zeros from the COMMAREA before transmission and restore them after transmission. The operation of these algorithms is transparent to the application programs, which always see the full-size COMMAREA.

When transmission time accounts for a significant part of the response time at a user terminal or workstation, application programs may be able to improve performance by using the DATALENGTH parameter in the LINK command. This parameter specifies a contiguous area of storage, at the start of the COMMAREA, to be passed to the invoked program. For example, if all the data to be transmitted is grouped in the first 100 bytes of a 30 000-byte COMMAREA, and DATALENGTH(100) is specified, only the first 100 bytes are transmitted.

Dynamic routing of DPL requests

CICS Transaction Server for OS/390 Release 3 allows you to route DPL requests dynamically. In CICS TS 390 Release 3, if a program defined as DYNAMIC is the subject of an EXEC CICS LINK command, the CICS dynamic routing program is invoked, and can select a remote region on which the program is to execute.

CICS TS 390 Release 3 can dynamically route:

- EXEC CICS LINK commands issued locally.
- DPL calls received from other CICS regions. The other CICS regions do not have to be CICS TS 390 Release 3 systems. This means that other CICS products can benefit from the enhanced workload balancing capabilities of CICS TS 390 Release 3.
- Any type of program-link request received from outside CICS. For example, CICS Transaction Server for OS/390 Release 3 can dynamically route any of the following:
 - Calls received from:
 - The CICS Web interface
 - The CICS Gateway for Java™
 - Calls from external CICS interface (EXCI) client programs
 - External call interface (ECI) calls from any of the CICS Client workstation products
 - Distributed Computing Environment (DCE) remote procedure calls (RPCs)
 - ONC RPC calls.

distributed program link

For definitive information about how to route program-link requests dynamically, see the *CICS Transaction Server for OS/390 Release 3 Intercommunication Guide*.

Asynchronous processing

Asynchronous processing is a form of intercommunication in which one transaction initiates another, and the two transactions then run independently of each other (that is, asynchronously).

Asynchronous processing is initiated when a transaction issues a START command for a remote transaction. The issue of the START command can be regarded as a special case of function shipping, in which the shipped request is a START command. Data passed by the starting transaction can be accessed by the started transaction using the RETRIEVE command.

A transaction can initiate and communicate synchronously with a remote transaction and then terminate, leaving the initiated transaction to continue asynchronously. The original initiating request can flow in either direction between System/390 and non-System/390 CICS systems.

Distributed transaction processing (DTP)

Distributed transaction processing (DTP) enables transactions running in one CICS system to initiate and communicate synchronously with transactions in another CICS system. DTP is supported between CICS on System/390 products and each of the non-System/390 CICS products. The initiating transaction can be in either the System/390 or non-System/390 CICS system. Sync level 1 is the maximum synclevel for CICS for OS/2 links.

DTP is an alternative to DPL as a way for other CICS systems to access DL/I and DB2 databases owned by a CICS on System/390 system that has a database-handling transaction.

Application programs can issue CICS commands for APPC conversations and so control the allocation and use of an APPC session. To do this, a program must be aware of the state of the conversation over the intersystem link at any given time.

The EXEC CICS commands used to control an APPC conversation are:

ALLOCATE, CONNECT PROCESS, EXTRACT PROCESS, SEND, RECEIVE, CONVERSE, WAIT, ISSUE CONFIRMATION, ISSUE ERROR, ISSUE ABEND, FREE.

Summary of CICS System/390–non-System/390 intercommunication

Table 1 on page 9 shows the communication functions that a CICS System/390 product can support on links between itself and each non-System/390 CICS system-level product.

Note: CICS on System/390's support for the CICS Clients workstation products is described in "Part 2. Server Support for CICS Clients" on page 89.

If a function is shown as supported in the table, it means that:

1. The function is supported on all current System/390 products (CICS Transaction Server for OS/390, CICS Transaction Server for VSE/ESA, CICS/ESA, and CICS/VSE).
2. Both inbound and outbound requests are supported.

All functions are supported on LU 6.2 connections only. Data conversion, where necessary, is supported at each end of the link.

Table 1. CICS interproduct communication

CICS TS 390, CICS TS VSE/ESA, CICS/ESA, CICS/VSE	CICS for OS/2	CICS for Windows NT	CICS on Open Systems	CICS/400
Function shipping	Yes	Yes	Yes	Yes
Transaction routing	Yes	Yes	Yes	Yes
Distributed program link	Yes	Yes	Yes	Yes
Distributed transaction processing	Yes	Yes	Yes	Yes
Asynchronous processing	Yes	Yes	Yes	Yes

summary

Chapter 2. Planning

CICS interproduct communication requires planning and setup at both ends. CICS on System/390 planners should consult the planners of all the systems with which CICS on System/390 is to communicate. This chapter discusses the areas where decisions must be made.

Path length and resource definition tradeoffs

If more than one System/390 CICS system is communicating with an ASCII system—that is, with CICS for OS/2, CICS for Windows NT, or CICS on Open Systems (CICS/400 uses the same EBCDIC code page as CICS on System/390)—direct and indirect links are possible. The links used affect resource definition effort and processing workload. The processing workload includes data transfer and data conversion.

Where user data conversion is performed by the System/390, it occurs at the first System/390 system for data inbound from the ASCII system, and at the last System/390 system for data outbound to the ASCII system. In Figure 1, an ASCII system running CICS for OS/2 is linked to two CICS on System/390 systems, directly to CICS1 and indirectly, through CICS1, to CICS2. CICS1 and CICS2 can be connected in any way supported for the particular products. Whatever the connection between CICS1 and CICS2, CICS1 does the conversion for data transferred in either direction.

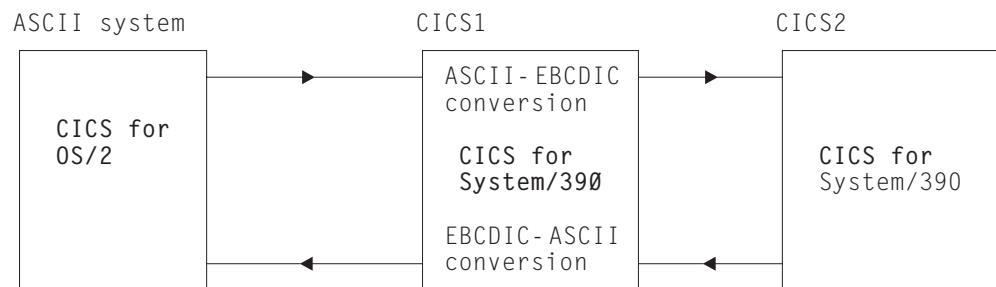


Figure 1. Where data conversion takes place

Figure 2 on page 12 shows an ASCII system, CICS for OS/2, CICS for Windows NT, or CICS on Open Systems, and four CICS System/390 systems. This figure is the basis of the discussion in the rest of this chapter.

The CICS on System/390 systems comprise:

- One terminal-owning region (TOR)
- Two application-owning regions (AOR1 and AOR2)
- One data-owning region (DOR).

The ASCII system can have a separate LU 6.2 link to each System/390 system. Figure 2 shows three such links: link X to TOR, link Y to AOR1, and link Z to DOR.

Assumptions

Under “Possible approaches” on page 12, the following assumptions are made about Figure 2 on page 12.

1. A user of the ASCII system can enter a transaction (TRN1) owned by system AOR1 that requires access to:

planning

- Temporary storage (TS) queues in DOR
 - Transient data (TD) queues in systems AOR1, AOR2, and DOR
 - File control (FC) files in DOR.
2. Function shipping can take place from the ASCII system directly to TOR, AOR1, and DOR, and indirectly to AOR2.

Note: Some or all of these requests may require data conversion.

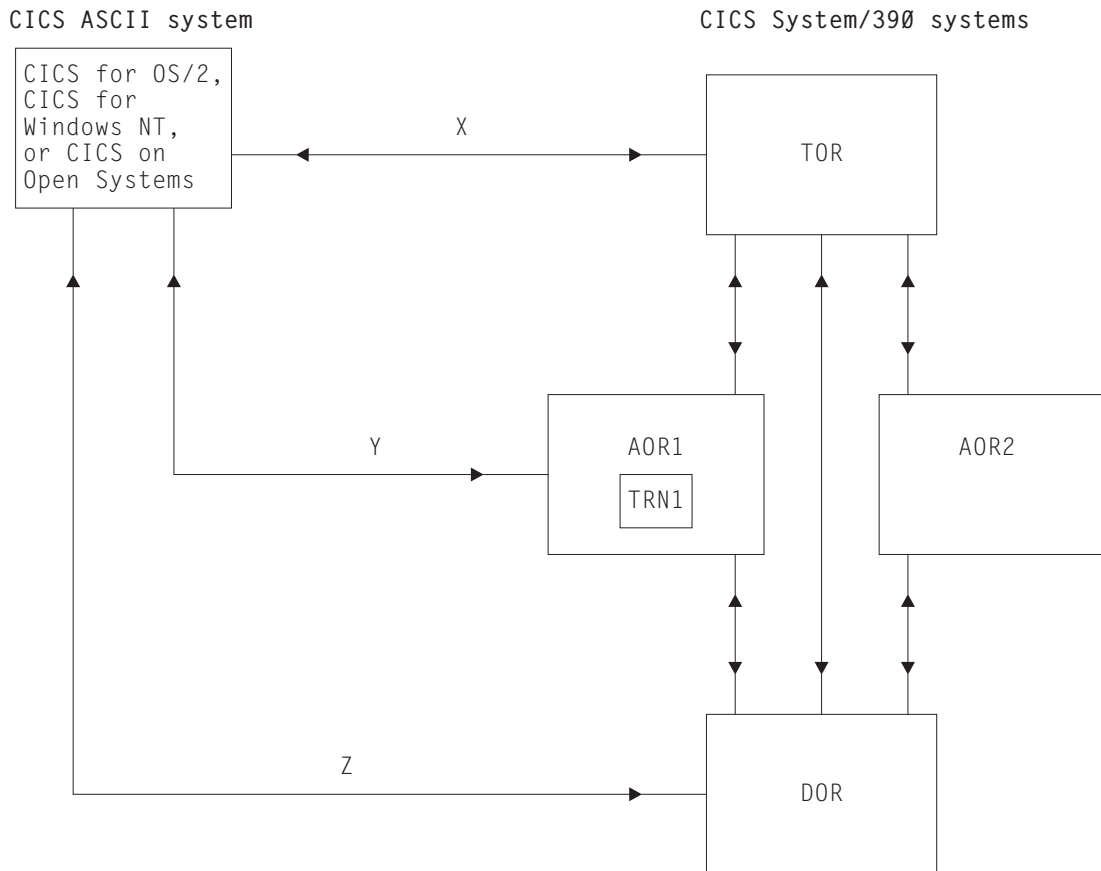


Figure 2. Sample configuration of System/390 and non-System/390 CICS systems

Possible approaches

With the setup in Figure 2, and the stated assumptions, various scenarios are possible, as discussed below.

Note: In the discussion, the term “data conversion modules” refers to the standard data conversion program, the data conversion table, and, if necessary, the user-replaceable data conversion program.

Transaction routing: ASCII—TOR—AOR1

The following definitions are necessary:

- In the ASCII system, a remote definition of the transaction (the remote system is specified as TOR).
- In TOR:
 - A remote terminal definition (or a shipped terminal definition)
 - A remote definition of the transaction (the remote system is specified as AOR1).

- In AOR1:
 - A remote terminal definition (or a shipped terminal definition).
 - Remote definitions of the files owned by DOR that are to be accessed by the transaction.
 - Remote definitions of the temporary storage queues in DOR.
 - Remote definitions of the transient data destinations in DOR.
 - Local transient data definitions.
 - Local transaction and program definitions.
 - If AOR1 is a CICS/VSE Version 2 or a pre-4.1 CICS/ESA system, an indirect connection to the ASCII system, via TOR. Indirect connections are required only for transaction routing across intermediate systems. For information about defining indirect connections, see the *Intercommunication Guide* for your CICS System/390 product.
- In DOR:
 - Local transient data definitions
 - Local temporary storage definitions
 - Local file definitions.

The ASCII system does the data conversion from ASCII to EBCDIC. If transaction routing occurs from more than one CICS for OS/2 or CICS for Windows NT workstation, the data conversion modules must be installed in each workstation.

Transaction routing: ASCII—AOR1

The same resource definitions are required as for transaction routing through TOR (see above), except that:

- In the ASCII system, on the remote transaction definition, the remote system is specified as AOR1
- In TOR, the remote terminal and transaction definitions are no longer necessary
- In AOR1, the indirect connection to the ASCII system is no longer necessary.

Function shipping: ASCII—TOR—AOR1—DOR

The following definitions are necessary:

- In the ASCII system, remote definitions of the resources to be accessed
- In TOR:
 - Remote definitions of the resources to be accessed
 - Definitions of the data conversion modules.
- In AOR1:
 - Local definitions of its own resources
 - Remote definitions of resources owned by DOR, that are to be accessed by the ASCII system.
- In DOR, local definitions of its own resources.

The data conversion modules need to be defined in only one system, TOR, which does the ASCII↔EBCDIC conversion on the transmitted user data.

Function shipping: ASCII—AOR1—DOR

The same resource definitions are required as for the previous example, except that:

- The definitions in TOR are not required
- The data conversion module definitions are in AOR1, which does the ASCII↔EBCDIC conversion on the transmitted user data.

planning

Function shipping: ASCII—AOR1 and ASCII—DOR

The same resource definitions are required as for the previous example, except that:

- The remote resource definitions are not required in AOR1
- AOR1 and DOR each do ASCII↔EBCDIC conversion on transmitted user data, depending on which system is the target of each function-shipped request. You must therefore define the data conversion modules in both AOR1 and DOR.

Summary

A direct link from the workstation to the target CICS system gives the shortest path length. If you have several target CICS on System/390 systems, you can ship all requests through a single system in which you have defined the data conversion modules. This enables you to define the data conversion modules in only one place, at the expense of a longer path length and the need to create more remote resource definitions.

Syncpointing (LU 6.2)

Synchronization level 2 is supported on LU 6.2 links by CICS on Open Systems, CICS/400 Version 3, and CICS for Windows NT.⁵ It is *not* supported by CICS for OS/2. LU 6.2 sessions between CICS for OS/2 and CICS on System/390 are bound at synchronization level 1 (synclevel 1), which allows the exchange of private synchronization requests and responses, but not the use of CICS syncpointing commands. However, for function shipping, CICS has defined session-local protocols that allow limited use of syncpoint commands. The effects of this are:

- Session-local protocols are needed to coordinate changes made in both systems.
- CICS cannot guarantee resource integrity after a session or system failure.

CICS has defined session-local protocols for use when function shipping occurs at synclevel 1. The following examples describe the use of these protocols.

Function shipping from CICS on System/390 to CICS for OS/2

A and B are two CICS on System/390 systems communicating with each other at synclevel 2 (SL 2 in the figure). B communicates at synclevel 1 (SL 1 in the figure) with three CICS for OS/2 systems, X, Y, and Z.

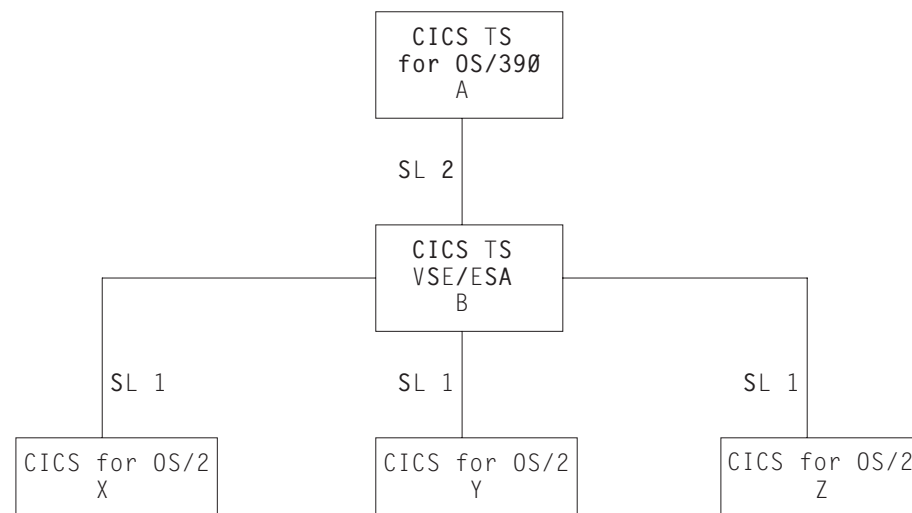


Figure 3. Synchronization between CICS on System/390 and CICS for OS/2. CICS for OS/2 systems do not support synclevel 2 over LU 6.2 links.

Provided there are no failures (transaction, session, or LU (either end of each link)) during the commit stage (while recoverable resource changes are being committed by each partner), data integrity is assured at either end of each link.

If there is a failure during the A-B commit stage, all resource updates can be backed out. However, if there is a failure during the commit stage on the B-X, B-Y, or B-Z sessions, the system requesting synchronization has no way of knowing if its partner has committed the changes to its resources.

The implications of this are best explained by an example.

5. For details of how to set up CICS for Windows NT synclevel 2 support, see your *CICS for Windows NT Intercommunication* manual.

syncpointing

Example

A transaction in B (defined with the INDOUBT(BACKOUT) attribute) is function shipping file update requests to A, X, Y, and Z. The function-shipping transaction requests synchronization by issuing an EXEC CICS RETURN command. CICS commits the changes on the synclevel 2 session (B-A) first, and then the changes on each synclevel 1 session in turn.

If a failure occurs during the B-A commit stage, the changes on all systems are rolled back. If a failure occurs during the commit stage on any of the synclevel 1 links (B-X, B-Y, or B-Z), a message is issued, and synclevel 1 processing continues on the remaining links, with the aim of committing as many synclevel 1 resources as possible. *User-defined procedures are needed to resynchronize function-shipped updates.*

Suggestion

If you use function shipping between CICS on System/390 and CICS for OS/2, you incur the risk described here. If this risk is unacceptable, you should use distributed transaction processing. You can then build your own integrity into the programs at either end of the link.

DPL or function shipping from CICS for OS/2 to CICS on System/390

DPL and function shipping incorporate synclevel 1 logic. The CICS for OS/2 system initiates the commit procedure by requesting CICS on System/390 to commit data changes. The CICS for OS/2 system then commits the changes itself when it receives confirmation of CICS on System/390 commitment.

Figure 4 illustrates synclevel 1 support in a simple example. You can, of course, have a number of connected CICS systems. In the case of multiple connected systems, the commit request is propagated through all the system connections.

When using DPL, you should take syncpoints from the CICS for OS/2 system. If you take a syncpoint in the linked-to System/390 program, the syncpoint request is not propagated back to the CICS for OS/2 system.

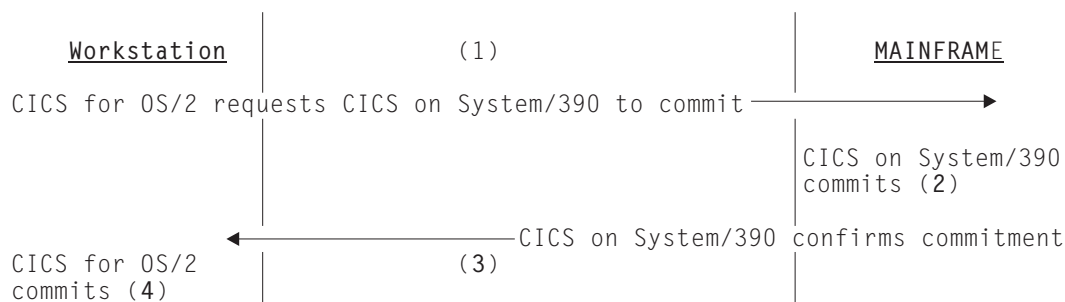


Figure 4. Sync level 1 logic in a CICS for OS/2—CICS on System/390 link

Chapter 3. Resource definition

This chapter describes the resource definitions needed in a CICS on System/390 system for communication with non-System/390 CICS products. SNA LU 6.2 (APPC) links are used.

Examples in this chapter

This chapter consists mainly of example resource definitions using the CEDA transaction. Note that:

- The examples do not show complete CEDA screens, but only options that are relevant to intercommunication.
- Some options are not available or relevant to all CICS System/390 products. Where this applies, it is noted in the text following the example screen.

For further information about the intercommunication options available on your System/390 system, see the *Intercommunication Guide* for your CICS on System/390 product.

- The examples are intended as generic definitions that will work on all CICS on System/390 systems. As such, options that are unavailable on some products, or that are retained solely for CSD-compatibility with earlier CICS releases, are given default values, or are not specified.

For definitive information about coding the intercommunication options, see the *Resource Definition Guide*⁶ for your CICS on System/390 product.

- If possible, you should use the same group name for all definitions associated with a particular remote system. The examples use the name CICS0S2.

6. *Resource Definition (Online)* in earlier releases

System generation and initialization

For CICS/VSE Version 2 systems, code the following operands of the DFHSG PROGRAM=TCP macro instruction (used during system generation and customization to create the terminal control program) with the values shown:

- ACCMETH=VTAM, to support VTAM
- CHNASSY=YES, to support SNA chain assembly
- VTAMDEV=(..,LUTYPE6,..) to identify the type of link you are using.

Set the following fields in the CICS system initialization table (SIT):

- ISC=YES, to include the intercommunication programs
- For CICS/VSE Version 2 only, EXEC=YES, to support command-level programs (CICS for OS/2, CICS for Windows NT, CICS/400, and CICS on Open Systems, like CICS TS 390, CICS Transaction Server for VSE/ESA, and CICS/ESA, do not support macro-level programs).

LU 6.2 link

You must use an LU 6.2 link for all the facilities described in “Chapter 1. Overview” on page 3. Depending on the intercommunication functions being used, create resource definitions similar to those required on a link between two System/390 CICS systems.

All communication with CICS non-System/390 systems

Define the following:

- A communications profile. (You can use the supplied profile, DFHCICSA, which is defined in the supplied group DFHSTAND.)
- For single-session links, the remote CICS system, using TERMINAL and TYPETERM definitions; see pages 23 and 25.
- For parallel sessions, CONNECTION and SESSIONS definitions, see pages 26 and 27.

No CONNECTION or SESSIONS definitions are necessary for single-session links, which can be completely defined by TERMINAL and TYPETERM definitions. The terminal name (for example, PSO2 on page 23) serves as the connection name.

If you require parallel-session links, omit the TERMINAL and TYPETERM definitions. The CONNECTION and SESSIONS definitions support both types of link.

If you are using CICS internal security (CICS/VSE Version 2 only), and you want IDENTIFY or VERIFY attach-time security, use the DFHSNT resource definition macro to create an SNT entry for each non-System/390 user who accesses CICS/VSE resources. If you are using an external security manager (ESM), such as RACF®, define user profiles to your ESM instead. See “Chapter 4. CICS internal security” on page 31.

Function shipping and DPL from CICS non-System/390 systems

Define the following:

- The mirror transaction (the name varies with product and release); see page 20.
- The mirror program (the name varies with product and release); see page 20.
- If the remote system is CICS on Open Systems, CICS for Windows NT, or CICS for OS/2, the data conversion program DFHCCNV, the data conversion table DFHCNV, and, if you need nonstandard conversion, the user-replaceable conversion program DFHUCNV; see page 22.

(The above three items are defined in the supplied group DFHISC.)

An incoming START command can use the TERMID option to specify a terminal that is to be associated with the started transaction. This terminal must be defined.

Function shipping and DPL to CICS non-System/390 systems

For function shipping, define the remote resource (file, temporary storage queue, or transient data queue).

For function shipping of a START command, define the remote program, unless the LINK or START command includes the SYSID option.

LU 6.2 definitions

Transaction routing from CICS non-System/390 systems

Create remote terminal definitions for those terminals whose definitions cannot be shipped from the non-System/390 system. Examples of such terminals are 3270 terminals or printers defined with the attribute SHIPPABLE(NO).

Transaction routing to CICS non-System/390 systems

Define the remote transaction. Unless you are using dynamic transaction routing, specify the REMOTESYSTEM value as the name of the connection to the transaction-owning system.

Defining remote resources

The following pages give examples of the definitions required to support intercommunication connections and functions at the system level. When defining a remote resource for a specific application (program, transaction, file, queue, or terminal), the REMOTESYSTEM value is the name of the connection to the resource-owning system. For an example of a remote resource definition, see page 28.

CICS on System/390 mirror transaction definition

Most of the values below must be entered as shown.

The profile name shown is that of an IBM-supplied profile. If you create your own profile, change this name. In your own profile, specify INBFMH (All), which is required for function shipping and distributed transaction processing.

```
DEFINE TRANSACTION(CPMI) GROUP(CICSOS2)
OVERTYPE TO MODIFY
CEDA DEFINE
Transaction      : CPMI
Group            : CICSOS2
PROGram         ==> DFHMIRS
TWAsize         ==> 00000           0-32767
PROFile         ==> DFHCICSA
PArTitionset    ==>
STatus          ==> Enabled       Enabled|Disabled
PRIMEdsize     ==> 00000         0-65520
SCHEDULING
PRIOrity       ==> 001           0-255
TCLass        ==> No             No|1-10
ALIASES
```

If you use the TRANSID option of the EXEC CICS LINK command to specify a different mirror transaction, you need to copy this definition, change the TRANSACTION name and, possibly, change the values of TWASIZE, PRIMEDSIZE, PRIORITY, and TCLASS.

CICS on System/390 mirror program definition

Enter all values as shown.

For CICS/VSE Version 2 systems that use CICS internal security, the RSL value of PUBLIC means that no CICS security restrictions are placed on the use of this program. If you wish to restrict its use, enter a number in the range 1 through 24. A value of 0 denies use of this program to any transaction defined with RSLC(YES). For details of the RSL and RSLC attributes, refer to the *CICS/VSE Version 2 Release 3 Resource Definition Guide*.

```

DEFINE PROGRAM(DFHMIRS) GROUP(CICS0S2)
OVERTYPE TO MODIFY
CEDA DEFine
PROGram      ==> DFHMIRS
Group        ==> CICS0S2
Language     ==> Assembler      Cobol|Assembler|Le370|C|Pli
                                         |Rpg
RELoad       ==> No              No|Yes
RESident     ==> No              No|Yes
RSL          ==> Public          0-24|Public
Status       ==> Enabled         Enabled|Disabled
    
```

The following notes apply to all program definition screens.

Notes:

1. The RSL option relates to CICS internal security, and does not apply to CICS TS 390, CICS Transaction Server for VSE/ESA, or CICS/ESA. For compatibility with earlier releases, it appears on a CEDA display, but is set to 00 and cannot be changed.
2. The RPG language is supported by CICS/VSE Version 2 only.

LU 6.2 definitions

Data conversion program definition

Enter all values as shown.

```
DEFINE PROGRAM(DFHCCNV)
OVERTYPE TO MODIFY
CEDA DEFine
PROGram    ==> DFHCCNV
Group      ==> CICSOS2
Language   ==> Assembler      Cobol|Assembler|Le370|C|Pl i
                                   |Rpg
RELoad     ==> No              No|Yes
RESident   ==> No              No|Yes
RS1        ==> 00              0-24|Public
Status     ==> Enabled         Enabled|Disabled
```

Data conversion table definition

Enter all values as shown.

```
DEFINE PROGRAM (DFHCCNV)
OVERTYPE TO MODIFY
CEDA DEFine
PROGram    ==> DFHCCNV
Group      ==> CICSOS2
Language   ==> Assembler      Cobol|Assembler|Le370|C|Pl i
                                   |Rpg
RELoad     ==> No              No|Yes
RESident   ==> No              No|Yes
RS1        ==> 00              0-24|Public
Status     ==> Enabled         Enabled|Disabled
```

User-replaceable conversion program definition

Enter all values as shown.

```
DEFINE PROGRAM (DFHUCNV)
OVERTYPE TO MODIFY
CEDA DEFine
PROGram    ==> DFHUCNV
Group      ==> CICSOS2
Language   ==> Assembler      Cobol|Assembler|Le370|C|Pl i
                                   |Rpg
RELoad     ==> No              No|Yes
RESident   ==> No              No|Yes
RS1        ==> 00              0-24|Public
Status     ==> Enabled         Enabled|Disabled
```

For all program definition screens, see the notes on page 21.

Remote system TERMINAL definition (single session)

For a single-session link, a terminal definition can serve as the CICS on System/390 definition of the LU 6.2 connection to CICS for OS/2, CICS for Windows NT, CICS/400, or CICS on Open Systems. This example is a definition of an LU 6.2 connection to CICS for OS/2.

No remote attributes are necessary because the non-System/390 system appears to CICS on System/390 as a locally-attached terminal. (In other words, CICS on System/390 owns the connection.)

NETNAME must be the name by which the remote system is known to VTAM. (For CICS for OS/2, this equates to the **LU name** specified on a Communication Manager **Local LU** definition. The Local LU definition maps to the **LU alias** on the CICS for OS/2 terminal control table system entry (TCS) for the System/390. See the *CICS for OS/2 Intercommunication* manual for further details.)

Most of the values should be entered as shown. There are no constraints on the terminal name or operator defaults. For an example accompanying TYPETERM definition, see page 25.

```

DEFINE TERMINAL(PS02) GROUP(CICS0S2)
OVERTYPE TO MODIFY
CEDA DEFine
Terminal      ==> PS02
Group         ==> CICS0S2
AUTOINSTModel ==> No                No|Yes|Only
AUTOINSTName ==>
TERMINAL IDENTIFIERS
Typeterm      ==> APPC
Netname       ==> T1112821
CONSOLE       ==> No                No|0-99
REMOTESystem  ==>
REMOTESysnet  ==>
REMOTENAME    ==>
Modename      ==> SIGMA
ASSOCIATED PRINTERS
PRINTER       ==>
PRINTERCopy   ==> No                No|Yes
ALTPRINTER    ==>
ALTPRINTERCopy ==> No                No|Yes
PIPELINE PROPERTIES
POOL          ==>
TASKLIMIT     ==> No                No|Yes
OPERATOR DEFAULTS
OPERID        ==>
OPERPriority   ==> 000                0-255
OPERRs1       ==> 0                  0-24,...
OPERSecurity  ==> 1                  1-64,...
PRESET SECURITY
Userid        ==>
TERMINAL USAGES
TRANSACTION    ==>
TERMPriority   ==> 000                0-255
Inservice     ==> Yes                Yes|No
SESSION SECURITY
Securityname   ==>
ATTACHSEC      ==> Local            Local|Identify|Verify|
                                           Persistent|Mixidpe
BINDPassword   ==>                  PASSWORD NOT SPECIFIED
BINDSecurity   ==> No                No|Yes

```

The following notes apply to all terminal definition screens.

LU 6.2 definitions

Notes:

1. The OPERID, OPERPRIORITY, OPERRSL, and OPERSECURITY options are obsolete in CICS TS 390, CICS Transaction Server for VSE/ESA, and in CICS/ESA releases from Version 3.3 onwards, but are retained for compatibility with earlier releases.
2. The REMOTESYSNET option applies only to CICS/ESA 4.1, CICS Transaction Server for OS/390, and CICS Transaction Server for VSE/ESA.
3. The ATTACHSEC, BINDPASSWORD, and BINDSECURITY options apply only to APPC (LUTYPE6.2) links. BINDSECURITY is not applicable to CICS/VSE Version 2. BINDPASSWORD is not applicable to CICS Transaction Server for OS/390 or CICS Transaction Server for VSE/ESA.

Remote system TYPETERM definition

This definition accompanies the TERMINAL definition on page 23.

Most of the values below must be entered as shown.

```

DEFINE TYPETERM(APPC) GROUP(CICSOS2)
OVERTYPE TO MODIFY
CEDA DEFINE
  TYPeterm      : APPC
  Group         : CICSOS2
RESOURCE TYPE
  DEvice        ==> APPC
  TERmodel      ==>
  SESsiontype   ==>
  LDcllist      ==>
  SHippable     ==> No                No|Yes
MAPPING PROPERTIES
  PAGesize      ==> 000 , 000        0-999
  ALTPage       ==> 000 , 000        0-999
  ALTSuffix     ==>
  FMhparm       ==> No                No|Yes
  OBOperid      ==> No                No|Yes
PAGING PROPERTIES
  AUTOPage      ==> No                No|Yes
DEVICE PROPERTIES
  SOSi          ==> No                No|Yes
  BAcKtrans     ==> No                No|Yes
  CGosgid       ==> 00000 , 00000    0-65535
SESSION PROPERTIES
  AScii         ==> No                No|7|8
  SENDsize      ==> 00256            0-30720
  RECEivesize   ==> 00256            0-30720
  BRacket       ==> Yes              Yes|No
  LOGMode       ==>
DIAGNOSTIC DISPLAY
  ERRLastline   ==> No                No|Yes
  ERRIntensity  ==> No                No|Yes
  ERRColor      ==> No                No|Blue|Red|Green
  |Turquoise|Yellow|NEutral
  |No|Blink|Reverse|Underline
  ERRHilight    ==> No
OPERATIONAL PROPERTIES
  AUTOConnect   ==> Yes              No|Yes|A11
RECOVERY
  RECOVOption   ==> Sysdefault        Sysdefault|Clearconv|
  |Releasesess|Uncondre1|None
  RECOVNotify   ==> None              None|Message|Transaction

```

Note: In CICS/ESA 4.1, CICS TS 390, and CICS Transaction Server for VSE/ESA, it is possible to:

- Autoinstall single-session LU 6.2 links initiated by BIND requests
- Use VTAM persistent sessions on LU 6.2 links.

For details, see the relevant *Intercommunication Guide*.

LU 6.2 definitions

CONNECTION definition for parallel sessions

This sample CONNECTION definition for parallel sessions accompanies the SESSIONS definition that follows. NETNAME (IYA79270 in this example) must be the name by which the remote system is known to VTAM.

The connection name (ISCA in this example) is the REMOTESYSTEM value in remote definitions of resources (files, terminals, and transactions) owned by the non-System/390 CICS system linked by this connection.

```
DEFINE CONNECTION(ISCA) GROUP(ISCAA)
OVERTYPE TO MODIFY
CEDA DEFINE
  Connection ==> ISCA
  Group ==> ISCAA
CONNECTION IDENTIFIERS
  Netname ==> IYA79270
  INDSys ==>
CONNECTION PROPERTIES
  AAccessmethod ==> VTAM          Vtam|IRc|INdirect|Xm
  PRotocol ==> APPC              Appc|Lu61
  SInglesess ==> No              No|Yes
  DAtastream ==> User            User|3270|SCs|STrfield|Lms
  REcordformat ==> U             U|Vb
  QueueLimit ==> No             No|0-9999
  Maxqtime ==> No                No|0-9999
OPERATIONAL PROPERTIES
  AUtoconnect ==> Yes            Yes|No
  INService ==> Yes              Yes|No
SECURITY
  SEcurityname ==>
  ATtachsec ==> Local            Local|Identify|Verify|
                                   Persistent|Mixidpe
  BINDPassword ==>
  BINDSecurity ==> No            PASSWORD NOT SPECIFIED
                                   No|Yes
RECOVERY
  Psrecovery ==> Sysdefault      Sysdefault|None
```

Notes:

1. QUEUELIMIT and MAXQTIME apply only to CICS/ESA 4.1, CICS TS 390, and CICS Transaction Server for VSE/ESA. They enable you to control the queuing of requests for free sessions on the connection. See the relevant *Intercommunication Guide* for more details.
2. PSRECOVERY applies only to CICS/ESA 4.1, CICS TS 390, and CICS Transaction Server for VSE/ESA. With the RECOVPTION attribute of DEFINE SESSIONS, it enables you to use VTAM persistent sessions on LU 6.2 links. See the relevant *Intercommunication Guide* for more details.
3. BINDSECURITY is not applicable to CICS/VSE Version 2. BINDPASSWORD is not applicable to CICS Transaction Server for OS/390 or CICS Transaction Server for VSE/ESA.

SESSIONS definition for parallel sessions

This sample SESSIONS definition for parallel sessions accompanies the preceding CONNECTION definition.

```

DEFINE SESSIONS(ISCA) GROUP(ISCAA)
OVERTYPE TO MODIFY
CEDA DEFINE
SESSION IDENTIFIERS
Sessions      ==> ISCA
Group         ==> ISCAA
Connection    ==> ISCA
MOdename     ==> CICSISCO
SESSION PROPERTIES
Protocol      ==> APPC           Appc|Lu61
MAximum      ==> 5,2
SENDSIZE     ==> 4096           1-30720
RECEIVESIZE ==> 1024           1-30720
SESSPriority ==> 000            0-255
OPERATIONAL PROPERTIES
Autoconnect  ==> YES           Yes|No
INservice    ==> Yes          Yes|No
Buildchain   ==> Yes          Yes|No
IOarealen    ==> 0,0          0-32767
RELreq       ==> Yes          Yes|No
DIScreq      ==> Yes          Yes|No
NEPclass     ==> 000          0-255
RECOVERY
RECOVoption  ==> Sysdefault    Sysdefault|Clearconv|
                                           Releasesess|Uncondre1|No
RECOVNotify  ==> None          None|Message|Transaction

```

Note: In CICS/ESA 4.1, CICS TS 390, and CICS Transaction Server for VSE/ESA, you can autoinstall parallel-session LU 6.2 links that are initiated by BIND requests. For details, see the relevant *Intercommunication Guide*.

LU 6.2 definitions

Remote terminal **TERMINAL** definition

CICS systems cannot ship definitions of:

- Non-VTAM terminals
- Terminals that have been defined as SHIPPABLE(NO).

If an incoming transaction routing request (or an automatic transaction initiation (ATI) request) is associated with a non-shippable terminal, CICS on System/390 must have a remote terminal definition that meets the following requirements:

- REMOTESYSTEM must be the System/390 name of the connection to the non-System/390 (terminal-owning) system.

For CICS/VSE Version 2 and releases of CICS/ESA before Version 4.1, if there is no direct connection to the terminal-owning system, REMOTESYSTEM must specify the name of an **indirect** connection. For details of indirect connections, when they are required and how to define them, see the *Intercommunication Guide* for your CICS on System/390 product.

For CICS/ESA 4.1, CICS TS 390, and CICS Transaction Server for VSE/ESA, if there is no direct connection to the terminal-owning system, REMOTESYSTEM must specify the name of the first connection in the path to it. REMOTESYSNET must specify the network name by which the terminal-owning system is known to VTAM. See the *Intercommunication Guide*.

- A subset of the 3270 extended data stream is supported.
- The color, highlight, PS, and outline extended attributes are supported but only to the extent that they are generated by BMS. A field can have these attributes but an individual character within a field cannot have separate attributes.
- TCTUALENG must be the same as TCTUAL in the non-System/390 TCT definition for the terminal.
- REMOTENAME must be the same as the terminal id in the non-System/390 system's definition for the terminal.

An example definition follows.

```

DEFINE TERMINAL(R123) GROUP(CICS0S2)
OVERTYPE TO MODIFY
CEDA DEFine
TERminal    ==> R123
Group       ==> CICS0S2
AUTINSTModl ==> No           No|Yes|On1y
TERMINAL IDENTIFIERS
TYpeterm   ==> LU62TR
NEtname     ==>
CONSOLE     ==> No           No|0-99
REMOTESystem ==> ISCA
REMOTESYsnet ==>
REMOTEName  ==> V123
ASSOCIATED PRINTERS
PRINTERCopy ==> No           No|Yes
ALTPRINTCopy ==> No           No|Yes
PIPELINE PROPERTIES
TAsklimit   ==> No           No|1-32767
OPERATOR DEFAULTS
OPERPID      ==>
OPERPriority ==> 000         0-255
OPERRs1     ==> 0           0-24,...
OPERSecurity ==> 1           1-64,...
PRESET SECURITY
Userid      ==>
TERMINAL USAGES
TERmpriority ==> 000         0-255
Inservice   ==> Yes         Yes|No
SESSION SECURITY
ATtachsec   ==> Local       Local|Identify|Verify|
                                     Persistent|Mixidpe
BINDPassword ==>              PASSWORD NOT SPECIFIED
BINDSecurity ==> No         No|Yes

```

For all terminal definitions, see the notes on page 23.

LU 6.2 definitions

Remote terminal TYPETERM definition

Here is an example of a TYPETERM definition to accompany the preceding TERMINAL definition.

```
DEFINE TYPETERM(LU62TR) GROUP(CICS0S2)
OVERTYPE TO MODIFY
CEDA DEFINE
  TYPeterm    ==> LU62TR
  Group       ==> CICS0S2
RESOURCE TYPE
  DEvice      ==> 3270
  TERmodel    ==> 2
  SESsiontype ==>
  LDcllist    ==>
  SHippable   ==> No           No|Yes
MAPPING PROPERTIES
  PAGesize    ==> 024 , 080    0-999
  ALTPage     ==> 024 , 080    0-999
  ALTSuffix   ==>
  FMhparm     ==> No           No|Yes
  OBOperid    ==> No           No|Yes
PAGING PROPERTIES
  AUTOPage    ==> No           No|Yes
DEVICE PROPERTIES
  SOSi        ==> No           No|Yes
  BAcKtrans   ==> No           No|Yes
  CGcsgid     ==> 00000 , 00000 0-65535
SESSION PROPERTIES
  AScii       ==> No           No|7|8
  SENDsize    ==> 00000        0-30720
  RECEivesize ==> 01920        0-30720
  BRacket     ==> Yes          Yes|No
  LOGMode     ==>
DIAGNOSTIC DISPLAY
  ERRLastline ==> Yes          No|Yes
  ERRIntensify ==> Yes         No|Yes
  ERRColor    ==> No           NO|Blue|Red|Pink|Green|
                               Turquoise|Yellow|NEutral
  ERRHilight  ==> No           No|Blink|Reverse|Underline
OPERATIONAL PROPERTIES
  AUTOConnect ==> Yes          No|Yes|A11
  ATi         ==> Yes          No|Yes
RECOVERY
  RECOVOption ==> Sysdefault  Sysdefault|Clearconv|
                               Releasesess|Uncondre1|None
  RECOVNotify ==> None        None|Message|Transaction
```

Chapter 4. CICS internal security

Important

- This chapter applies only to CICS/VSE Version 2 users of CICS internal security.
- It includes sample resource definition fragments, using the CEDA transaction and the DFHFCT and DFHSNT macros. For full information on the CEDA transaction and the DFHFCT and DFHSNT macros, see the *CICS/VSE Version 2 Release 3 Resource Definition Guide*.

You can protect the System/390's resources against unauthorized access by remote users. Depending on your CICS on System/390 product, you may be able to use either CICS internal security or an external security manager (ESM), such as the Resource Access Control Facility (RACF). CICS TS 390, CICS Transaction Server for VSE/ESA, and CICS/ESA releases after 3.1.1 do not support CICS internal security. CICS/VSE Version 2 supports either method.⁷

This chapter describes only CICS internal security. CICS/ESA and CICS TS 390 users of RACF should refer to the *CICS-RACF Security Guide* for their System/390 product. CICS/ESA and CICS TS 390 users of other external security managers should refer to the documentation for their ESM. CICS Transaction Server for VSE/ESA users should refer to the *CICS Transaction Server for VSE/ESA Security Guide*. CICS/VSE Version 2 users of external security should refer to the documentation for their ESM.

Introduction to CICS/VSE internal security

CICS/VSE internal security, apart from resource security, is defined at two levels:

User security

You define a security profile for each workstation user.

Link security

You define a security profile for the link. This gives the link access to all the resources that the users can access collectively. No user has access to a resource that the link itself cannot access.

Security profile

Security profiles consist of one or more numeric keys, chosen from the digits 1 through 24. When a protected resource is defined, it is associated with one of these values. A user who has a matching key is allowed to access that resource, provided that the link also has a matching key (user security is a subset of link security).

Link profile

If you do not need to specify security for individual users, you can let all user security profiles default to the link profile. For this, you specify ATTACHSEC(Local) on the CEDA DEFINE CONNECTION⁸ command. You define the link security profile by specifying OPERRSL on the same command. Let this option default if you want the link to access only unprotected resources.

7. CICS/VSE Version 2 and CICS Transaction Server for VSE/ESA support external security but not RACF.

8. If the link is single-session LU 6.2, specify ATTACHSEC and SECURITYNAME in the CEDA DEFINE TERMINAL command.

introduction to internal security

SNT entry for link

An alternative way to specify link security is to define an SNT entry for the link. Specify RSLKEY to define the link security profile. The user ID you give to the link has to be matched to SECURITYNAME on the CEDA DEFINE CONNECTION⁸ command.

Required specifications in remote systems

To enable protected access to CICS/VSE resources, specifications are needed in each remote system.

CICS for OS/2 specifications

To enable security checking of CICS for OS/2 users by CICS/VSE, **Attach security** must be specified as V in the CICS for OS/2 TCS definition of the CICS/VSE system. All user IDs must be defined in the CICS for OS/2 signon table (SNT).

OS/2 communications manager: **Conversation security** must be specified as YES in the OS/2 Communications Manager profile for the associated partner LU alias.

CICS for AIX specifications

In the communications definition (CD) stanza, the entry for the CICS/VSE system should specify **RemoteSystemSecurity=IDENTIFY**, which is consistent with either IDENTIFY or VERIFY in the SNA Services connection profile. All user IDs, whether or not they use intercommunication, must be in the user definition (UD) stanza.

AIX SNA Services: In the connection profile for the CICS/VSE system, the conversation security access list must contain the user IDs and passwords of all users that are to access the CICS/VSE system. The connection profile should specify **SecurityLevel=IDENTIFY** or **VERIFY**, depending on the security required.

CICS/400 specifications

An AS/400 user profile, containing a user ID and password, is required for each CICS/400 user who accesses protected CICS/VSE resources. In the AS/400 configuration list, the entry for the CICS/VSE system should specify **Secure Loc(*YES)**, which is the equivalent of ATTACHSEC=Verify in the CICS/VSE CONNECTION definition.

CICS/VSE specifications

For CICS/VSE resource security, entries are needed in the SNT for all remote users. Each entry must match a corresponding entry in a remote system's SNT or equivalent.⁹The level of security on a link depends on the ATTACHSEC option of the CEDA DEFINE CONNECTION¹⁰ command.

If you are using an external security manager, you probably need only the *default* entry in the CICS SNT. This covers both link and users.

Because the mirror transaction accesses all resources for the users, CICS/VSE does not apply resource security checking unless you specify RSLC(YES) or RSLC(EXTERNAL) on the CEDA DEFINE TRANSACTION for the mirror transaction.

For further guidance, see the *CICS/VSE Version 2 Release 3 Intercommunication Guide*. Note that bind-time security is not supported.

Implementation

Implementation of security for CICS for OS/2, CICS for Windows NT, CICS on Open Systems, or CICS/400 access to CICS/VSE resources is similar to that for CICS/VSE—CICS/VSE intercommunication.

Sign-on security

If ATTACHSEC=IDENTIFY is specified in the CICS/VSE and CICS non-System/390 connection definitions, the remote user ID must match an entry in the CICS/VSE SNT. For ATTACHSEC=VERIFY, the user ID and password transmitted with the request must match the user ID and password in a CICS/VSE SNT entry. For ATTACHSEC=LOCAL, there is no user security.

Attach-time and resource access security

For ATTACHSEC=LOCAL, the resources accessed must have security keys that are a subset of the range of the OPERRSL keys specified for the connection.

For ATTACHSEC=VERIFY|IDENTIFY, in addition to the requirements for ATTACHSEC=LOCAL, the user's SNT operator class must match the RSL key for the resource. Additional checks may be needed, depending on the definitions of mirror and routed transactions.

9. For example, for CICS for AIX this is the UD stanza; for CICS/400, the AS/400 user profiles.

10. If the link is single-session LU 6.2, specify ATTACHSEC and SECURITYNAME in the CEDA DEFINE TERMINAL command.

Specifying LU 6.2 security requirements

Specifying your security requirements involves entries in several CEDA and macro resource definitions:

- Mirror transaction
- Data conversion modules
- Connection
- Session
- Routed transaction
- File control table
- Sign-on table.

Note: If a definition is included in the supplied group DFHISC, you should copy it to another group before making any changes.

Mirror transaction

If you want security protection of CICS/VSE resources such as files, transient data destinations, and temporary storage queues, you must specify RSLC(YES) for the mirror transaction. You can protect the mirror transaction itself by specifying a TRANSEC value other than 1 or an RSL value other than PUBLIC, or both. TRANSEC (2–64) causes a check of the operator class and link security keys when the mirror is invoked from a CICS non-System/390 terminal. RSL(00) prevents access to the mirror transaction by any other transaction that is itself specified with RSLC(YES).

The simplest way to protect your resources is suggested below. With these definitions, the mirror transaction itself is unprotected. You control CICS/VSE resource security by the resource RSL key and the connection, terminal, and user keys.

```

ALTER TRANSACTION (CPMI) GROUP(CICS0S2)
OVERTYPE TO MODIFY
CEDA ALTER
.....
SECURITY
Extsec          No          No|Yes
TRANsec         01          1-64
RSL             00          0-24|Public
RSLC            YES        No|Yes|External
.....
    
```

Mirror program and data conversion modules

If you specify RSLC=YES for the mirror transaction, CICS checks the RSL keys for all resources it accesses. These resources include the mirror program, and the data conversion modules, DFHCNV, DFHCCNV, and DFHUCNV. Each resource can be associated with only one key, and the RSL key for each resource is likely to be different. It is therefore easiest to specify the key for all these modules as PUBLIC, which allows unprotected access. This creates no security exposure if you protect your data resources.

It is recommended that you specify the RSL key for the mirror program as shown, and repeat for DFHCNV, DFHCCNV, and DFHUCNV.

```
ALTER PROGRAM(DFHMIRS) GR(CICS0S2)
OVERTYPE TO MODIFY
CEDA ALTer
.....
RSL          Public          0-24|Public
.....
```

Connection

The ATTACHSEC value in the connection definition determines the level of user security. If you want CICS/VSE to verify each user and password, alter the connection definition as shown below. **For CICS for OS/2—CICS/VSE or CICS for Windows NT—CICS/VSE links, do not specify BINDPASSWORD.**

```
ALTER CONNECTION (APPC) GR(CICS0S2)
OVERTYPE TO MODIFY
CEDA ALTer
.....
ATTachsec    Verify          Local|Identify|Verify
.....
```

Session

The OPERRSL values in the session definition must include the RSL key of any protected resource accessed. The OPERSECURITY values must include the TRANSEC key of any protected transaction accessed. Only resources with an RSL key of **4** and transactions with a TRANSEC key of **10** can be accessed by links set up with the session definition below.

```
ALTER SESSION (ONE) GR(CICS0S2)
OVERTYPE TO MODIFY
CEDA ALTer
.....
OPERRs1      4              0-24
OPERSECurity 10             1-64
.....
```

Transaction

Security of transaction routing from a CICS non-System/390 system is affected by the definition of the routed transaction. The example shows the security fields in the definition of a transaction. The meanings of the entries shown are:

EXTSEC(NO)

specifies the use of CICS resource security level (RSL) checking rather than an external security manager.

TRANSEC(10)

specifies that routing of this transaction is permitted only if the link and terminal OPERSECURITY keys include the value **10**. If the terminal has no OPERSECURITY key, the user's SNT operator keys must include the value **10**. For example, TRANSEC(1) means that the transaction can be attached by any user or terminal.

RSL(0)

prevents invocation of this transaction by other transactions defined with RSLC(YES). This protects the transaction without affecting transaction routing.

specifying LU 6.2 security

RSLC(YES)

specifies that security checking is required for resources accessed by this transaction.

```
ALTER TRANSACTION (RTED) GR(CICS0S2)
OVERTYPE TO MODIFY
CEDA ALTer
.....
SECURITY
Extsec          No          No|Yes
TRANsec         10          1-64
RSL             00          0-24|Public
RSLC            YES         No|Yes|External
.....
```

File control table (FCT)

If FILEA is to have a resource security value of 4, generate your file control table (FCT) to include the following definition:

To protect transient data destinations and temporary storage queues, use a similar

```
DFHFCT TYPE=FILE,          X
        DATASET=FILEA,     X
        RSL=4,              X
```

Figure 5. File control table

entry in any DFHDCT TYPE=EXTRA, DFHDCT TYPE=INTRA, and DFHTST TYPE=SECURITY macros.

Signon table (SNT)

Specify each CICS non-System/390 user in the SNT on CICS/VSE, as shown in Figure 6. For a CICS non-System/390 user to use a CICS/VSE link with ATTACHSEC=VERIFY security, an entry in the non-System/390 signon table (or equivalent) must have a password and user ID that exactly match an entry in the CICS/VSE signon table.

For ATTACHSEC=IDENTIFY, only the user IDs must match.

For ATTACHSEC=LOCAL, all users can use the link.

Assuming the connection is defined with ATTACHSEC=VERIFY, the example CICS/VSE SNT entry below makes the link available to a CICS for OS/2 user with a CICS for OS/2 SNT entry that specifies user ID **USR1** and password **PAS1**.

```
DFHSNT TYPE=ENTRY,          X
        OPIDENT=USR1,       X
        PASSWORD=PAS1,     X
        USERID=USR1
```

Figure 6. Example signon table

specifying LU 6.2 security

The user's authority to access resources is determined by the RSLKEY, SCTKEY, and OPCLASS options in this entry.

Chapter 5. VTAM/NCP definitions

CICS on System/390 requires VTAM/NCP definitions for all links to CICS for OS/2, CICS for Windows NT, CICS/400, or CICS on Open Systems, except DFT 3270 links. This chapter gives examples of a VTAM application definition, and of NCP and MODETABLE definitions for both single sessions and parallel sessions. To understand the details of the examples, see the *VTAM Resource Definition Reference* manual, SC33-6412.

VTAM application definition

```
          VBUILD  TYPE=APPL
DBDCCICS  APPL    ...,PARSESS=YES,MODETAB=MODELU62,SONSCIP=YES,...
```

Figure 7. Host VTAM application definition

Single sessions

```
G115D128  GROUP  MAXDATA=265,
:
          MODETAB=MTSIGMA
:
L11SD128  LINE  ADDRESS=(128,HALF)
          SERVICE ORDER=(P111283,P111284)
P111283  PU    ADDR=C3,PUTYPE=2,ISTATUS=INACTIVE
T1112830  LU   LOCADDR=2
T1112831  LU   LOCADDR=3
T1112832  LU   LOCADDR=4
T1112833  LU   LOCADDR=5
T1112834  LU   LOCADDR=6
T1112835  LU   LOCADDR=7
T1112836  LU   LOCADDR=8
T1112837  LU   LOCADDR=9
P111284  PU    ADDR=C4,PUTYPE=2,ISTATUS=INACTIVE
T1112840  LU   LOCADDR=2
T1112841  LU   LOCADDR=3
T1112842  LU   LOCADDR=4,DLOGMOD=SIGMA
T1112843  LU   LOCADDR=5
T1112844  LU   LOCADDR=6
T1112845  LU   LOCADDR=7
T1112836  LU   LOCADDR=8
T1112837  LU   LOCADDR=9
```

Figure 8. Host VTAM NCP definitions for single sessions

VTAM/NCP definitions for parallel sessions

```
MTSIGMA  MODEENT  LOGMODE=MTSIGMA

SIGMA    MODEENT  LOGMODE=SIGMA,
          TYPE=0,      NEGOTIABLE BIND
          FMPROF=X'13', FM PROFILE
          TSPROF=X'07', TS PROFILE
          PRIPROT=X'B0', PRIM PROTOCOL
          SECPROT=X'B0', SEC PROTOCOL
          COMPROT=X'50B1', COMMON PROTOCOL
          PSNDPAC=X'00', PRIM SEND PACING
          SRCVPAC=X'00', SEC RECEIVE PACING
          SSNSPAC=X'00', SEC SEND PACING
          RUSIZES=X'8585', RU IN=256. RU OUT=256.
          PSERVICE=X'060200000000000000002C00'
```

Figure 9. Host VTAM MODETABLE definitions for single sessions

Parallel sessions

The example in Figure 10 on page 41 is for a System/390–PS/2 SDLC LU 6.2 link with six PUs. The LU definitions with LOCADDR=0 are for PU2.1 independent parallel sessions. All the other definitions are for PU2.0 dependent single-session LUs.

VTAM/NCP definitions for parallel sessions

```
* Line 027 set up for parallel sessions
* XID=YES & RESSCB are required for parallel sessions
*
IYA7L027 LINE ADDRESS=(027,FULL),OWNER=HP
SERVICE ORDER=(IYA7C027,IYA7C327,IYA7C627)
SERVICE ORDER=(IYA7C927,IYA7CC27,IYA7CF27)
IYA7C027 PU ADDR=C1,PUTYPE=2,ISTATUS=ACTIVE,XID=YES
IYA70270 LU LOCADDR=0,MODETAB=MODEL62,DLOGMOD=LU62PS,RESSCB=30
IYA70271 LU LOCADDR=1
IYA70272 LU LOCADDR=2
IYA70273 LU LOCADDR=3
IYA70274 LU LOCADDR=4
IYA70275 LU LOCADDR=5
IYA70276 LU LOCADDR=6
IYA70277 LU LOCADDR=7
IYA7C327 PU ADDR=C2,PUTYPE=2,ISTATUS=ACTIVE,XID=YES
IYA73270 LU LOCADDR=0,MODETAB=MODEL62,DLOGMOD=LU62PS,RESSCB=30
IYA73271 LU LOCADDR=1
IYA73272 LU LOCADDR=2
IYA73273 LU LOCADDR=3
IYA73274 LU LOCADDR=4
IYA73275 LU LOCADDR=5
IYA73276 LU LOCADDR=6
IYA73277 LU LOCADDR=7
IYA7C627 PU ADDR=C3,PUTYPE=2,ISTATUS=ACTIVE,XID=YES
IYA76270 LU LOCADDR=0,MODETAB=MODEL62,DLOGMOD=LU62PS,RESSCB=30
IYA76271 LU LOCADDR=1
IYA76272 LU LOCADDR=2
IYA76273 LU LOCADDR=3
IYA76274 LU LOCADDR=4
IYA76275 LU LOCADDR=5
IYA76276 LU LOCADDR=6
IYA76277 LU LOCADDR=7
IYA7C927 PU ADDR=C4,PUTYPE=2,ISTATUS=ACTIVE,XID=YES
IYA79270 LU LOCADDR=0,MODETAB=MODEL62,DLOGMOD=LU62PS,RESSCB=30
IYA79271 LU LOCADDR=1
IYA79272 LU LOCADDR=2
IYA79273 LU LOCADDR=3
IYA79274 LU LOCADDR=4
IYA79275 LU LOCADDR=5
IYA79276 LU LOCADDR=6
IYA79277 LU LOCADDR=7
IYA7CC27 PU ADDR=C5,PUTYPE=2,ISTATUS=ACTIVE,XID=YES
IYA7C270 LU LOCADDR=0,MODETAB=MODEL62,DLOGMOD=LU62PS,RESSCB=30
IYA7C271 LU LOCADDR=1
IYA7C272 LU LOCADDR=2
IYA7C273 LU LOCADDR=3
IYA7C274 LU LOCADDR=4
IYA7C275 LU LOCADDR=5
IYA7C276 LU LOCADDR=6
IYA7C277 LU LOCADDR=7
IYA7CF27 PU ADDR=C6,PUTYPE=2,ISTATUS=ACTIVE,XID=YES
IYA7F270 LU LOCADDR=0,MODETAB=MODEL62,DLOGMOD=LU62PS,RESSCB=30
IYA7F271 LU LOCADDR=1
IYA7F272 LU LOCADDR=2
IYA7F273 LU LOCADDR=3
IYA7F274 LU LOCADDR=4
IYA7F275 LU LOCADDR=5
IYA7F276 LU LOCADDR=6
IYA7F277 LU LOCADDR=7
```

Figure 10. Host VTAM NCP definitions for parallel sessions

VTAM/NCP definitions for parallel sessions

```

MODELU62 MODETAB
*
* * * * *
* Log mode table for use with lu62 -
*
* PARALLEL SESSIONS=YES
*
*
* * * * *
LU62PS  MODEENT LOGMODE=LU62PS,
        TYPE=0,          ONLY TYPE RECOGNIZED
        FMPROF=X'13',    SNA
        TSPROF=X'07',    SNA
        PRIPROT=X'B0',   PRIMARY PROTOCOL
        SECPROT=X'B0',   SECONDARY PROTOCOL
        COMPROT=X'78A5', COMMON PROTOCOL
        SSNDPAC=X'00',
        SRCVPAC=X'00',
        RUSIZES=X'8989',  RUSIZES IN-4096 OUT-4096
        PSNDPAC=X'00',
        PSERVIC=X'0602000000000000122F00'
* * * * *
* LU62 REQUIRES SNASVCMG TO RUN PROPERLY for //11. session support
        MODEENT LOGMODE=SNASVCMG,COS=CICS2,
        TYPE=0,          ONLY TYPE RECOGNIZED
        FMPROF=X'13',    SNA
        TSPROF=X'07',    SNA
        PRIPROT=X'B0',   PRIMARY PROTOCOL
        SECPROT=X'B0',   SECONDARY PROTOCOL
        COMPROT=X'78A5', COMMON PROTOCOL
        SSNDPAC=X'00',
        SRCVPAC=X'00',
        RUSIZES=X'8989',  RUSIZES IN-4096 OUT-4096
        PSNDPAC=X'00',
        PSERVIC=X'0602000000000000122F00'
        MODEEND
        END

```

Figure 11. Host VTAM MODETABLE definitions for parallel sessions

Figure 11 shows the VTAM MODETABLE definitions required for a parallel-session LU 6.2 link.

Chapter 6. Data conversion

Whenever data is passed from one CICS system to another, some or all of the data may have to be converted from ASCII to EBCDIC format, or vice versa.

Where data conversion takes place

CICS intercommunication takes place on SNA links, which assume that all system data is in EBCDIC format. To ensure that transmitted data is in EBCDIC format, CICS ASCII-based systems convert all fields except application data areas, which are converted by the receiving system. Table 2 shows where data conversion is done for function shipping and DPL between CICS systems. The table includes *all* the conversion done automatically by CICS systems.

Table 2. Data conversion for function shipping and DPL

Request type	Data	Conversion type	Where converted
TS	Queue name	Character	ASCII system
	IOAREA	As specified in DFHCNV table	Receiving system
TD	Queue name	Character	ASCII system
	IOAREA	As specified in DFHCNV table	Receiving system
FC	File name	Character	ASCII system
	IOAREA	As specified in DFHCNV table	Receiving system
	Key	As specified in DFHCNV table	Receiving system
IC	Transaction ID	Character	ASCII system
	FROM area	As specified in DFHCNV table	Receiving system
	RTERMID, RTRANSID, REQID	Character	ASCII system
PC	Program name	Character	ASCII system
	COMMAREA	As specified in DFHCNV table	Receiving system

Function shipping and DPL

For function shipping and DPL *from* an ASCII system to CICS on System/390, the ASCII system converts the resource names, and CICS on System/390 converts the user data (see Table 2).

For function shipping *to* an ASCII system from CICS on System/390, the ASCII system does all the necessary conversion.

Distributed transaction processing

DTP uses application-specific data areas and cannot have a general procedure for data conversion. It is the application's responsibility to perform data conversion. Application design determines whether conversion is at the System/390 or the workstation. If the workstation is running the OS/2 EE operating system, Communications Manager provides conversion facilities that may be useful.

To facilitate conversion at the System/390, an application program can link directly to the user-replaceable conversion program DFHUCNV. If DFHUCNV is also to be called by the standard conversion program, DFHCCNV, the application must pass a parameter list as mapped by the DFHUNVDS DSECT, listed in Figure 23 on page 77, and set up all the fields addressed by the parameter list. For product-sensitive programming interface information, refer to "User-replaceable conversion program" on page 76.

Transaction routing

CICS on System/390 systems do no data conversion for transaction routing. Screen data always flows as 3270 data streams. COMMAREAs and TCTUAs (which are relevant to pseudoconversational transactions) are converted by the ASCII system.

Avoiding data conversion

Application design can reduce the amount of data conversion.

For example, if a System/390 CICS system acts as a file manager for CICS for OS/2 systems, the data in the file can be coded in ASCII, eliminating the need for data conversion.

If data is held at the workstation purely for the purpose of communicating with a System/390 CICS system, it can be coded in EBCDIC.

Types of conversion

The possible types of conversion are:

Standard conversion

This applies to:

- Single-byte character sets (SBCS)
- Graphic or double-byte character sets (DBCS)
- Mixed character sets (containing SBCS and DBCS data)
- Multi-byte character sets (MBCS)
- By default, to binary data in INTEL[®] format.

No conversion

This applies to:

- Character data encoded as UCS-2 or UTF-8
- By default, to binary data in System/390 format
- Packed decimal data.

User-defined nonstandard conversion

You can write your own version of DFHUCNV to apply any conversion. If specified, a user-defined conversion is applied instead of the standard conversion. You can apply user-defined conversion to selected fields, and leave others to be converted by the CICS standard conversion program.

types of data conversion

If the nonstandard conversion applies only to character data, you may not need to write your own version of DFHUNCV. Instead, you could create your own conversion tables for use with the standard conversion program, DFHCCNV. See “User-defined conversion tables” on page 68.

Attention

DFHUCNV must not convert any data that the standard conversion program attempts to convert. Double conversion gives unpredictable results. To ensure that double conversion does not occur, your DFHUCNV program must convert only fields defined as DATATYP=USERDATA (see page 67).

Character data

For standard conversion, the SRVERCP keyword on the DFHCNV TYPE=ENTRY macro determines the “server” code page; that is, the character set identifiers and code page identifiers for EBCDIC data associated with the specified resource.

The CLINTCP keyword on the DFHCNV TYPE=ENTRY macro determines the default “client” code page; that is the default character set identifiers and code page identifiers for ASCII character data associated with the specified resource.

The default client code page can be overridden. This allows several workstations, each using a different ASCII-coded graphic character representation, to share data with the System/390.

If the resource can be accessed from both CICS for OS/2 (or CICS for Windows NT) and CICS on Open Systems, the default client code page must be set to the code page used by CICS for OS/2. For example, specifying CLINTCP=932 allows CICS for OS/2 using code page 932 and CICS on Open Systems using code page 954 to access the same resource.

CICS-supported conversions

This section provides a complete list of the Coded Character Set Identifiers (CCSIDs) supported by CICS on System/390. Additional CCSIDs will be supported as necessary. However, there is no guarantee that such CCSIDs will be supported by all of the products/releases to which this edition applies.

For unsupported CCSIDs, you can create your own conversion tables, for use with the standard conversion program, DFHCCNV. See “User-defined conversion tables” on page 68.

For nonstandard conversions, you must supply your own conversion program, DFHUCNV—see “User/CICS conversion” on page 56.

character data

CICS on System/390 is able to convert character data between ASCII and EBCDIC provided that the client and server CCSIDs belong to the same group; the groups being:

Arabic

Baltic Rim

Latvia, Lithuania

Cyrillic

Eastern Europe; Bulgaria, Russia, Yugoslavia

Estonian

Estonia

Greek

Greece

Hebrew

Israel

Latin-1

Western Europe

Latin-1 including euro support

Western Europe

Latin-2

Eastern Europe; Albania, Czech Republic, Hungary, Poland, Romania, Slovakia, Yugoslavia, Former Yugoslavia

Latin-5

Turkey

Latin-9

Japanese

Japan

Korean

Korea

Simplified Chinese

Peoples' Republic of China

Traditional Chinese

Republic of China

The following tables list the CCSIDs supported for each group. For each CCSID, they show:

- The value to be specified for the CLINTCP or SRVERCP keyword.
- The codepage identifier or identifiers (CPGIDs).
- The minimum level of CICS that supports the CCSID. The CICS levels defined are "Base" and "T01".

Base

- CICS Transaction Server for OS/390
- CICS/ESA Version 4 Release 1
- CICS/ESA Version 3
- CICS/MVS Version 2
- CICS/VSE Version 2 Release 2

T01

- CICS Transaction Server for OS/390 Release 3
- CICS Transaction Server for OS/390 Release 1 or Release 2 plus APAR PQ19018
- CICS/ESA Version 4 Release 1 plus APAR PQ18896
- CICS/VSE Version 2 Release 3 plus APAR PQ19019

Arabic

Table 3. Arabic, ASCII

CLINTCP	in	CCSID	CPGID	Comments
864	Base	00864	00864	PC data: Arabic
1089 8859-6	Base	01089	01089	ISO 8859-6: Arabic
1256	T01	01256	01256	MS Windows: Arabic

Table 4. Arabic, EBCDIC

SRVERCP	in	CCSID	CPGID	Comments
420	Base	00420	00420	Host: Arabic

Note: Data conversion does not change the direction of Arabic data.

Baltic Rim

Table 5. Baltic Rim, ASCII

CLINTCP	in	CCSID	CPGID	Comments
921	T01	00921	00921	PC data: Latvia, Lithuania
1257	T01	01257	01257	MS Windows: Baltic Rim

Table 6. Baltic Rim, EBCDIC

SRVERCP	in	CCSID	CPGID	Comments
1112	T01	01112	01112	Host: Latvia, Lithuania

Cyrillic

Table 7. Cyrillic, ASCII

CLINTCP	in	CCSID	CPGID	Comments
855	Base	04951	00855	PC data: Cyrillic
866	Base	00866	00866	PC data: Cyrillic
915 8859-5	Base	00915	00915	ISO 8859-5: Cyrillic
1251	T01	01251	01251	MS Windows: Cyrillic

Table 8. Cyrillic, EBCDIC

SRVERCP	in	CCSID	CPGID	Comments
1025	Base	01025	01025	Host: Cyrillic multilingual

character data

Estonia

Table 9. Estonian, ASCII

CLINTCP	in	CCSID	CPGID	Comments
922	T01	00922	00922	PC data: Estonia
1257	T01	01257	01257	MS Windows: Baltic Rim

Table 10. Baltic Rim, EBCDIC

SRVERCP	in	CCSID	CPGID	Comments
1122	T01	01122	01122	Host: Estonia

Greek

Table 11. Greek, ASCII

CLINTCP	in	CCSID	CPGID	Comments
869	Base	00869	00869	PC data: Greece
813 8859-7	Base	00813	00813	ISO 8859-7: Greek
1253	T01	01253	01253	MS Windows: Greek

Table 12. Greek, EBCDIC

SRVERCP	in	CCSID	CPGID	Comments
875	Base	00875	00875	Host: Greek

Hebrew

Table 13. Hebrew, ASCII

CLINTCP	in	CCSID	CPGID	Comments
856	Base	00856	00856	PC data: Hebrew
916 8859-8	Base	00916	00916	ISO 8859-8: Hebrew
1255	T01	01255	01255	MS Windows: Hebrew

Table 14. Hebrew, EBCDIC

SRVERCP	in	CCSID	CPGID	Comments
424	Base	00424	00424	Host: Hebrew

Note: Data conversion does not change the direction of Hebrew data.

Latin-1

Table 15. Latin-1, ASCII

CLINTCP	in	CCSID	CPGID	Comments
437	Base	00437	00437	PC data: PC Base; USA, many other countries
819 8859-1	Base	00819	00819	ISO 8859-1: Latin-1 countries
850	Base	00850	00850	PC data: Latin-1 countries
1252	T01	01252	01252	MS Windows: Latin-1 countries

Table 16. Latin-1, EBCDIC

SRVERCP	in	CCSID	CPGID	Comments
037	Base	00037	00037	Host: USA, Canada (ESA), Netherlands, Portugal, Brazil, Australia, New Zealand
273	Base	00273	00273	Host: Austria, Germany
277	Base	00277	00277	Host: Denmark, Norway
278	Base	00278	00278	Host: Finland, Sweden
280	Base	00280	00280	Host: Italy
284	Base	00284	00284	Host: Spain, Latin America (Spanish)
285	Base	00285	00285	Host: United Kingdom
297	Base	00297	00297	Host: France
500	Base	00500	00500	Host: Belgium, Canada (AS/400), Switzerland, International Latin-1
871	Base	00871	00871	Host: Iceland
1047	T01	01047	01047	Host: Latin-1 Open Systems

Note: CICS/390 does not support conversion between ASCII CCSID 437 and EBCDIC CCSID 1047.

character data

Latin-1 including euro support

Table 17. Latin-1 including euro support, ASCII

CLINTCP	in	CCSID	CPGID	Comments
858	T01	00858	00858	PC data: Latin-1 countries
5348	T01	05348	01252	MS Windows: Latin-1 countries, version 2

Table 18. Latin-1 including euro support, EBCDIC

SRVERCP	in	CCSID	CPGID	Comments
1140	T01	01140	01140	Host: USA, Canada (ESA), Netherlands, Portugal, Brazil, Australia, New Zealand
1141	T01	01141	01141	Host: Austria, Germany
1142	T01	01142	01142	Host: Denmark, Norway
1143	T01	01143	01143	Host: Finland, Sweden
1144	T01	01144	01144	Host: Italy
1145	T01	01145	01145	Host: Spain, Latin America (Spanish)
1146	T01	01146	01146	Host: United Kingdom
1147	T01	01147	01147	Host: France
1148	T01	01148	01148	Host: Belgium, Canada (AS/400), Switzerland, International Latin-1
1149	T01	01149	01149	Host: Iceland

Latin-2

Table 19. Latin-2, ASCII

CLINTCP	in	CCSID	CPGID	Comments
852	Base	00852	00852	PC data: Latin-2 multilingual
912 8859-2	Base	00912	00912	ISO 8859-2: Latin-2 multilingual
1250	T01	01250	01250	MS Windows: Latin-2

Table 20. Latin-2, EBCDIC

SRVERCP	in	CCSID	CPGID	Comments
870	Base	00870	00870	Host: Latin-2 multilingual

Latin-5

Table 21. Latin-5, ASCII

CLINTCP	in	CCSID	CPGID	Comments
857	Base	04953	00857	PC data: Latin-5 (Turkey)
920 8859-9	Base	00920	00920	ISO 8859-9: Latin-5 (ECMA-128, Turkey TS-5881)
1254	T01	01254	01254	MS Windows: Turkey

Table 22. Latin-5, EBCDIC

SRVERCP	in	CCSID	CPGID	Comments
1026	Base	01026	01026	Host: Latin-5 (Turkey)

Latin-9

Table 23. Latin-9, ASCII

CLINTCP	in	CCSID	CPGID	Comments
923	T01	00923	00923	ISO 8859-15: Latin-9

Table 24. Latin-9, EBCDIC

SRVERCP	in	CCSID	CPGID	Comments
924	T01	00924	00924	Host: Latin-9

character data

Japanese

Table 25. Japanese, ASCII

CLINTCP	in	CCSID	CPGID	Comments
932	Base	00932	00897	PC data: SBCS
			00301	PC data: DBCS including 1880 user-defined characters
942	Base	00942	01041	PC data: Extended SBCS
			00301	PC data: DBCS including 1880 user-defined characters
943	T01	00943	00897	PC data: SBCS
			00941	PC data: DBCS for Open environment including 1880 IBM user-defined characters
954 EUCJP	Base	00954	00895	G0: JIS X201 Roman
			00952	G1: JIS X208-1990
			00896	G2: JIS X201 Katakana
			00953	G3: JIS X212

Table 26. Japanese, EBCDIC

SRVERCP		CCSID	CPGID	Comments
930	Base	00930	00290	Katakana Host: extended SBCS
			00300	Kanji Host: DBCS including 4370 user-defined characters
930	Base	05026	00290	Katakana Host: extended SBCS
			00300	Kanji Host: DBCS including 1880 user-defined characters
931	Base	00931	00037	Latin Host: SBCS
			00300	Kanji Host: DBCS including 4370 user-defined characters
939	Base	00939	01027	Latin Host: extended SBCS
			00300	Kanji Host: DBCS including 4370 user-defined characters
939	Base	05035	01027	Latin Host: extended SBCS
			00300	Kanji Host: DBCS including 1880 user-defined characters

Korean

Table 27. Korean, ASCII

CLINTCP	in	CCSID	CPGID	Comments
934	Base	00934	00891	PC data: SBCS
			00926	PC data: DBCS including 1880 user-defined characters
944	Base	00944	01040	PC data: Extended SBCS
			00926	PC data: DBCS including 1880 user-defined characters
949	Base	00949	01088	IBM KS Code - PC data: SBCS
			00951	IBM KS code - PC data: DBCS including 1880 user-defined characters
970 EUCKR	Base	00970	00367	G0: ASCII
			00971	G1: KSC X5601-1989 including 1880 user-defined characters
1363	T01	01363	01126	PC data: MS Windows Korean SBCS
			01362	PC data: MS Windows Koran DBCS including 11172 full Hanguk

Table 28. Korean, EBCDIC

SRVERCP	in	CCSID	CPGID	Comments
933	Base	00933	00833	Host: Extended SBCS
			00834	Host: DBCS including 1880 user-defined characters

Simplified Chinese

Table 29. Simplified Chinese, ASCII

CLINTCP	in	CCSID	CPGID	Comments
946	Base	00946	01042	PC data: Extended SBCS
			00928	PC data: DBCS including 1880 user-defined characters
1381	Base	01381	01115	PC data: Extended SBCS (IBM GB)
			01380	PC data: DBCS (IBM GB) including 31 IBM-selected, 1880 user-defined characters
1383 EUCCN	T01	01383	00367	G0: ASCII
			01382	G1: GB 2312-80 set
1386	T01	01386	01114	PC data: S-Chinese GBK and T-Chinese IBM BIG-5
			01385	PC data: S-Chinese GBK

Table 30. Simplified Chinese, EBCDIC

SRVERCP	in	CCSID	CPGID	Comments
935	Base	00935	00836	Host: Extended SBCS
			00837	Host: DBCS including 1880 user-defined characters

character data

Traditional Chinese

Table 31. Traditional Chinese, ASCII

CLINTCP	in	CCSID	CPGID	Comments
938	Base	00938	00904	PC data: SBCS
			00927	PC data: DBCS including 6204 user-defined characters
948	Base	00948	01043	PC data: Extended SBCS
			00927	PC data: DBCS including 6204 user-defined characters
950 BIG5	Base	00950	01114	PC data: SBCS (IBM BIG5)
			00947	PC data: DBCS including 13493 CNS, 566 IBM selected, 6204 user-defined characters
964 EUCTW	Base	00964	00367	G0: ASCII
			00960	G1: CNS 11643 plane 1
			00961	G2: CNS 11643 plane 2

Table 32. Traditional Chinese, EBCDIC

SRVERCP	in	CCSID	CPGID	Comments
937	Base	00937	00037	Host: Extended SBCS
			00835	Host: DBCS including 6204 user-defined characters

Unicode data

CICS on System/390 provides limited support for Unicode-encoded character data. The support allows workstations to share UCS-2 or UTF-8 encoded data with the System/390 provided, that no conversion is required.

Table 33. Unicode

CLINTCP SRVERCP	in	CCSID	CPGID	Comments
1200 UCS-2	T01	01200	01400	UCS-2 level 3, maximal (growing) character set
1208 UTF-8	T01	01200	01400	UTF-8 based on UCS-2 level 3, maximal (growing) character set
13488	T01	13488	01400	UCS-2 level 1 (level 3 tolerant), subset (fixed) character set

Binary data

For binary data you can specify, on the DFHCNV TYPE=FIELD macro, either:

DATATYP=BINARY

The default format for binary data received by CICS on System/390 is System/390 (big-endian).

DATATYP=NUMERIC

The default format for binary data received by CICS on System/390 is INTEL (little-endian).

The default binary format can be overridden. This allows CICS for OS/2 or CICS for Windows NT (which can hold binary data in either System/390 or INTEL format) and CICS on Open Systems (which holds binary data in System/390 format) to

access the same resource. It is therefore important that you code a DFHCNV TYPE=FIELD macro for every binary field. If the resource can be accessed from both CICS for OS/2 (or CICS for Windows NT) and CICS on Open Systems, you must set the default binary format to that used by CICS for OS/2.

You should ensure that the format of each individual binary data field—BINARY or NUMERIC—is consistent across all INTEL platforms.

The conversion process

This section describes in more detail how data conversion works in a CICS on System/390 system.

Components

The CICS or user-supplied mirror transactions do the System/390 conversions, using the following CICS components:

DFHCNV

The conversion table. For each resource for which conversion is required, DFHCNV contains a **conversion template**. A conversion template is a table entry defining fields in a data area that are to be converted, and the conversion method to be applied to each field.

You define the DFHCNV table with the DFHCNV resource definition macros, described on pages 59–68.

DFHCCNV

The CICS program that drives the conversion process. DFHCCNV uses the DFHCNV table to determine the required conversions. It applies standard conversion to those fields in the conversion templates for which nonstandard, user-handled conversion is not specified.

DFHUCNV

A user-replaceable program that allows you to override the standard conversions applied by CICS. You can use DFHUCNV to apply your own conversion logic to specific data fields. (How to do this is described in “User/CICS conversion” on page 56.)

You can use the supplied program as a model on which to base your own version.

Process

This section describes the standard conversions that can be applied by DFHCCNV to specific fields in a conversion template. Other types of conversion are possible, if you write a DFHUCNV program.

Character data

Character data can be converted:

- From ASCII to EBCDIC, on receipt of a request from CICS for OS/2, CICS for Windows NT, or CICS on Open Systems, before invoking the EXEC interface
- From EBCDIC to ASCII, on return from the EXEC interface, before transmitting the response to CICS for OS/2, CICS for Windows NT, or CICS on Open Systems.

the conversion process

The translation tables shipped with CICS—for example, those used for conversion between code page 037 (US EBCDIC) and code page 437 (US ASCII)—conform to the standards described in the *IBM Character Data Representation Architecture Level 2 - Registry*, SC09-1391.

Binary data

Binary data can be converted:

- From INTEL to System/390 format, on receipt of a request from CICS for OS/2 or CICS for Windows NT.
- From System/390 to INTEL format, before transmitting the response to CICS for OS/2 or CICS for Windows NT.

In CICS for OS/2 and CICS for Windows NT, COBOL/2™ programs can be compiled so that binary and packed decimal fields are in System/390-compatible format and require no conversion at the System/390 (see page 66).

Standard and nonstandard conversion

Consider a single resource, a file for example, that requires data conversion. There are three possibilities; you can use:

- CICS-only conversion—all data fields are handled by the standard CICS conversion program, DFHCCNV
- User/CICS conversion—a combination of nonstandard and standard conversion, in which some data fields are handled by user code in DFHUCNV and some by DFHCCNV
- User-only conversion—all data fields are handled by DFHUCNV.

CICS-only conversion

The resource contains no data fields that require nonstandard conversion; all can be converted by standard means. You must:

1. Create a conversion template, using the DFHCNV macros described in “Defining the conversion table” on page 59. This enables DFHCCNV to handle the resource.
2. Specify USREXIT=NO (or let it default) on the DFHCNV TYPE=ENTRY macro that defines the resource. This prevents DFHUCNV from being called unnecessarily.
3. Do not specify DATATYP=USERDATA on any of the DFHCNV TYPE=FIELD macros that define the data fields.

User/CICS conversion

The resource contains some fields that can be converted by standard means, and some that require nonstandard conversion. You must:

1. Create a conversion template.
2. Specify USREXIT=YES on the DFHCNV TYPE=ENTRY macro that defines the resource, so that DFHUCNV is called.
3. Specify DATATYP=USERDATA on the DFHCNV TYPE=FIELD macros that define the nonstandard data fields. Optionally, also define nonstandard fields with a USRTYPE value in the range X'50' through X'80' (see page 67). These values are passed to your user program, and can be used to distinguish between different types of nonstandard field.

Define standard fields as DATATYP=CHARACTER, PD, BINARY, GRAPHIC, or NUMERIC, as appropriate.

4. Supply a user-written version of DFHUCNV to handle the nonstandard fields.

“User-replaceable conversion program” on page 76 gives a description and listing of DFHUCNV, with guidance on how to use it as a basis for your own conversion program. (The default version supplied with CICS handles only temporary storage requests for which templates have been defined.)

User-only conversion

The resource contains no fields that can be converted by standard means; all require nonstandard conversion. There are two methods of enabling user-only conversion:

Method 1: This is the *recommended method*:

1. Create a conversion template.
2. Specify USREXIT=YES on the DFHCNV TYPE=ENTRY macro that defines the resource.
3. Specify DATATYP=USERDATA on *all* the DFHCNV TYPE=FIELD macros that define the data fields. Optionally, define all fields with a USRTYPE value in the range X'50' through X'80'.
4. Supply a user-written version of DFHUCNV to handle all fields.

Method 2:

1. Do not create a conversion template. If there is no template for a resource, CICS invokes DFHUCNV (but not DFHCCNV), on the assumption that it is to handle all conversion.
2. Supply a user-written version of DFHUCNV to handle all fields.

This method is *not recommended*, because:

- The parameter list passed to your DFHUCNV program does not contain as much information as it does if you define a conversion template. The fields that contain data are listed in “Parameter list (DFHUVNDS)” on page 77.
- It relies on the following:
 - That all workstations connected to the CICS on System/390 server share a common encoding (for example, ASCII 437) for character data, and a common format (for example, INTEL) for binary data
 - That your version of DFHUCNV knows the ASCII encoding and binary format, and knows the corresponding EBCIDIC encoding
 - That DFHUCNV provides the appropriate ASCII/EBCIDIC translation tables.

Sequence of conversion processing

The sequence of conversion processing is as follows:

1. Unless USREXIT=NO is specified in the DFHCNV TYPE=ENTRY macro that defines the conversion template for the resource, DFHCCNV links to DFHUCNV, passing the parameter list described in “Parameter list (DFHUVNDS)” on page 77.

Notes:

- a. If you have not defined a template, DFHUCNV is invoked, on the assumption that the user program is to handle all conversions for the resource.
 - b. DFHUCNV *must* be present in your system unless *all* DFHCNV TYPE=ENTRY macros specify USREXIT=NO.
2. If a conversion template is defined for the resource, DFHUCNV is responsible for converting any fields with a type in the user-data range.

the conversion process

If no conversion template is defined for the resource, DFHUCNV is responsible for determining the format of the data, and for converting all appropriate fields.

3. On return from DFHUCNV, DFHCCNV carries out any standard conversions specified in the conversion template for fields that are not subject to user-defined conversion.
4. The shipped request is executed.

If data conversion is required, a DPL request from CICS for OS/2 or CICS for Windows NT should not use the TRANSID option to specify a transaction other than the default CPML, which is required to trigger conversion.

Figure 12 summarizes System/390 conversion procedures.

Mirror transaction in CICS on System/390

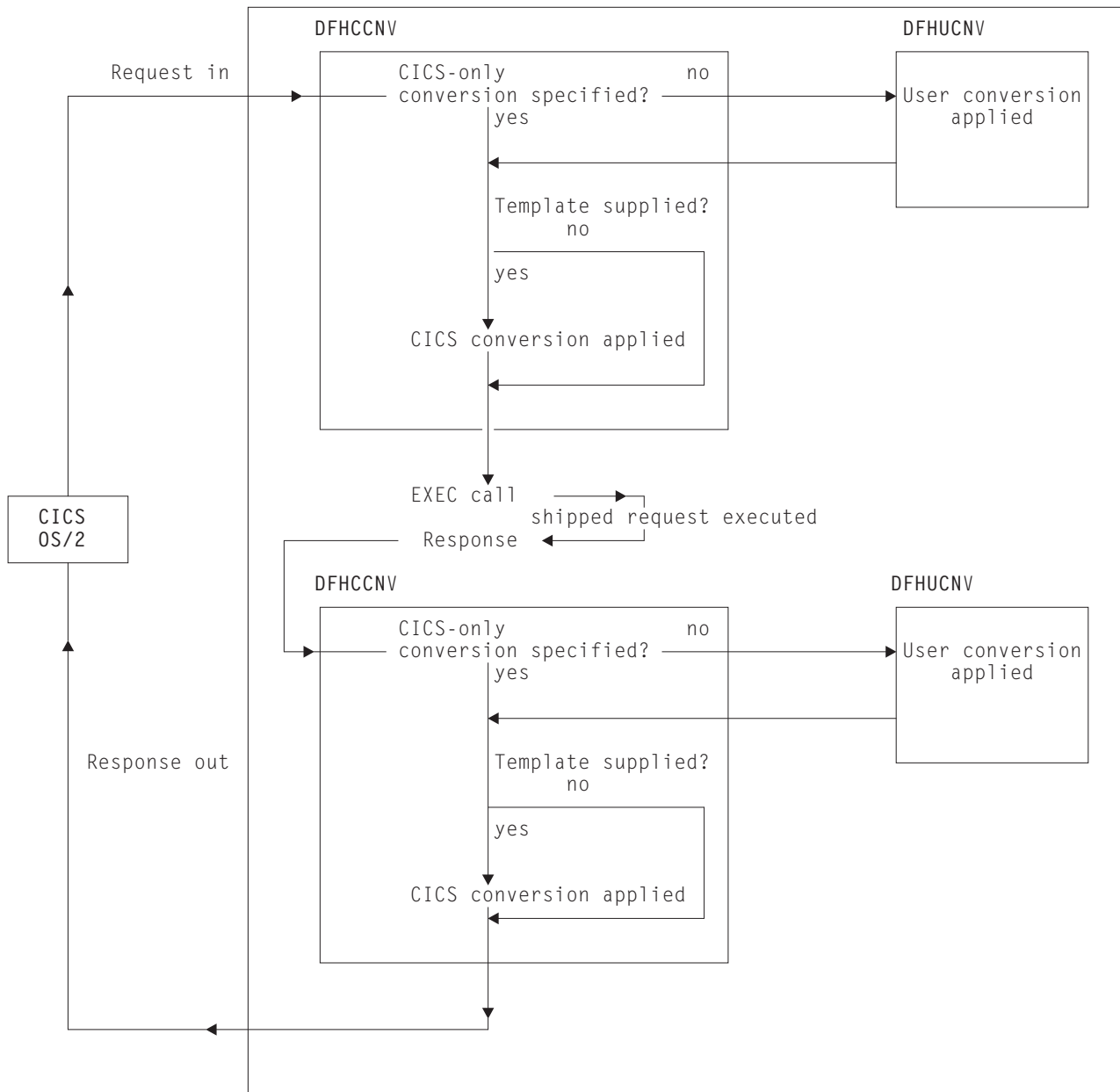


Figure 12. Data conversion in CICS on System/390

Resource definition to enable data conversion

To allow data conversion at the System/390, the following resources must be defined to the CICS on System/390 system:

- DFHCNV, conversion table
- DFHCCNV, standard conversion program
- DFHUCNV, user-defined conversion program.

For details of how to create the conversion table, see the next section “Defining the conversion table”.

Defining the conversion table

You define the conversion table with DFHCNV resource definition macros, described below. If you are running CICS on System/390 in an MVS/XA™ or MVS/ESA™ environment, see “Assembling and link-editing the conversion programs” on page 76.

Note: The output of the DFHCNV macro assembly contains templates specifying resource conversion requirements and conversion tables to enable the required conversions. User-generated conversion tables must be placed in the DFHCNV macro source.

DFHCNV macro types

The following macros define the conversion table:

- DFHCNV TYPE=INITIAL defines the beginning of the conversion table. It gives a list of valid code pages.
- DFHCNV TYPE=ENTRY specifies a name and type to uniquely identify a data resource. There must be one for each resource for which conversion is required (no conversion is applied to a resource that is not defined in a DFHCNV TYPE=ENTRY macro). The entry for one resource is concluded by the next TYPE=ENTRY statement, or by the end of the table. The code page to be used is specified.
- DFHCNV TYPE=KEY applies only to an FC entry. Use this macro only if a record might need to be accessed by key (if records are always accessed by relative record number or relative byte address, do not code a TYPE=KEY macro). When used, this macro must immediately follow a TYPE=ENTRY macro, and must be followed by one or more TYPE=FIELD macros, which define the data conversion to be applied to the key.
- DFHCNV TYPE=SELECT defines selection of a record ¹¹ for data conversion based on the value of a field in the record. Each TYPE=SELECT macro is followed by one or more TYPE=FIELD macros, which define the data conversion to be applied if the record satisfies the test defined in the TYPE=SELECT macro. The last TYPE=SELECT macro for each entry is an OPTION=DEFAULT macro, which defines the conversion to be applied to a record that satisfies no preceding TYPE=SELECT macro.
- DFHCNV TYPE=FIELD specifies the position and length of a field, and the conversion to be applied to it. There must be a TYPE=FIELD macro for each field for which conversion is required.
- DFHCNV TYPE=FINAL concludes the conversion table definition.

11. FC record, TS data, TD data, IC start “from” data, or COMMAREA transmitted with DPL

defining the conversion table

Conversion and key templates

Templates are table entries defining fields in a data area or key that are to be converted and the conversion method to be applied to each field. A conversion template is defined by one or more DFHCNV TYPE=FIELD macros following a DFHCNV TYPE=SELECT macro. A key template consists of one or more DFHCNV TYPE=FIELD macros following a DFHCNV TYPE=KEY macro. Each type of template is terminated by the next non-FIELD macro in the table definition. Figure 14 on page 61 shows templates within a complete conversion table definition.

Example sequence—DFHCNV macros

Figure 14 on page 61 shows a typical sequence of DFHCNV macros. The figure is annotated to show the sets of entries that correspond to resource entries, conversion templates, and key templates. (The indentation is to illustrate nesting. When coding the macros, as with all CICS resource definition macros, observe assembler rules.)

Conversion table for initial program verification (IVP)

When running the IVP jobs for your CICS on System/390 system, you need the conversion table whose source is given in Figure 13. You don't need to code all these macros. You can generate exactly the same conversion table by assembling the special macro, DFHCNV TYPE=IVP.

Figure 13 is a simple example of a conversion table definition.

All the fields are character, so only a single TYPE=SELECT macro is needed. It specifies OPTION=DEFAULT, and has a single TYPE=FIELD macro to define the whole data record.

The TYPE=KEY macro is followed by a single TYPE=FIELD macro, which redefines the first six bytes of the data record.

```
DFHCNV TYPE=INITIAL
DFHCNV TYPE=ENTRY,RTYPE=FC,RNAME=FILEA,USREXIT=NO
DFHCNV TYPE=KEY
DFHCNV TYPE=FIELD,OFFSET=0,DATATYP=CHARACTER,DATALEN=6, LAST=YES
DFHCNV TYPE=SELECT,OPTION=DEFAULT
DFHCNV TYPE=FIELD,OFFSET=0,DATATYP=CHARACTER,DATALEN=80, LAST=YES
DFHCNV TYPE=FINAL
```

Figure 13. Conversion table for IVP

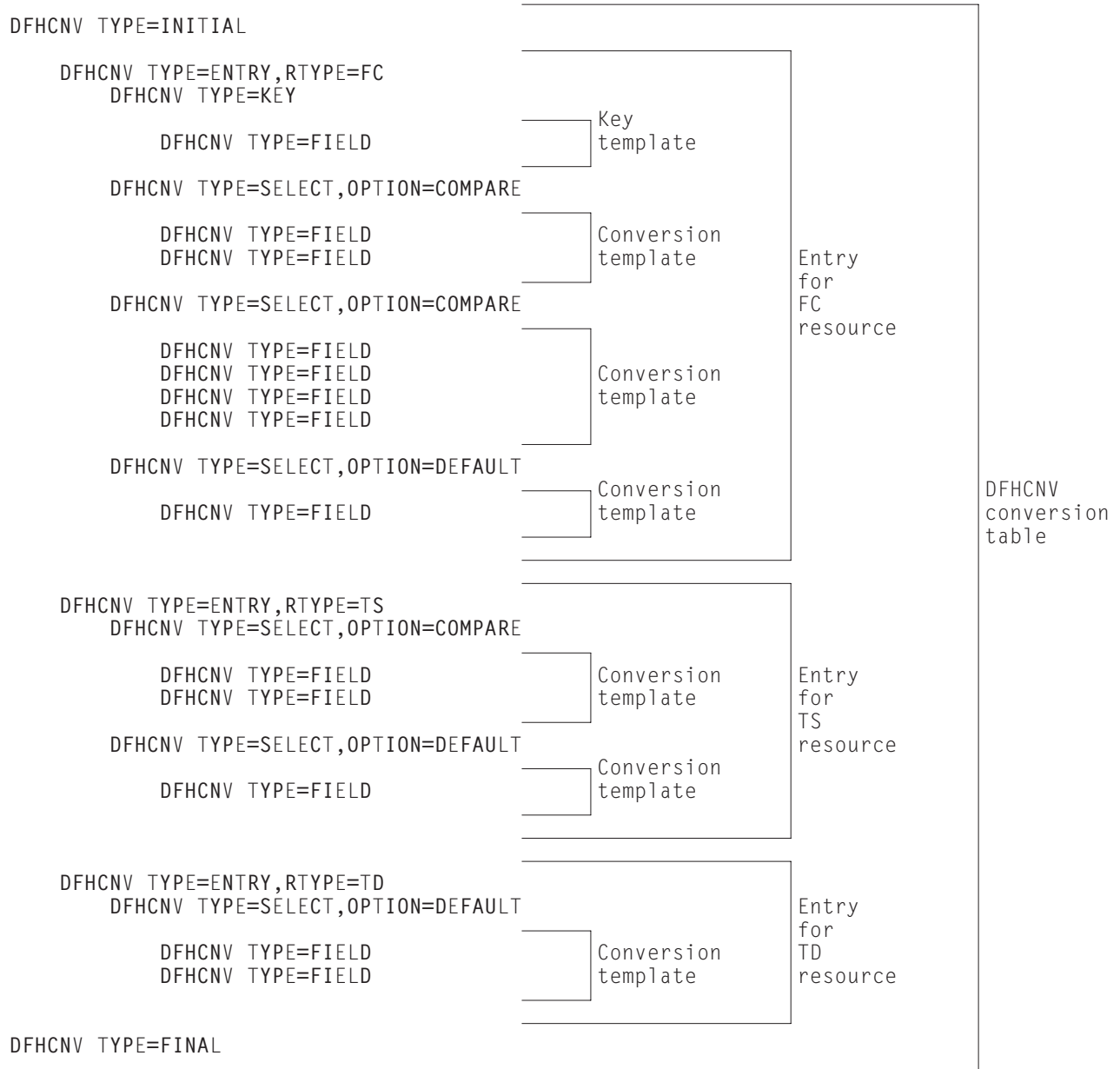


Figure 14. Example of DFHCNV macro sequence

DFHCNV TYPE=INITIAL

The DFHCNV TYPE=INITIAL macro has the following format:

	DFHCNV	TYPE=INITIAL [,CLINTCP={437 nnnn[, nnnn, ...}}] [,SRVERCP={037 nnnn[, nnnn, ...}}] [,CDEPAGE=nnnn[, nnnn...]]
--	--------	--

TYPE=INITIAL

Defines the beginning of the conversion table.

defining the conversion table

CLINTCP={437|nnnn[,nnnn, ...]}

The first operand defines the default client code page to be used when the CLINTCP and CDEPAGE operands are omitted from a DFHCNV TYPE=ENTRY macro.

You can specify further code pages; they are validated but are not used.

For an explanation of code pages, and a list of those that you can specify, see "Character data" on page 45.

SRVERCP={037|nnnn[,nnnn, ...]}

The first operand defines the default server code page to be used when the SRVERCP and CDEPAGE operands are omitted from a DFHCNV TYPE=ENTRY macro.

You can specify further code pages; they are validated but are not used.

For an explanation of code pages, and a list of those that you can specify, see "Character data" on page 45.

CDEPAGE=nnnn[,nnnn...]

(Retained for compatibility with earlier releases. Do not use for new definitions. There is no need to change existing definitions that support communication with CICS for OS/2.)

Each possible value is equivalent to a pair of CLINTCP and SRVERCP entries or (for user-defined conversion) to a SRVERCP entry.

437

Is equivalent to:
CLINTCP=437
SRVERCP=037

932K

Is equivalent to:
CLINTCP=932
SRVERCP=930

932

Is equivalent to:
CLINTCP=932
SRVERCP=931

USR

Is equivalent to:
SRVERCP=USR

USRD

Is equivalent to:
SRVERCP=USRD

DFHCNV TYPE=ENTRY

The DFHCNV TYPE=ENTRY macro instruction has the following format:

	DFHCNV	<pre> TYPE=ENTRY ,RTYPE={FC TS TD IC PC} ,[CLINTCP=nnnn[,nnnn,. . .]] ,[SRVERCP=nnnn] ,{RNAME=resourcename XRNAME=xxxxxxxxxxxxxxxxxxx} ,[USREXIT={YES NO}] ,[CDEPAGE=nnnn] </pre>
--	--------	---

TYPE=ENTRY

Specifies that this macro defines a resource by name and type.

RTYPE={FC|TS|TD|IC|PC}

Specifies the type of resource:

- FC** A file
- TS** A temporary storage queue
- TD** A transient data queue
- IC** An interval control start with a FROM operand
- PC** An EXEC CICS LINK command (DPL) with a COMMAREA.

CLINTCP=nnnn[,nnnn,. . .]

The first operand defines the default client code page to be used.

You can specify further code pages; they are validated but are not used.

For an explanation of code pages, and a list of those that you can specify, see “Character data” on page 45.

SRVERCP=nnnn

The operand defines the server code page to be used.

For an explanation of code pages, and a list of those that you can specify, see “Character data” on page 45.

RNAME=resourcename

Specifies the name of the resource in up to eight characters. If shorter, it is padded with blanks; if longer, it is truncated. The name can be:

- A System/390 file name (up to eight characters)
- A TS queue name (up to eight characters)
- A TD queue name (up to four characters)
- An IC start transaction id (up to four characters)
- The name of the program being linked (up to eight characters).

XRNAME=xxxxxxxxxxxxxxxx (RTYPE=TSIC only)

Specifies the resource name in hexadecimal notation. It can include up to 16 hexadecimal digits, padded with blanks if necessary.

USREXIT={YES|NO}

Specifies whether the user data conversion exit is called.

YES

The user data conversion exit is called. Code this only if you need the user conversion exit to convert some data for this resource.

defining the conversion table

NO

The user data conversion exit is not called. This eliminates the overhead of calling the exit unnecessarily.

CDEPAGE=nnnn

(Not for new definitions. Retained for compatibility with earlier releases. Do not change existing definitions that support communication with CICS for OS/2.)

The code page must be one of those entered in the CDEPAGE option of the DFHCNV TYPE=INITIAL macro. Each possible value is equivalent to a pair of CLINTCP and SRVERCP entries or (for user-defined conversion) to a SRVERCP entry. The CLINTCP and SRVERCP values to which each value resolves are given in the description of the CDEPAGE option of the DFHCNV TYPE=INITIAL macro.

DFHCNV TYPE=KEY

The DFHCNV TYPE=KEY macro is valid only for FC TYPE requests, and, if coded, must immediately follow a DFHCNV TYPE=ENTRY macro. The macro has the following format:

	DFHCNV	TYPE=KEY
--	--------	----------

TYPE=KEY

Indicates the start of conversions to be applied to a key. This macro is not required if access is only by RRN or RBA. If access is by key but no TYPE=KEY statement is present, the key is not converted. You must provide matching conversion details (DFHCNV TYPE=FIELD macros) for the key, as part of each conversion template that applies to this file, or an INVREQ condition may be returned on the file control EXEC CICS request.

DFHCNV TYPE=SELECT

The DFHCNV TYPE=SELECT macro instruction has the following format:

	DFHCNV	<pre>TYPE=SELECT ,OPTION={COMPARE DEFAULT} ,OFFSET=nnn {,DATA='dd...dd' ,XDATA='xx...xx'}</pre>
--	--------	---

TYPE=SELECT

Indicates the start of conversion definitions (DFHCNV TYPE=FIELD macros) to be applied to a record that satisfies the comparison defined in this macro. If the defined comparison is not satisfied by the data in the record, the conversion program (DFHCCNV) skips to the next TYPE=SELECT macro, until it finds a match or reaches the OPTION=DEFAULT macro. Every TYPE=SELECT macro must be followed by at least one TYPE=FIELD macro.

OPTION={COMPARE|DEFAULT}

States the basic selection options:

COMPARE

Indicates that the data should be converted according to the specifications in the following DFHCNV TYPE=FIELD macros, if the record satisfies the comparison defined in this macro (OFFSET and DATA or XDATA options).

DEFAULT

Indicates that the data should be converted according to the specifications in the following DFHCNV TYPE=FIELD macros, if the record has not satisfied the comparison defined in any previous DFHCNV TYPE=SELECT COMPARE macro.

For each resource entry (started by a TYPE=ENTRY macro) the last TYPE=SELECT macro must specify OPTION=DEFAULT. No other TYPE=SELECT macro in the entry should specify OPTION=DEFAULT.

The following options are ignored if OPTION=DEFAULT is coded.

OFFSET=nnnn

Specifies the byte offset in the record at which the comparison should be made, up to a maximum of 65535.

DATA='dd...dd'

(Use only if the data to be tested is defined as DATATYP=CHARACTER, SOSI=NO)

Specifies the comparison data as a character string, with a maximum length of 254 characters. Because you specify this data on the System/390 system, it is in EBCDIC. The system converts the incoming data from ASCII to EBCDIC before checking it against the comparison data, so that EBCDIC is compared with EBCDIC. Outgoing data is in EBCDIC, so the comparison is made in EBCDIC without conversion.

defining the conversion table

XDATA='xx...xx'
(Use if DATA option is not used)

Specifies the comparison data as a hexadecimal string, with an even number of digits, maximum length 254 digits. Data is compared against this field, without conversion.

DFHCNV TYPE=FIELD

The DFHCNV TYPE=FIELD macro instruction occurs as many times as needed, and has the following format:

	DFHCNV	TYPE=FIELD ,OFFSET=nnnn ,DATATYP={CHARACTER PD BINARY USERDATA GRAPHIC NUMERIC} ,USRTYPE=xx ,DATALEN=nnnn [,LAST=YES] [,SOSI=YES NO]
--	--------	--

TYPE=FIELD

Specifies start of conversion specifications for a data field. There must be one such statement for each field in a record, and every TYPE=FIELD macro must be preceded by a TYPE=SELECT macro.

OFFSET=nnnn

Specifies the byte offset in the record or key at which the conversion should start, up to a maximum of 65535. (For TYPE=KEY conversions, this is the byte offset from the start of the key not from the start of the record. The file's FCT entry defines the offset of the first byte of the key from the start of the record.)

DATATYP={CHARACTER|PD|BINARY|USERDATA|GRAPHIC|NUMERIC}

Specifies the type of conversion required:

CHARACTER

Specifies character fields.

PD

Specifies packed decimal data in System/390 format.

If workstation packed decimal (PD) fields are to be transmitted to a System/390, specify the IBMCOMP and SIGN EBCDIC directives with the CICSCOMP command used to compile a CICS for OS/2 COBOL/2 application program. If this program sends data to the System/390, packed decimal fields are in System/390-compatible format. Define such fields as PD (no conversion needed). Any packed decimal data in non-System/390 format (for example, workstation native mode) should be defined for USERDATA conversion, and the user-replaceable program DFHUCNV must contain the necessary conversion code.

BINARY

Specifies binary data in System/390 format.

The CICS for OS/2 user can specify the IBMCOMP directive with the CICSCOMP command used to compile a CICS for OS/2 COBOL/2 application program that creates binary fields in System/390-compatible format.

By default, COBOL/2 binary fields are held on the workstation in INTEL format, and should be defined as NUMERIC. C language integer fields are always held on the workstation in INTEL format, and should be defined as NUMERIC.

By default, BINARY data is not converted. This default action can be overridden to allow requests from platforms that support different binary architectures to access the same System/390 resource using the same conversion table.

USERDATA

Specifies data to be converted by the user-replaceable program DFHUCNV. The DFHCCNV conversion code bypasses these fields. See the USRTYPE operand below.

GRAPHIC

Specifies fields that contain DBCS characters only.

NUMERIC

Specifies that binary fields held on the workstation in INTEL format (for example, C Language integer datatype) need to be converted to System/390 format. Integers (four bytes) or short integers (two bytes) can be converted.

USRTYPE=xx

Specifies a value that is made available to the user-replaceable conversion program DFHUCNV. The values you provide can be in the range 80 to 128 (X'50' to X'80'). The default value is 80 (X'50'). If more than one type of user-defined conversion is possible, you can use this value to specify to DFHUCNV what conversion is needed for each field.

This option is ignored if DATATYP=USERDATA is not specified.

DATALEN=n

Specifies the length of the data field to be converted, in bytes, up to a maximum of 65535. For variable length fields, specify the maximum possible length.

If DATATYP=NUMERIC, DATALEN must be 2 or 4.

LAST=YES

Specifies that this is the last field definition for this TYPE=SELECT statement.

SOSI=YESINO

Enter YES for a mixed string containing SBCS and DBCS characters; enter NO for an SBCS string. This field is valid only if DATATYPE=CHARACTER has been entered in this macro. The default is NO.

defining the conversion table

DFHCNV TYPE=FINAL

The DFHCNV TYPE=FINAL macro instruction ends the table. It must occur only once, as the last definition.

	DFHCNV	TYPE=FINAL
--	--------	------------

Hints on coding the macros

CICS does not check the order of the fields defined in the table, or for overlap. This is relevant to the first two hints below:

1. Define entries for the most frequently-used resources first, to reduce search time.
2. Define USERDATA fields in consecutive entries. This reduces the time needed by your conversion program to scan the template.
3. For variable-length fields, define the maximum length required. (Comparisons and conversions are applied to the shorter of the actual data length or the template length. For example, if the data is 100 bytes long but the template describes 120 bytes, up to 100 bytes are converted. If the data is 100 bytes and the template describes 80 bytes, only 80 bytes are converted.)
4. If function-shipped data is accessed only by a remote CICS for OS/2, CICS for Windows NT, or CICS on Open Systems system, and never by the System/390 system, there is no need to provide conversion details. A typical case is a System/390 file used to pass data between CICS for OS/2 users.
5. CICS for OS/2 and CICS for Windows NT assume that record key formats are not redefinable, so they must be the same for all redefined record types. If a resource has a key template, then all conversion templates for that resource must exactly replicate the key conversion specified in the key template.

User-defined conversion tables

If you specify SRVERCP=USR or USRD in a DFHCNV TYPE=ENTRY macro, you must provide user-defined conversion tables. The standard conversion program (DFHCCNV) uses these tables, and they are made available to the user-replaceable conversion program, DFHUCNV.

Place your user-defined conversion tables in the DFHCNV macro source, anywhere after the DFHCNV TYPE=INITIAL macro. For source readability, the best place is probably after the DFHCNV TYPE=FINAL macro.

The following are descriptions of the types of table you may need to define and the way to label each type.

SRVERCP=USR

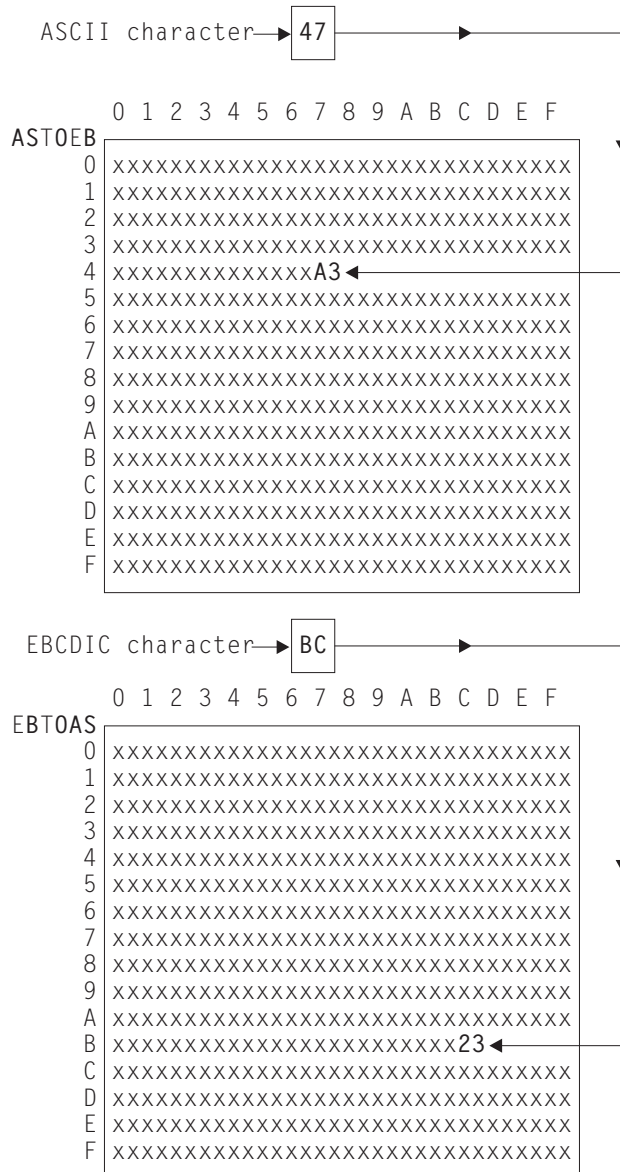
You must provide two character conversion tables, labelled ASTOEB and EBTOAS. Each table must be 256 bytes long. ASTOEB is used for ASCII to EBCDIC conversion and EBTOAS is used for EBCDIC to ASCII conversion. The hexadecimal value of a character byte is used as an offset in the conversion table to obtain the converted value of the character. Figure 15 on page 70 illustrates this process.

SRVERCP=USRD

You must provide DBCS character conversion tables labelled DBASTOEB and DBEBTOAS, in the DFHCNV source. These must be after the DFHCNV TYPE=INITIAL macro, but otherwise anywhere in the source. Each table must be a list of 256 four-byte pointers and 256 pairs of 256-byte translate tables. The first byte of a DBCS character is used as an index to the list of pointers. Using the first byte of the DBCS character as a hexadecimal offset in the list, the pointer found is the address of a pair of 256-byte translate tables. The second byte of the DBCS character is used as an offset in each of the two 256-byte translate tables to obtain the first and second bytes of the converted DBCS character. Figure 16 on page 71 illustrates this process.

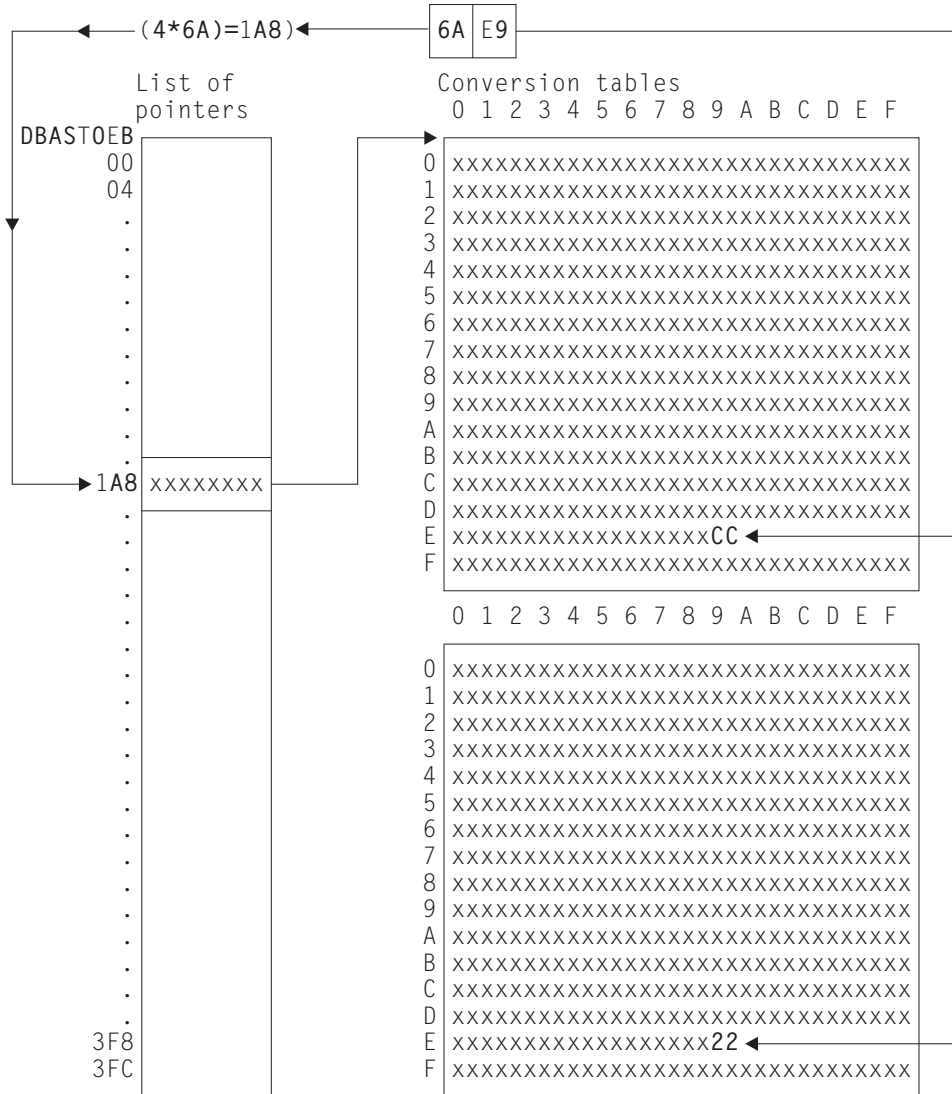
You must also provide an SBCS conversion table as specified under USR above.

user-defined conversion tables



In this example, the ASCII character X'47' converts to the EBCDIC character X'A3', and the EBCDIC character X'BC' converts to the ASCII character X'23'. These values have no significance, and are used simply to illustrate the structure of the conversion tables.

Figure 15. Structure of SBCS conversion tables



In this example, the double-byte character X'6AE9' converts to X'CC22'. The value, at offset 6A in the pointer list, is the address of a pair of 256-byte translate tables. At offset E9 in these tables, the byte values are X'CC' and X'22' respectively. These are random values, used purely for illustration.

This is an ASCII-EBCDIC conversion, because the pointer list is labelled DBASTOEB. A complete set of ASCII-EBCDIC tables contains 256 pairs of 256-byte tables, one pair for each possible value of the first byte of a double-byte character. DBEBTOAS is the label of a similar set of EBCDIC-ASCII tables.

Figure 16. Structure of DBCS conversion tables

Invalid and undefined DBCS characters

In ASCII and EBCDIC, certain code ranges are valid DBCS code. Any double-byte value outside these ranges is an invalid DBCS character. In the supplied conversion tables, invalid DBCS characters convert to X'FFFF', as defined by the code page architecture.

Within the valid code range, several thousand double-byte values are defined as actual DBCS characters. A double-byte value within the valid code range, but not defined as a DBCS character, is an undefined DBCS character. In the supplied

user-defined conversion tables

conversion tables, undefined DBCS characters convert to X'FEFD' (ASCII-EBCDIC), and X'FCFB' (EBCDIC-ASCII), as defined by the code page architecture.

User-defined tables should follow the above conventions for invalid and undefined characters.

Example macros

Figure 17 shows an example of a record layout for a file called VSAM99. The key is offset 0 for length 6, and the record contains no redefinition.

```
02  FILEREC.
03  STAT      PIC X.
03  NUMB      PIC X(6).
03  NAME      PIC X(20).
03  ADDR      PIC X(20).
03  PHONE     PIC X(8).
03  DATEX     PIC X(8).
03  AMOUNT    PIC X(8).
03  COMMENT   PIC X(9).
03  COUNTER1  PIC 9999 USAGE COMP-4.
03  COUNTER2  PIC 9999 USAGE COMP-4.
03  ADDLCMT   PIC X(30).
```

Figure 17. Record layout for VSAM99

Figure 18 gives a full set of conversion macros for file VSAM99. Figure 19 shows the same conversion expressed more briefly, by combining adjoining fields of the same type.

```
DFHCNV TYPE=INITIAL,CDEPAGE=437
DFHCNV TYPE=ENTRY,RTYPE=FC,RNAME=VSAM99
DFHCNV TYPE=KEY
DFHCNV TYPE=FIELD,OFFSET=0,DATATYP=CHARACTER,DATALEN=6, LAST=YES
DFHCNV TYPE=SELECT,OPTION=DEFAULT
DFHCNV TYPE=FIELD,OFFSET=00,DATATYP=CHARACTER,DATALEN=1
DFHCNV TYPE=FIELD,OFFSET=01,DATATYP=CHARACTER,DATALEN=6
DFHCNV TYPE=FIELD,OFFSET=07,DATATYP=CHARACTER,DATALEN=20
DFHCNV TYPE=FIELD,OFFSET=27,DATATYP=CHARACTER,DATALEN=20
DFHCNV TYPE=FIELD,OFFSET=47,DATATYP=CHARACTER,DATALEN=8
DFHCNV TYPE=FIELD,OFFSET=55,DATATYP=CHARACTER,DATALEN=8
DFHCNV TYPE=FIELD,OFFSET=63,DATATYP=CHARACTER,DATALEN=8
DFHCNV TYPE=FIELD,OFFSET=71,DATATYP=CHARACTER,DATALEN=9
DFHCNV TYPE=FIELD,OFFSET=80,DATATYP=BINARY,DATALEN=2
DFHCNV TYPE=FIELD,OFFSET=82,DATATYP=BINARY,DATALEN=2
DFHCNV TYPE=FIELD,OFFSET=84,DATATYP=CHARACTER,DATALEN=30, LAST=YES
DFHCNV TYPE=FINAL
```

Figure 18. Full description of VSAM99


```
DFHCNV TYPE=INITIAL,CDEPAGE=437
DFHCNV TYPE=ENTRY,RTYPE=FC,RNAME=VSAM99
DFHCNV TYPE=KEY
DFHCNV TYPE=FIELD,OFFSET=0,DATATYP=CHARACTER,DATALEN=6, LAST=YES
DFHCNV TYPE=SELECT,OPTION=DEFAULT
DFHCNV TYPE=FIELD,OFFSET=00,DATATYP=CHARACTER,DATALEN=80
DFHCNV TYPE=FIELD,OFFSET=80,DATATYP=BINARY,DATALEN=4
DFHCNV TYPE=FIELD,OFFSET=84,DATATYP=CHARACTER,DATALEN=30, LAST=YES
DFHCNV TYPE=FINAL
```

Figure 19. Condensed description of VSAM99

Note: Be careful when combining adjoining fields, even if they are of the same data type. Do not combine NUMERIC fields. Do not combine fields defined as CHARACTER, if SOSI=YES is specified for one or more of them. Whether you can combine USERDATA fields depends on user-defined data structures and conversion code.

Figure 20 shows a redefined record layout for file VSAM99. Figure 21 shows a set of conversion macros for the redefined record layout in Figure 20.

```
02  FILEREC.
03  STAT          PIC X.
03  NUMB          PIC X(6).
03  NAME          PIC X(20).
03  ADDR          PIC X(20).
03  PHONE         PIC X(8).
03  DATEX         PIC X(8).
03  AMOUNT        PIC X(8).
03  COMMENT       PIC X(9).
03  VARINF1.
03  COUNTER1      PIC 9999 USAGE COMP-4.
03  COUNTER2      PIC 9999 USAGE COMP-4.
03  ADDLCMT       PIC X(30).
03  VARINF2 REDEFINES VARINF1.
03  COUNTER1      PIC 9999 USAGE COMP-4.
03  COUNTER2      PIC 9999 USAGE COMP-4.
03  COUNTER3      PIC 9999 USAGE COMP-4.
03  COUNTER4      PIC 9999 USAGE COMP-4.
03  ADDLCMT2     PIC X(26).
```

Figure 20. Redefined record layout for VSAM99

data conversion examples

```
DFHCNV TYPE=INITIAL
DFHCNV TYPE=ENTRY,RTYPE=FC,RNAME=VSAM99
DFHCNV TYPE=KEY
DFHCNV TYPE=FIELD,OFFSET=0,DATATYP=CHARACTER,DATALEN=6, LAST=YES
*
* If offset 0 is a character 'X' use the following
* conversion definitions:
*
DFHCNV TYPE=SELECT,OPTION=COMPARE,OFFSET=00,DATA='X'
DFHCNV TYPE=FIELD,OFFSET=00,DATATYP=CHARACTER,DATALEN=80
DFHCNV TYPE=FIELD,OFFSET=80,DATATYP=BINARY,DATALEN=4
DFHCNV TYPE=FIELD,OFFSET=84,DATATYP=CHARACTER,DATALEN=30, LAST=YES
*
* Otherwise use the following (default)
* conversion definitions
*
DFHCNV TYPE=SELECT,OPTION=DEFAULT
DFHCNV TYPE=FIELD,OFFSET=00,DATATYP=CHARACTER,DATALEN=80
DFHCNV TYPE=FIELD,OFFSET=80,DATATYP=BINARY,DATALEN=8
DFHCNV TYPE=FIELD,OFFSET=88,DATATYP=CHARACTER,DATALEN=26, LAST=YES
DFHCNV TYPE=FINAL
```

Figure 21. Description for redefined record layout for VSAM99

Figure 22 shows user-defined conversion tables, EBTOAS and ASTOEB, illustrating how they are preceded with DFHCNV macros in the source that is submitted to the assembler.

data conversion examples

```

*
LABL1  DFHCNV TYPE=INITIAL,CDEPAGE=USR
*
      DFHCNV TYPE=ENTRY,RTYPE=FC,RNAME=VSAM80
      DFHCNV TYPE=KEY
      DFHCNV TYPE=FIELD,OFFSET=0,DATATYP=BINARY,DATALEN=2
      DFHCNV TYPE=FIELD,OFFSET=2,DATATYP=CHARACTER,DATALEN=4,   X
          LAST=YES
LABLX  DFHCNV TYPE=SELECT,OPTION=COMPARE,OFFSET=6,XDATA='C1C2C3'
      DFHCNV TYPE=FIELD,OFFSET=0,DATATYP=BINARY,DATALEN=2
      DFHCNV TYPE=FIELD,OFFSET=2,DATATYP=CHARACTER,DATALEN=4
      DFHCNV TYPE=FIELD,OFFSET=9,DATATYP=CHARACTER,DATALEN=8,   X
          LAST=YES

      :
      :
      DFHCNV TYPE=ENTRY,RTYPE=TS,RNAME=ABCD
      DFHCNV TYPE=SELECT,OPTION=DEFAULT
      DFHCNV TYPE=FIELD,OFFSET=0,DATATYP=CHARACTER,DATALEN=40
      DFHCNV TYPE=FIELD,OFFSET=40,DATATYP=BINARY,DATALEN=4,     X
          LAST=YES
LABLN  DFHCNV TYPE=FINAL
*
*  EXAMPLE OF A USER-DEFINED CONVERSION TABLE EBCDIC to ASCII
EBTOAS DC   XL16'000102030405060708090A0B0C0D0E0F'
      DC   XL16'101112131415161718191A1B1C1D1E1F'
      DC   XL16'202122232425262728292A2B2C2D2E2F'
      DC   XL16'303132333435363738393A3B3C3D3E3F'
      DC   XL16'404142434445464748494A4B4C4D4E4F'
      DC   XL16'505152535455565758595A5B5C5D5E5F'
      DC   XL16'606162636465666768696A6B6C6D6E6F'
      DC   XL16'707172737475767778797A7B7C7D7E7F'
      DC   XL16'80C1C2C3C4C5C6C7C8C98A8B8C8D8E8F'
      DC   XL16'90D1D2D3D4D5D6D7D8D99A9B9C9D9E9F'
      DC   XL16'A0A1E2E3E4E5E6E7E8E9AAABACADAEAF'
      DC   XL16'B0B1B2B3B4B5B6B7B8B9BABBBCBDBEBF'
      DC   XL16'C0C1C2C3C4C5C6C7C8C9CACBCCDCECF'
      DC   XL16'D0D1D2D3D4D5D6D7D8D9DADBDCDDDEDF'
      DC   XL16'E0E1E2A3E4E5E6E7E8E9EAEBECEDEEEF'
      DC   XL16'F0F1F2F3F4F5F6F7F8F9FAFBFCFDFEFF'
*
*  EXAMPLE OF A USER-DEFINED CONVERSION TABLE ASCII to EBCDIC
*
ASTOEB DC   XL16'000102030405060708090A0B0C0D0E0F'
      DC   XL16'101112131415161718191A1B1C1D1E1F'
      DC   XL16'202122232425262728292A2B2C2D2E2F'
      DC   XL16'303132333435363738393A3B3C3D3E3F'
      DC   XL16'404142434445464748494A4B4C4D4E4F'
      DC   XL16'505152535455565758595A5B5C5D5E5F'
      DC   XL16'606162636465666768696A6B6C6D6E6F'
      DC   XL16'707172737475767778797A7B7C7D7E7F'
      DC   XL16'808182838485868788898A8B8C8D8E8F'
      DC   XL16'909192939495969798999A9B9C9D9E9F'
      DC   XL16'A0A1A2A3A4A5A6A7A8A9AAABACADAEAF'
      DC   XL16'B0B1B2B3B4B5B6B7B8B9BABBBCBDBEBF'
      DC   XL16'C0818283848586878889CACBCCDCECF'
      DC   XL16'D0919293949596979899DADBDCDDDEDF'
      DC   XL16'E0E1A2A3A4A5A6A7A8A9EAEBECEDEEEF'
      DC   XL16'F0F1F2F3F4F5F6F7F8F9FAFBFCFDFEFF'
      END  DFHCNVBA

```

Figure 22. SBCS user-defined conversion table

Assembling and link-editing the conversion programs

In MVS/XA or MVS/ESA, you can use either of the standard procedures DFHAUPL and DFHAUPLK to assemble the DFHCNV table. If your CICS product is CICS/VSE 2.3 or CICS Transaction Server for VSE/ESA, or if your CICS product can run with MVS/XA or MVS/ESA, you can optimize CICS virtual storage use by link-editing the DFHCNV table and the DFHUCNV program with a MODE statement specifying AMODE(31) and RMODE(ANY). The table and program are then loaded above the 16MB line if enough CICS storage is available. For a successful link-edit, before assembly:

- Insert the following statement in front of the DFHCNV source:

```
PUNCH ' PHASE DFHCNV,*'.
```

- Insert the following statement after the DFHCNV=FINAL instruction:

```
END DFHCNVBA
```

User-replaceable conversion program

This section describes the user-replaceable data conversion program DFHUCNV. For an explanation of why you might need to amend or replace DFHUCNV, refer to “User/CICS conversion” on page 56. If you find that the standard conversion supplied by DFHCCNV meets your needs, you don’t need to use DFHUCNV.

DFHUCNV is described in the following topics:

- “Input to DFHUCNV”:
This describes the parameter list (DFHUVNDS), which points to the data, the templates, the conversion table, and so on. It also describes the structure of the conversion and key templates created by your DFHCNV resource definition macros.
- “Supplied user-replaceable conversion program” on page 83:
This lists the commented version of DFHUCNV that is supplied with your CICS System/390 product.

With this information you can write your own conversion program, using the supplied program as a base.

Input to DFHUCNV

The first statement in the supplied version of DFHUCNV is a DFHCNV TYPE=DSECT macro, which generates DSECTs that describe the parameter list (see “Parameter list (DFHUVNDS)” on page 77) and the conversion template (see the general description of conversion and key templates on page 60 and the reference information on page 80).

DFHUCNV starts with a DFHCNV TYPE=DSECT in the following format:

	DFHCNV	TYPE=DSECT
--	--------	------------

The DFHCNV TYPE=DSECT macro generates the following:

- The DFHUNVDS DSECT, which maps the parameter list in the COMMAREA passed by DFHCCNV.

user-replaceable conversion program

- An assembler DSECT for field conversion records (these are the basic components of a template; see Figure 25 on page 80).
- Equates for resource types and field types.

Parameter list (DFHUVNDS)

Figure 23 shows the DFHUNVDS DSECT, which maps the parameter list passed to DFHUCNV in the COMMAREA. If a parameter is zero, no data is available. *If you do not create a conversion template for the resource, DFHUCNV is invoked, but only the following fields in the parameter list contain data:*

UNVRSTP
UNVRNMP
UNVDIRP
UNVOVLY

DFHUNVDS	DSECT		
UNVRSTP	DS	AL4	PTR-TO-RESOURCE TYPE
UNVRNMP	DS	AL4	PTR-TO-RESOURCE NAME
UNVDIRP	DS	AL4	PTR-TO-CONVERSION DIRECTIVE
CNVRQATE	EQU	X'02'	REQUEST ASCII TO EBCDIC
CNVRPETA	EQU	X'04'	RESPONSE EBCDIC TO ASCII
UNVDTMP	DS	AL4	PTR-TO-DATA CONV TEMPLATE
UNVDLNP	DS	AL4	PTR-TO-DATA TEMPLATE LENGTH
UNVKTMP	DS	AL4	PTR-TO-KEY CONV TEMPLATE
UNVKLNP	DS	AL4	PTR-TO-KEY TEMPLATE LENGTH
UNVATEP	DS	AL4	PTR-TO-ASCII/EBCDIC TRANS TABLE
UNVETAP	DS	AL4	PTR-TO-EBCDIC/ASCII TRANS TABLE
UNVATED	DS	AL4	PTR-TO-DBCS ASCII/EBCDIC TRANS TABLE
UNVETAD	DS	AL4	PTR-TO-DBCS EBCDIC/ASCII TRANS TABLE
UNVOVLY	DS	0H	OVERLAY SECTION
	ORG	UNVOVLY	TS REQUEST OVERLAY
UNVTSDP	DS	AL4	PTR-TO-TS DATA
UNVTSLNP	DS	AL4	PTR-TO-TS DATA LENGTH
	ORG	UNVOVLY	TD REQUEST OVERLAY
UNVTDDP	DS	AL4	PTR-TO-TD DATA
UNVTDLNP	DS	AL4	PTR-TO-TD DATA LENGTH
	ORG	UNVOVLY	IC REQUEST OVERLAY
UNVICDP	DS	AL4	PTR-TO-IC DATA
UNVICLNP	DS	AL4	PTR-TO-IC DATA LENGTH
	ORG	UNVOVLY	PC REQUEST OVERLAY
UNVPCDP	DS	AL4	PTR-TO-PC DATA
UNVPCLNP	DS	AL4	PTR-TO-PC DATA LENGTH
	ORG	UNVOVLY	FC REQUEST OVERLAY
UNVFCDP	DS	AL4	PTR-TO-FC DATA
UNVFCLNP	DS	AL4	PTR-TO-FC DATA LENGTH
UNVFCKP	DS	AL4	PTR-TO-FC KEY
UNVFCKLP	DS	AL4	PTR-TO-FC KEY LENGTH
	ORG	,	
UNVMRTNE	DS	A	PTR-TO-MBCS TRANSLATION ROUTINE

Figure 23. DFHUNVDS—DSECT that maps the parameter list passed to DFHUCNV

The following is a detailed description of the parameters:

UNVRSTP

Points to a one-byte resource type that indicates the resource being referenced by this request. The meanings of the resource types are defined in DSECT DFHCNVDS. The resource types are FC, IC, TS, TD, and PC.

user-replaceable conversion program

UNVRNMP

Points to an eight-character field containing the resource name, padded with blanks if necessary. These may be:

- For an FC request, an eight-byte file name
- For a TS request, an eight-byte TS queue name
- For a TD request, a four-byte TD queue name
- For an IC request, a four-byte transaction name
- For a PC request, an eight-byte program name.

UNVDIRP

Points to a one-byte field that shows what conversion is required:

- CNVRQATE (X'02') indicates a request needing ASCII to EBCDIC conversion.
- CNVRPETA (X'04') indicates a response needing EBCDIC to ASCII conversion.

UNVDTMP

Points to the start of the conversion template found by CICS to match this resource; zero if none is found.

UNVDLNP

Points to a half-word field that gives the length of the conversion template.

UNVKTMP (file control requests only)

Points to the start of the template found by CICS for the key part of the request. If UNVKTMP is zero, either there is no key template or the record is accessed by relative record number or relative byte address.

UNVKLNP

Points to a half-word field that gives the length of the key conversion template.

UNVATEP

Points to a 256-byte translation table used for ASCII to EBCDIC translation. You can use this in your DFHUCNV program.

UNVETAP

Points to a 256-byte translation table used for EBCDIC to ASCII translation. You can use this in your DFHUCNV program.

UNVATED

Points to an ASCII-to-EBCDIC translation table for DBCS data. You can use this in your DFHUCNV program.

UNVETAD

Points to an EBCDIC-to-ASCII translation table for DBCS data.

The overlay section depends on resource type:

TS requests:

UNVTSDP

Points to the start of the TS record being read or written.

UNVTSLNP

Points to a half-word field that gives the length of the TS record.

TD requests:

UNVTDDP

Points to the start of the TD record being read or written.

UNVTDLNP

Points to a half-word field that gives the length of the TD record.

IC requests:

UNVICDP

Points to the “from” area of an IC START request if one was supplied; if none, this is zero.

UNVICLNP

Points to a half-word field that gives the length of the “from” area.

PC requests:

UNVPCDP

Points to the start of the COMMAREA being supplied.

UNVPCLNP

Points to a half-word field that gives the length of the COMMAREA.

FC requests:

UNVFCDP

Points to the start of the file control record being read or written.

UNVFCLNP

Points to a half-word field that gives the length of the file control record.

UNVFCKP

Points to the start of the file control key being read or written.

UNVFCKLP

Points to a half-word field that gives the length of the key.

UNVMRTNE

Points to a translation routine that must be used for translations to or from an MBCS code page. The relevant client code pages are 954, 964, and 970.

The routine expects Register 1 to point to a structure defined by the DFHUNVM DSECT:

DFHUNVM DSECT			
UNVMTABP	DS	AL4	Set to value in UNVATED or UNVETAD
UNVMINP	DS	AL4	Address of source data
INVMINL	DS	FL4	Length of source data
UNVMOUTP	DS	AL4	Address of target buffer
UNVMOUTL	DS	FL4	Length of target buffer

user-replaceable conversion program

Conversion and key templates

In the COMMAREA, fields UNVDTMP and UNVDLNP point to the conversion template and its length. Fields UNVKTMP and UNVKLNP point to the key template and its length. Figure 24 illustrates the use and meaning of these fields.

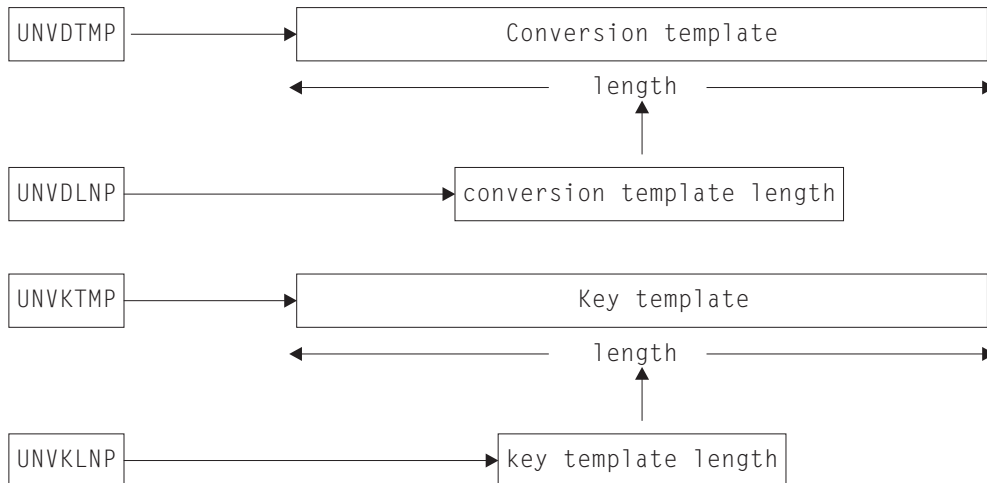


Figure 24. Parameter fields and the conversion templates

Each type of template consists of field conversion records, one for each field in the data record or key. Each field conversion record has the same layout, shown under "Field conversion records" on page 81, and mapped by a supplied DSECT, DFHCNVDS (see page 82). Figure 25 shows the relationship between a template, field conversion records, and DFHCNVDS. The figure shows DFHCNVDS overlaying the first field conversion record in a template for a data record or key with six fields.

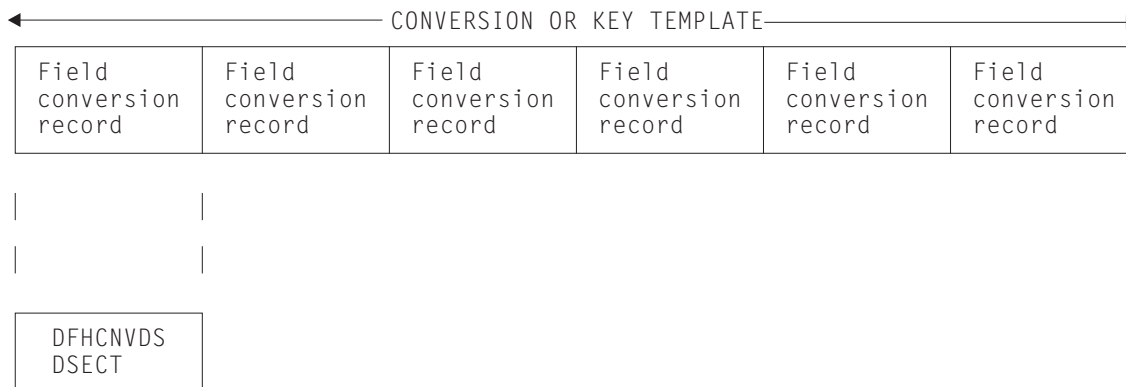


Figure 25. Field conversion records and a conversion or key template

Field conversion records

A field conversion record has the following layout:

Table 34. Layout of a field conversion record

CNVRLen	CNVRTYPE		CNVDATTY	CNVDATAO	CNVDATAL
Record length	Record type	Reserved	Data type	Data offset	Data length
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5-6	Byte 7-8

In Table 34, record length and type refer to the length and type of the field conversion record. The names in the top row are those used in the DSECT DFHCNVDS which maps field conversion records (see Figure 26 on page 82). A template has as many field conversion records as are necessary to describe all the fields in the data record or key.

For DFHUCNV, CNVRLen is always X'08' and CNVRTYPE is always X'04' (field). DFHUCNV must interpret CNVDATTY values in the range X'50' through X'80' according to user specifications, and apply the appropriate conversions. DFHUCNV should ignore fields with CNVDATTY values outside the range X'50' to X'80'.

EQUATEs in DFHCNVDS: Note that DFHCNVDS contains EQUATEs that are useful in your conversion program, as follows:

For resource type addressed by the parameter list:

CNVFC	FILE CONTROL
CNVTS	TEMPORARY STORAGE
CNVTD	TRANSIENT DATA
CNVIC	INTERVAL CONTROL
CNVPC	PROGRAM CONTROL

For field type in the template:

DTBIN	BINARY
DTPD	PACKED DECIMAL
DTCHAR	CHARACTER
DTMIX	MIXED CHARACTER
DTDBCS	DBCS CHARACTER
DTNUM	INTEL INTEGER

Two additional EQUATEs, DTUSRMIN and DTUSRMAX, define the limits of the range of data types (X'50' to X'80') reserved for user definition. Ensure that DFHUCNV can deal with any data type in this range that can be used in your installation.

The supplied DFHUCNV program contains examples of the use of CNVTS, DTUSRMIN, and DTUSRMAX (see "Supplied user-replaceable conversion program" on page 83).

user-replaceable conversion program

DFHCNVDS, DSECT for field conversion records:

```
DFHCNVDS DSECT
*
* PROVIDES A MAPPING OF THE FIELD CONVERSION RECORDS USED
* WHEN DECIDING WHETHER TO CONVERT USER DATA.
* A SET OF FIELD DEFINITIONS MAKE UP A TEMPLATE
*
CNVRLEN DS AL1 LENGTH OF THIS RECORD
CNVRTYPE DS XL1 TYPE OF RECORD
*
* EQUATES FOR RECORD TYPES
*
CNVTFLD EQU X'04' FIELD (ONLY VALID TYPE IN
* TEMPLATE)
CNVOVLY DS 0H
**
**
ORG CNVOVLY TYPE FIELD
DS XL1 RESERVED
CNVDATTY DS XL1 DATA TYPE
*
* EQUATES FOR DATA TYPES
*
DTBIN EQU X'01' BINARY
DTPD EQU X'02' PACKED DECIMAL
DTCHAR EQU X'03' CHARACTER
DTMIX EQU X'04' MIXED CHARACTER
DTDACS EQU X'05' DACS
DTNUM EQU X'06' NUMERIC
DTUSRMIN EQU X'50' MINIMUM USER DATA TYPE
DTUSRMAX EQU X'80' MAXIMUM USER DATA TYPE
*
CNVDATAO DS AL2 DATA OFFSET
CNVDATAL DS AL2 DATA LENGTH
**
*
* EQUATES FOR RESOURCE TYPES
*
CNVFC EQU X'01' FILE CONTROL
CNVTS EQU X'02' TEMP STORAGE
CNVTD EQU X'03' TRANS DATA
CNVIC EQU X'05' INTERVAL CONTROL
CNVPC EQU X'06' PROGRAM CONTROL
```

Figure 26. DFHCNVDS, DSECT that maps conversion/key templates passed to DFHUCNV

Supplied user-replaceable conversion program

Figure 27 lists the supplied version of DFHUCNV, which is written in assembler. This checks for a resource type of TS. If it finds one, it scans down the passed template looking for fields defined with a type in the user-data range. If any are present, DFHUCNV converts them as characters; you can rewrite the conversion code to your own requirements.

```

*   MODULE NAME = DFHUCNV
*   DESCRIPTIVE NAME = C.I.C.S./.....
**       CICS for OS/2 USER CONVERSION SAMPLE PROGRAM
*
*   TRANSACTION NAME = Cxxx
**       NOT A TRANSACTION
*
*           5665-403
*
*   STATUS = n.n.n
*
*   FUNCTION =
*       THIS IS A SAMPLE PROGRAM FOR USER DATA CONVERSION
*       IT IS INVOKED AS A RESULT OF A FUNCTION
*       SHIPPED REQUEST OR RESPONSE VIA THE LU2 REMOTE SERVER
*       OR LU6.2 HOST MIRROR PROGRAM. IT IS ACTUALLY CICS
*       LINKED FROM DFHCCNV TO ALLOW A USER PROGRAM TO
*       CONVERT DATA OF TYPE USERDATA AS DEFINED IN THE
*       CICS for OS/2 CONVERSION MACROS (DFHCNV).
*
*       THIS PROGRAM IS CALLED FOR EACH EXEC CICS REQUEST/RESPONSE
*       FOR WHICH DATA EXISTS FOR CONVERSION FROM ASCII TO EBCDIC.
*       IF A REQUEST DOES NOT CONTAIN ANY SUCH DATA, THIS PROGRAM
*       IS NOT INVOKED. THE PROGRAM IS INVOKED BEFORE THE CICS
*       CONVERSION PROGRAM (DFHCCNV) ATTEMPTS ANY CONVERSION
*       INBOUND FROM CICS for OS/2 (ASCII TO EBCDIC) OR OUTBOUND
*       FROM CICS for OS/2 (EBCDIC TO ASCII).
*
*       A COMMAREA IS PASSED WITHIN WHICH IS A SERIES OF POINTERS
*       TO INFORMATION THAT CAN BE USED BY THE PROGRAM TO
*       DETERMINE HOW TO CONVERT ANY RELEVANT DATA. THIS PROGRAM
*       SHOULD ONLY CONVERT DATA OF TYPE USERDATA AS INDICATED IN
*       THE CONVERSION TEMPLATES. ANY DATA OF TYPE CHARACTER
*       WILL BE CONVERTED BY THE CICS CONVERSION MODULE DFHCCNV.
*
*       SEE A LATER DESCRIPTION FOR WHAT THE SAMPLE DOES
*
*

```

Figure 27. DFHUCNV, user-replaceable conversion program for CICS on System/390—CICS for OS/2 link (Part 1 of 6)

user-replaceable conversion program

```
* NOTES :
*   DEPENDENCIES = S/370
*       IN A CICS MVS/XA ENVIRONMENT, THIS PROGRAM SHOULD BE
*       LINKED WITH RMODE(ANY) AND RMODE(31). ALL ADDRESSES SHOULD
*       BE TREATED AS 31 BIT.
*   RESTRICTIONS =
*       NONE
*   REGISTER CONVENTIONS =
*       STANDARD EXEC
*   PATCH LABEL = Via DFHPATCH Macro
*   MODULE TYPE = EXECUTABLE
*   PROCESSOR = ASSEMBLER
*   ATTRIBUTES = READ ONLY, SERIALY REUSABLE
*   ENTRY POINT = DFHUCNV
*
*   PURPOSE =
*       THIS IS THE ONLY ENTRY POINT FOR ALL FUNCTIONS
*
*   LINKAGE =
*       EXEC CICS LINK FROM DFHCCNV IS THE ONLY WAY THIS PROGRAM
*       IS INVOKED
*
*   INPUT =
*       THE PARAMETERS ARE PASSED USING A COMMAREA AND THE
*       DSECT DFHUNVDS DESCRIBES THE STRUCTURE OF THESE PARAMETERS
*       THIS DSECT IS INCLUDED IN THIS PROGRAM BY ISSUING THE
*       DFHUCNV TYPE=DSECT MACRO CALL.
*
*   OUTPUT =
*       NO SPECIFIC PARAMETERS ARE RETURNED, AS THE PURPOSE OF
*       THIS PROGRAM IS PERFORM CONVERSION ON USER DATA.
*
*   EXIT-NORMAL =
*       NORMAL RETURN IS VIA AN EXEC CICS RETURN
*
*   EXIT-ERROR =
**      SAME AS EXIT NORMAL
*
*-----*
*
*   EXTERNAL REFERENCES =
*       NONE
*
*   ROUTINES =
*       NONE
*
*   DATA AREAS =
*       NONE
```

Figure 27. DFHUCNV, user-replaceable conversion program for CICS on System/390—CICS for OS/2 link (Part 2 of 6)

user-replaceable conversion program

```
* CONTROL BLOCKS =
* THE 2 MAIN CONTROL BLOCKS REFERENCED ARE
* DFHUNVDS
* DESCRIBES THE PARAMETER LIST PASSED IN THE COMMAREA FROM
* DFHCCNV. IT IS BASICALLY CONSISTS OF A LIST OF POINTERS
* TO INFORMATION OF INTEREST TO THIS USER PROGRAM. THE FIRST
* PART OF THE LIST IS FIXED, AND THE SECOND PART CONSISTS
* OF OVERLAYS DEPENDING ON THE RESOURCE TYPE IN QUESTION.
* DFHCNVDS
* DESCRIBES THE STRUCTURE OF INDIVIDUAL FIELDS IN THE
* PASSED TEMPLATE.
*
* GLOBAL VARIABLES =
* NONE
*
* TABLES =
* DATA FROM THE DFHCNV TABLE IS USED BUT THE NECESSARY
* ADDRESSES ARE OBTAINED BY DFHCCNV AND PASSED IN THE
* COMMAREA
*
* MACROS =
* NONE
*
*-----*
*
* DESCRIPTION
* WHAT THIS SAMPLE DOES
*
* DFHUCNV EXECUTES AS AN EXEC CICS PROGRAM.
* DFHUCNV IS CALLED FOR ALL EXEC CICS REQUESTS/RESPONSES THAT
* HAVE RESULTED FROM A CICS for OS/2 FUNCTION SHIP REQUEST AND
* MAY REQUIRE CONVERSION OF USER DATA FROM ASCII TO EBCDIC OR
* VICE VERSA. THE FIRST THING THAT THE SAMPLE DOES IS TO
* OBTAIN ADDRESSABILITY TO THE PASSED COMMAREA, AND THEN
* CHECK THAT THE REQUEST IS A TEMPORARY STORAGE (TS) REQUEST.
* IF NOT WE JUST RETURN.
* NEXT WE CHECK IF DFHCCNV MANAGED TO LOCATE A CONVERSION
* TEMPLATE FOR THE RESOURCE (TS QUEUE) WITH THIS NAME.
* IF ONE WAS NOT FOUND (UNVDTMP IS ZERO) THIS MEANS THAT
* NO CONVERSION INFORMATION WAS PROVIDED (USING DFHCNV MACROS)
* FOR THIS RESOURCE. IN THIS CASE WE WILL NEVER BE ABLE TO
* LOCATE ANY USERDATA FIELDS, SO WE JUST RETURN.
* ASSUMING WE DID HAVE A TEMPLATE, WE NOW SCAN DOWN THE
* TEMPLATE USING THE SUPPLIED TEMPLATE PTR AND LENGTH. THE
* MAPPING OF THIS IS PROVIDED BY DFHCNVDS WHICH GIVES
* THE STRUCTURE OF THE CONSTITUENT FIELDS.
* EACH FIELD IS EXAMINED, AND WHEN ONE OF TYPE USERDATA
* IS FOUND WE DO SOME FURTHER CHECKS AS FOLLOWS.
```

Figure 27. DFHUCNV, user-replaceable conversion program for CICS on System/390—CICS for OS/2 link (Part 3 of 6)

user-replaceable conversion program

```
*      IT IS POSSIBLE THAT THE CONVERSION TEMPLATE HAS
*      DEFINITIONS FOR OFFSETS (AND OFFSETS PLUS LENGTHS) THAT ARE
*      GREATER THAN THE ACTUAL DATA ON THE EXEC REQUEST/RESPONSE.
*      OBVIOUSLY IT IS VERY IMPORTANT TO DETERMINE THE LESSER OF
*      THE ACTUAL DATA AND THE PARTICULAR TEMPLATE FIELD DEFINITION
*      TO ENSURE WE DO NOT PERFORM CONVERSION OFF THE END OF THE
*      REAL DATA. ONCE THESE CHECKS ARE DONE THE USERDATA FIELD
*      CAN BE TRANSLATED AS APPROPRIATE. PURELY AS AN EXAMPLE,
*      THE SAMPLE PROGRAM CONVERTS THE USERDATA FIELDS AS
*      CHARACTER, BUT IN A REAL PROGRAM, YOU WOULD PERFORM YOUR
*      OWN SPECIAL TESTING AND CONVERSION AT THIS POINT.
*      THIS LAST STEP IS REPEATED FOR EACH FIELD IN THE TEMPLATE
*      OF TYPE USERDATA, UNTIL THE END OF THE TEMPLATE IS FOUND,
*      AT WHICH TIME A RETURN IS MADE TO THE CALLER (DFHCCNV).
*
*      WHEN WRITING A VERSION OF THIS PROGRAM TO EXECUTE IN A
*      CICS MVS/XA ENVIRONMENT, YOU MUST BE PREPARED TO HANDLE
*      ALL ADDRESSES AS POSSIBLY 31 BIT, AS DFHCCNV AND THE
*      DFHCVN TABLE (CONTAINING THE TEMPLATES) WILL BE LOADED
*      ABOVE THE 16M LINE.
*
*-----*
*
* CHANGE ACTIVITY :
**
*      $MOD(DFHUCNV),COMP(ISC),PROD(CICS/MVS)
*
*      PN= REASON REL YMMDD HDXIII : REMARKS
*      P0= REASON REL YMMDD HDXIII : Implicit flag.
*      $01 Reserved for APAR fix
*      $02 Reserved for APAR fix
*      $03 Reserved for APAR fix
*      $D1 Reserved for DCR
*      $D2 Reserved for DCR
*      $D3 Reserved for DCR
*      $H1 Reserved for hardware support
*      $H2 Reserved for hardware support
*      $H3 Reserved for hardware support
*      $L0      210 880722 HD1HSS : CREATE DFHUCNV
*      $L1 Reserved for line item
*      $L2 Reserved for line item
*      $L3 Reserved for line item
*      $P1 Reserved for PTM
*      $P2 Reserved for PTM
*      $P3 Reserved for PTM
*
*****
```

Figure 27. DFHUCNV, user-replaceable conversion program for CICS on System/390—CICS for OS/2 link (Part 4 of 6)

user-replaceable conversion program

```

DFHUCNV DFHCNV TYPE=DSECT
CSECT
B IDBYP
DFHVM UCNV,ENTRY=DFHUCNV
IDBYP DS 0H
DFHREGS ,
OC EIBCALEN,EIBCALEN ANY COMMAREA ?
BZ RETURN NO, JUST RETURN
L R2,DFHEICAP
USING DFHUNVDS,R2 ADDRESSABILITY TO COMMAREA
L R10,UNVRSTP ADDRESS THE RESOURCE TYPE
CLI 0(R10),CNVTS IS IT A TEMPORARY STORAGE TYPE?
BNE RETURN NO, JUST RETURN
ICM R10,B'1111',UNVDTMP IS THERE A CONVERSION TEMPLATE ?
BZ RETURN NO, JUST RETURN
USING DFHCNVDS,R10 ADDRESSABILITY TO CONVERSION RECS
L R4,UNVDLNP
SR R5,R5
LH R5,0(0,R4) GET TOTAL TEMPLATE LENGTH
AR R5,R10 END OF TEMPLATE
PROCESS DS 0H
CR R10,R5 HAVE WE REACHED END OF TEMPLATE?
BNL RETURN YES
CLI CNVRTYPE,CNVTFD DOUBLE CHECK ITS A FIELD TYPE REC
BNE RETURN NO, BETTER RETURN
CLI CNVDATTY,DTUSRMIN IN THE USER RANGE ?
BL NEXTREC NO, TOO LOW
CLI CNVDATTY,DTUSRMAX IN THE USER RANGE ?
BH NEXTREC NO, TOO HIGH
L R4,UNVDIRP CHECK THE TYPE OF CONVERSION
CLI 0(R4),CNVRQATE ASCII TO EBCDIC REQUEST
BNE TRYEBC NO...
L R6,UNVATEP YES, ADDRESS THE RELEVANT TABLE
B CONT1
TRYEBC DS 0H MUST BE EBCDIC TO ASCII
L R6,UNVETAP ADDRESS THE RELEVANT TABLE
CONT1 DS 0H
*

```

Figure 27. DFHUCNV, user-replaceable conversion program for CICS on System/390—CICS for OS/2 link (Part 5 of 6)

user-replaceable conversion program

```

*      GET LOWER VALUE OF ACTUAL LENGTH AND POTENTIAL LENGTH
*      INTO R4
L      R4,UNVTSLNP
LH     R4,0(0,R4)           PICK UP ACTUAL TS DATA LENGTH
LTR    R4,R4               JUST CHECK ITS POSITIVE
BNP    RETURN              IF NOT RETURN
LH     R7,CNVDATAO         GET THE OFFSET FROM THE TEMPLATE
CR     R7,R4               IS THE OFFSET PAST THE DATA
BNL    NEXTREC            YES, TRY THE NEXT RECORD
AH     R7,CNVDATAL         ADD IN THE LENGTH(TEMPLATE)
CR     R7,R4               COMPARE OFFSET+LEN WITH REAL DATA
BH     LENOK
LR     R4,R7
LENOK  DS      0H
*      HERE R4 SHOULD BE THE SMALLER OF THE 2 LENGTHS
*      NOW CALCULATE THE REAL LENGTH FOR CONVERSION
SH     R4,CNVDATAO         SUBTRACT THE OFFSET
L      R7,UNVTSDP          ADDRESS ACTUAL DATA
AH     R7,CNVDATAO         ....PLUS OFFSET
*
*      R7 POINTS AT THE START OF WHERE WE TRANSLATE AND R4
*      INDICATES THE LENGTH (ENSURING WE DON'T GO FURTHER THAN THE
*      ACTUAL DATA)
TRANSMOR DS      0H
CH     R4,=H'256'          AT LEAST 256 BYTES TO DO
BL     TRREST              NO
TR     0(256,R7),0(R6)     TRANSLATE 256 BYTES
SH     R4,=H'256'          DECREMENT THE COUNT
AH     R7,=H'256'          INCREMENT THE POINTER
B      TRANSMOR            DO SOME MORE
TRREST DS      0H
LTR    R4,R4               ANY LEFT TO DO ?
BNP    DONETR              NO
SH     R4,=H'1'            DECREMENT THE COUNTER FOR THE TR
EX     R4,TRNSLT
B      DONETR
TRNSLT TR     0(0,R7),0(R6)
DONETR DS      0H          ALL DATA TRANSLATED
NEXTREC DS      0H
SR     R4,R4
IC     R4,CNVRLLEN         GET LENGTH OF THIS RECORD
AR     R10,R4              AND ADDRESS THE NEXT ONE
B      PROCESS
RETURN DS      0H
EXEC   CICS RETURN
DFHPATCH
END    DFHUCNV

```

Figure 27. DFHUCNV, user-replaceable conversion program for CICS on System/390—CICS for OS/2 link (Part 6 of 6)

Study the supplied version of DFHUCNV and its introductory comments. This will enable you to write your own conversion program. If you are running in an XA environment, your program must be able to handle 31-bit addresses.

Part 2. Server Support for CICS Clients

This part of the book describes how to set up a CICS on System/390 system to act as a server to the CICS Clients workstation products.

Important

- This part of the book describes the full-function support for CICS Clients provided by:
 - CICS Transaction Server for System/390
 - CICS Transaction Server for VSE/ESA
 - CICS/ESA Version 4.1.

The support for Clients provided by these products is identical, *except that*:

- In CICS Transaction Server for VSE/ESA and CICS/ESA Version 4.1, the autoinstall user program is not called for autoinstall of Client virtual terminals.
- CICS Transaction Server for VSE/ESA does not support:
 - The Resource Access Control Facility (RACF)
 - TCP/IP connections to Clients.
- Because of platform-specific variations, CICS/VSE Version 2.3's support for Clients is described in a separate manual—*CICS/VSE 2.3 Server Support for CICS Clients*, SC33-1712.
- Earlier versions of CICS System/390 products provide limited support for CICS Clients, as described on page 91.¹²

12. They support the Client External Call Interface (ECI), but not the External Presentation Interface (EPI), nor the Terminal Emulation function.

Chapter 7. Introduction to CICS Clients

CICS Clients are a family of workstation products that provide a standard set of functions for **client/server** computing.

What is a CICS Client?

A CICS Client is not a full-function CICS system. But it contains code to enable it to access the services of CICS systems. (*Services* mean things like transactions and programs.) CICS systems to which Clients are connected are known as CICS servers.

Each CICS Client is designed to run on a particular operating system. Each can attach to a common range of CICS systems, using a variety of protocols. CICS Clients thus allow users to access resources owned by CICS server-systems, from a variety of operating environments.

There is a CICS Client for each of the following operating systems:

- AIX (Version 3 Client only).
- DOS (Version 2 Client only)
- Microsoft Windows 3.1 (Version 2 Client only)
- Microsoft Windows NT
- Microsoft Windows 95
- Microsoft Windows 98 (Version 3 Client only)
- OS/2
- Solaris (Version 3 Client only).

Each Client can attach to any or all of the following CICS servers:

- CICS/400 Version 3.1
- CICS/ESA Version 3.3 *
- CICS/ESA Version 4.1
- CICS for OS/2
- CICS for Windows NT
- CICS on Open Systems
- CICS Transaction Server for OS/390
- CICS Transaction Server for VSE/ESA
- CICS/VSE Version 2.3

Note: CICS/ESA Version 3.3 offers limited server support. It supports the client External Call Interface (ECI), but not the External Presentation Interface (EPI), nor the Terminal Emulation function. All the Client functions are described in the next section.

What functions do CICS Clients provide?

CICS Clients provide a standard set of functions for client/server computing. This section gives an overview of the most important functions; it is not meant to be exhaustive.

The External Call Interface

The External Call Interface (ECI) is an application programming interface (API) that allows a non-CICS program running on a Client to call a CICS program located on a CICS server. This enables the Client to make use of existing server routines that could be used, for example, to make enquiries on a database.

Introduction to Clients

The Client program can make the following types of call to a CICS server:

- Program-link calls, which can be **synchronous** (that is, the calling program waits for a response from the linked-to program), or **asynchronous** (that is, the two programs continue to execute independently).
- Calls to retrieve a response from a previous asynchronous call.
- Calls that return a value indicating the status of the CICS system. This allows an application to test for availability of the CICS server or to monitor it by waiting for a change in its status.

Dynamic routing of ECI calls

CICS Transaction Server for OS/390 Release 3 allows you to route ECI calls dynamically. In CICS TS 390 Release 3, if a program defined as DYNAMIC is the subject of an ECI program-link call from a CICS Client, the dynamic routing program is invoked, and can select a remote region on which the server program is to execute.

This means that CICS Clients can benefit from the workload balancing capabilities of CICS TS 390 Release 3.

For definitive information about how to route program-link requests dynamically, see the *CICS Transaction Server for OS/390 Release 3 Intercommunication Guide*.

The External Presentation Interface

The External Presentation Interface (EPI) is an API that allows a non-CICS Client program to appear to a CICS server as one or more standard 3270 terminals. This enables the Client to access, for example, CICS on System/390 transactions written for 3270 terminals, without needing to change the System/390 code.

The Client program can start CICS transactions and send and receive standard 3270 datastreams to and from the transactions. It can present the 3270 data to the user by emulating a 3270 terminal, or by means of a graphical user interface such as Presentation Manager[®] (OS/2 Clients) or Windows (Windows Clients).

The EPI consists of a set of calls that can be made from a Client program.

Depending on the Client version and platform, the Client program can be written in any of a number of languages, including:

- C
- C++
- COBOL
- PL/I
- REXX (on OS/2)
- Visual Basic (on Windows NT).

The EPI calls are provided in a library that is linked to the application. Among the functions available are calls to:

- Initialize the EPI.
- Terminate the EPI.
- Obtain a list of CICS servers to which a “terminal” may attach.
- Attach a virtual terminal.
- Detach a virtual terminal.
- Start a transaction for a “terminal”.
- Send data from a “terminal” to a transaction.

- Obtain details of an “event” that has occurred for a terminal. An example of an event is when the transaction is expecting a reply from the terminal.
- Obtain detailed error information for the last error that occurred for a terminal.

Terminal emulation

CICS Clients can run 3270 terminal emulators. A Client terminal emulator transmits or receives standard CICS transaction routing flows to or from a CICS server. This allows a user to interact with the server, and run transactions, as if the Client were a locally-attached 3270 terminal.

It is possible to run multiple terminal emulators on a single Client. The emulators can be connected to the same CICS server, or to different servers. In the former case, each instance of the emulator requires a unique name and represents a separate terminal to the server.

Users can customize the colors and keyboard mapping of their emulators.

What protocols are supported?

APPC

CICS on System/390-CICS Client applications must use the LU 6.2 (APPC) communication protocol. Single- or parallel-session connections can be used.

On CICS on System/390 systems, you can use autoinstall to define APPC connections to Clients dynamically, on their first use. On CICS/ESA 4.1, CICS Transaction Server for OS/390, and CICS Transaction Server for VSE/ESA, you can autoinstall both single-session and parallel-session APPC connections. On all other CICS on System/390 products, only single-session APPC connections can be autoinstalled. Alternatively, you can use the RDO CEDA DEFINE and INSTALL commands to define static connections to Clients.

APPC links to CICS Clients support data synchronization levels (sync levels) 0 and 1.

TCP/IP support

CICS/ESA 4.1 and CICS Transaction Server for OS/390 can communicate with some types of CICS Client over a Transport Control Protocol/Internet Protocol (TCP/IP) network. The Clients supported are those for:

- Microsoft Windows 95 and 98
- Microsoft Windows NT
- OS/2.

Communication using TCP/IP requires IBM TCP62 support to be installed on the Client workstation. TCP62 is a protocol mapper that enables partner APPC applications to communicate using TCP/IP.

For information about how to set up a TCP/IP connection to a CICS Client, see “Using TCP/IP” on page 96.

Benefits of Client support

CICS on System/390’s support for CICS Clients has the following benefits:

Introduction to Clients

- Company managers can migrate to client/server solutions quickly, in a staged manner, by:
 1. Using the Client 3270 emulator to run existing CICS on System/390 3270 applications.
 2. Using the EPI to add graphical user interface (GUI) front-ends to existing CICS on System/390 3270 applications. This should increase end-user satisfaction and productivity.
 3. Using the ECI to develop new client/server applications in which the display and processing logic is appropriately split between the client and the server.

Different applications at different stages in the above migration scenario can coexist on the same client.

- The handling of code page translation between EBCDIC (used on CICS on System/390) and ASCII (used on workstations) is simplified.
- CICS on System/390's support for autoinstall of connections and virtual terminals means that systems developers can design large CICS client/server networks without being concerned about the problems of installing and maintaining definitions of all the clients in all the servers; or of installing and maintaining definitions of all the different virtual 3270 terminals in all the clients.

It also means that system administrators do not have to extend the resource definitions of server systems when a new client is added to the network.

- Users of CICS on System/390 3270 applications benefit from the improved end-user interfaces available to intelligent workstations.

Further information about Clients

For programming information about the Client ECI and EPI APIs, see the *CICS Family: Client/Server Programming* manual, SC33-1435. For information about setting up the Client side of a CICS Client-CICS server link, see the *CICS Clients: Administration* manual. The remainder of this document deals with setting up and administering the CICS on System/390 side of the Client-server link.

Chapter 8. Installation of server support for Clients

Important

This chapter applies only to:

- CICS Transaction Server for OS/390
- CICS Transaction Server for VSE/ESA
- CICS/ESA Version 4.1.

It is intended to be read in conjunction with the *Resource Definition Guide* for your CICS on System/390 system, and with “Chapter 3. Resource definition” on page 17.

The following is a checklist of the things you must do to install CICS on System/390 support for CICS Clients:

- Install the supplied resource definition group, DFHCLNT—see “DFHCLNT resource group”
- Define the CSCC transient data queue—see “CSCC transient data queue”
- Install APPC connections to the Clients—see “Connections to Clients”
- Install some Client virtual terminals—see “Client virtual terminals” on page 98
- Specify the level of security to be used for Client-CICS on System/390 links—see “Security” on page 106
- Specify the code pages to be used for data conversion—see “Chapter 9. Data conversion for Clients” on page 109.

DFHCLNT resource group

You must install the CICS-supplied resource definition group, DFHCLNT, which includes definitions of the CICS on System/390 internal transactions, CCIN and CTIN, and of the programs they use. CCIN allows Clients to pass information to the server, such as the client code page to be used for data conversion between ASCII and EBCDIC. CTIN is required to install remote definitions of Client virtual terminals.

CSCC transient data queue

Messages relating to Client support are written to the CSCC transient data queue. You must define CSCC to CICS. There is a sample definition in the supplied resource definition group, DFHDCTG. The sample defines CSCC as an indirect extrapartition destination, pointing to CSSL.

For further information about transient data queues, see your CICS on System/390 *Resource Definition Guide*.

Connections to Clients

You must install APPC connections to the Clients. The connections can be single- or parallel-session links.

Static APPC definitions

You can use CEDA DEFINE and INSTALL commands to create static definitions.

installation of Client support

For information about defining APPC connections to non-System/390 systems, see “Chapter 3. Resource definition” on page 17.

Autoinstalled APPC connections

The preferred method of installing connections to Clients is to use autoinstall. If you use autoinstall, you must create some suitable CONNECTION and SESSIONS template definitions, if these do not already exist.

For information about autoinstall and defining templates, see your CICS on System/390 *Resource Definition Guide*. For information about customizing your autoinstall user program to handle APPC connections, see your CICS on System/390 *Customization Guide*.

A connection is autoinstalled on CICS on System/390 when the Client initiates communication. (This could be, for example, when the end-user issues a CICSCLI /s=*servername* command to connect to the CICS on System/390 server, a CICSTERM /s=*servername* command to start a 3270 emulator session, or when an ECI or EPI program is started on the Client.)

Using TCP/IP

CICS/ESA 4.1 and CICS Transaction Server for OS/390 can communicate with some types of CICS Client over a TCP/IP network—see “TCP/IP support” on page 93.

System/390 software required for TCP/IP support

Support for TCP/IP links to Clients requires the AnyNet[®] feature of VTAM (which in earlier VTAM releases is known as the “Multi-Protocol Transport Feature”). The other System/390 software you require depends on the level of VTAM you are running. Table 35 shows the acceptable software combinations.

Table 35. System/390 software required for TCP/IP support. Each row lists an acceptable software combination.

<ul style="list-style-type: none">• VTAM Version 3 Release 4.2• Multi-Protocol Transport Feature of VTAM Version 3 Release 4.2• C/370 Version 2 or later• MVS/ESA SP Version 3 Release 1.3 or later• TCP/IP Version 2 Release 2.1 or later
<ul style="list-style-type: none">• VTAM Version 4 Release 2• AnyNet Feature of VTAM Version 4 Release 2• C/370 Version 2 or later• MVS/ESA SP Version 3 Release 1.3 or later• TCP/IP Version 2 Release 2.1 or later
<ul style="list-style-type: none">• VTAM Version 4 Release 3• AnyNet Feature of VTAM Version 4 Release 3• Language Environment[®] for MVS Version 1 or later• MVS/ESA SP Version 3 Release 1.3 or later• TCP/IP Version 2 Release 2.1 or later
<ul style="list-style-type: none">• VTAM Version 4 Release 4 (also shipped as part of OS/390 Release 3 and above)• AnyNet (integrated into VTAM Version 4 Release 4)• Language Environment for MVS Version 1 Release 3 or later• MVS/ESA SP Version 4 Release 3 or later• TCP/IP Version 3 Release 1 or later

Enabling TCP/IP support

Enabling CICS on System/390 to communicate with a CICS Client using TCP/IP requires actions on OS/390, CICS, VTAM, and the Client workstation.

On OS/390: On OS/390, you must:

1. Install a TCP major node. For example:

```
ABHTCP  VBUILD  TYPE=TCP,
          CONTIMER=25,
          DGTIMER=40,
          DNSUFFIX=HURSLEY.IBM.COM,
          EXTIMER=5,
          IATIMER=60,
          PORT=397,
          TCB=10,
          TCPIPJOB=TCPIP
ABHIGRP  GROUP  ISTATUS=ACTIVE
ABH1LINE LINE  ISTATUS=ACTIVE
ABH1PU   PU     ISTATUS=ACTIVE
```

This defines the AnyNet interface between TCP/IP and VTAM. For further information about how to do this, see the *Guide to SNA over TCP/IP* manual, SC31-6527.

2. Install a CDRSC major node. For example:

```
AP23ACDS  VBUILD  TYPE=CDRSC
TCPCDGRP  GROUP
IYCNT999  CDRSC  ALSLIST=ABH1PU,MODETAB=MTCICS
```

This defines the remote Client device and instructs VTAM to route any session requests through the TCP/IP Physical Unit (ALSLIST).

3. Check that the Physical Unit (PU) for the AnyNet interface is active. If it is in a PCON2 state, check that VTAM has a StepLib to:

```
DSN=PP.ADLE370.OS390R2.SCEERUN
```

This module should be present by default.

On CICS and VTAM: On CICS, you must:

1. Define an APPC connection to the Client workstation. (The connection can be statically defined, or autoinstalled.)
 - On the MODENAME option of the SESSIONS definition, specify the same modename as that specified in the Client INI file. (The default modename in the Client INI file is TCP62.)
 - On the MAXIMUM option of the SESSIONS definition, specify the second value as zero—that is, that CICS is to have no contention winners. For example, MAXIMUM(8,0) means that the modeset is to support eight sessions, and that CICS is to be the contention loser in each case.
2. Add an entry to the VTAM logon mode (LOGMODE) table for the modename specified on the SESSIONS definition. This entry specifies the class of service required for the group of sessions.

On the Client workstation: On the Client workstation, you must install TCP62 support and configure the Client initialization (INI) file. For information about how to do this, see the *CICS Clients: Administration* manual.

Client virtual terminals

If the EPI or the Client terminal emulator is to be used, the Client virtual terminals must be installed on CICS on System/390. Client virtual terminals are defined to CICS on System/390 as remote 3270 terminals.

Note the following:

- CICS Clients do not ship remote terminal definitions to CICS on System/390 for use as virtual terminals. Instead, you define the virtual terminals to CICS on System/390 as remote 3270 devices, using either static definitions or autoinstall. If you use autoinstall, an autoinstall model is used as the basis for the virtual terminal.
- VTAM definitions are not required for Client virtual terminals.
- Although CICS Clients do not ship definitions of virtual terminals to CICS on System/390, once installed virtual terminals can be shipped by CICS on System/390 to connected CICS systems. Therefore transaction routing can be used from a Client virtual terminal.

Using static definitions

You can use CEDA DEFINE and INSTALL commands to create static definitions of Client terminals. When you use static definitions:

- Client EPI programs reserve particular definitions for their use by quoting the TERMIDs on the **NetName** parameter of **CICS_EpiAddTerminal** calls (see “How CICS installs Client terminals” on page 102). For information about the **CICS_EpiAddTerminal** function, see the *CICS Family: Client/Server Programming* manual.
- A workstation user chooses a particular definition for an emulator session by quoting the TERMID on the **/n** (NetName) parameter of the **cicsterm** command used to start the emulator. For information about the **cicsterm** command, see the *CICS Clients: Administration* manual.

Figure 28 on page 99 and Figure 29 on page 100 are example TERMINAL and TYPETERM definitions for Client terminals.

Example Client terminal definition

Note that:

- On the TERMINAL definition:
 - TERMINAL is the name by which the terminal is to be known to the Client—that is, the name that the Client will quote on the **NetName** parameter of **CICS_EpiAddTerminal** calls, or on the **/n** parameter of a **cicsterm** command.

Note that the terminal names passed by Clients are case-sensitive. That is, if a Client passes a name that contains lowercase letters, CICS does not translate it into uppercase. CICS searches for a remote terminal definition whose TERMINAL name is the same combination of mixed or lowercase letters.
 - NETNAME should be allowed to take its default value.
 - REMOTESYSTEM must be the name of the connection to the Client.
 - Specify REMOTENAME if you want to use an alias terminal identifier by which the terminal will be known to CICS. (Do not specify REMOTENAME if the TERMINAL definition is to be used as an autoinstall model.)
 - USERID must be left blank—you cannot use preset security with Client virtual terminals.

installation of Client support

- On the TYPETERM definition, value 2 of the CGCSGID option should specify the code page to be used by the CICS on System/390 server for data conversion. You can use value 1 to specify the server character set. Alternatively, by leaving value 1 set to '00000', you can allow CICS to choose a default character set, deduced from the server code page—see page 110.
- On both definitions, some inessential options have been omitted. You can allow these to take their default values.

```

DEFINE TERMINAL(VT12) GROUP(CLEPIVT)
OVERTYPE TO MODIFY
CEDA DEFine
TERminal    ==> VT12
GRoup       ==> CLEPIVT
AUTINSTModl ==> No           No|Yes|Only
AUTINSTName ==>
TERMINAL IDENTIFIERS
TYpeterm    ==> CLIVT
NEtname     ==>
CONSOLE     ==> No           No|0-99
REMOTESystem ==> CLIA
REMOTESYsnet ==>
REMOTEName  ==>
ASSOCIATED PRINTERS
PRINTERCopy ==> No           No|Yes
ALTPRINTCopy ==> No           No|Yes
PIPELINE PROPERTIES
TAsklimit   ==> No           No|1-32767
OPERATOR DEFAULTS
OPERPID     ==>
OPERPriority ==> 000         0-255
OPERRs1     ==> 0           0-24,...
OPERSecurity ==> 1          1-64,...
PRESET SECURITY
Userid      ==>
TERMINAL USAGES
TERmpriority ==> 000         0-255
Inservice   ==> Yes         Yes|No
SESSION SECURITY
ATtatchsec  ==> Local       Local|Identify|Verify|
                                     Persistent|Mixidpe
BINDPassword ==>
BINDSecurity ==> No         PASSWORD NOT SPECIFIED
                                     No|Yes

```

Figure 28. Example TERMINAL definition for statically defined Client virtual terminal

installation of Client support

Here is an example TYPETERM definition to accompany the preceding TERMINAL definition. It is based on the CICS-supplied TYPETERM, DFHLU2.

```
DEFINE TYPETERM(CLIVT) GROUP(CLEPIVT)
OVERTYPE TO MODIFY
CEDA DEFINE
  TYPeterm    ==> CLIVT
  Group       ==> CLEPIVT
RESOURCE TYPE
DEVIce       ==> LUTYPE2
TERmodel     ==> 2
SESSiontype  ==>
LDcllist     ==>
SHippable    ==> Yes                No|Yes
MAPPING PROPERTIES
PAGesize     ==> 024 , 080          0-999
ALTPage      ==> 000 , 000          0-999
ALTSuffix    ==>
FMhparm      ==> No                No|Yes
OBOperid     ==> No                No|Yes
PAGING PROPERTIES
AUTOPage     ==> No                No|Yes
DEVICE PROPERTIES
DEFscreen    ==> 024 , 080          0-999
AUDiblealarm ==> Yes                No|Yes
EXTendeddds  ==> Yes                No|Yes
Query        ==> A11                No|Co1d|A11
SOSi         ==> No                No|Yes
BACKtrans    ==> No                No|Yes
CGcsgid      ==> 00000 , 00037      0-65535
SESSION PROPERTIES
AScii        ==> No                No|7|8
SENdsizes    ==> 01536              0-30720
RECEivesize  ==> 00256              0-30720
BRacket      ==> Yes                Yes|No
LOGMode      ==>
DIAGNOSTIC DISPLAY
ERRLastline  ==> Yes                No|Yes
ERRIntensify ==> Yes                No|Yes
ERRColor     ==> No                No|Blue|Red|Pink|Green|
Turquoise|Yellow|NEutral
ERRHilight   ==> No                No|Blink|Reverse|Underline
OPERATIONAL PROPERTIES
AUTOConnect  ==> No                No|Yes|A11
ATi          ==> Yes                No|Yes
CREatesess   ==> No                No|Yes
RELreq       ==> Yes                No|Yes
DIScreq      ==> Yes                Yes|No
MESSAGE RECEIVING PROPERTIES
ROUTedmsgs   ==> A11                A11|None|Specific
LOGOnmsg     ==> Yes                No|Yes
APPLICATION FEATURES
BUildchain   ==> Yes                No|Yes
USerarealen  ==> 000                0-255
IOarealen    ==> 00256 , 04000      0-32767
UCtran       ==> Yes                No|Yes|Tranid
```

Figure 29. Example TYPETERM definition for statically defined Client virtual terminal

Using autoinstall

The preferred method of defining Client terminals is to use autoinstall, because it is more convenient when many terminals connect to a single server.

Autoinstall models

If you use autoinstall, you may need to create some model terminal definitions (that is, TERMINAL-TYPETERM pairs) for use with Client terminals, unless you have some existing ones that are suitable. The CICS-supplied autoinstall model DFHLU2

may be appropriate (it is the default if the Client does not specify a model name). Value 2 of the CGCSGID option of the TYPETERM definitions should specify the code page to be used by the CICS on System/390 server for data conversion.

Note: Because Client virtual terminals are not “seen” by VTAM, there is no need to create matching entries for the autoinstall models in the VTAM LOGMODE table.

The autoinstall model used to install a virtual terminal is determined using the following sequence:

1. **For EPI programs:** From the **DevType** parameter of the **CICS_EpiAddTerminal** function, if specified by the Client EPI program. (For details of EPI calls, see the *CICS Family: Client/Server Programming* manual.)
For the Client terminal emulator: From the **/m** (Modelname) parameter of the **cicsterm** command used to start the emulator, if specified by the workstation user. (For details of the **cicsterm** command, see the *CICS Clients: Administration* manual.)
2. The CICS-supplied autoinstall model, DFHLU2.

The autoinstall user program cannot choose a different autoinstall model.

Terminal identifiers

The terminal identifier (TERMID) passed to the CICS autoinstall function at install of a virtual terminal is determined using the following sequence:

1. **For EPI programs:** From the **NetName** parameter of the **CICS_EpiAddTerminal** function, if specified by the Client EPI program.
For the Client terminal emulator: From the **/n** parameter of the **cicsterm** command used to start the emulator, if specified by the workstation user.
Note that the terminal names passed by Clients are case-sensitive—that is, if they contain lowercase letters, these are not translated into uppercase by CICS.
2. A name generated automatically by CICS. TERMIDs generated by CICS for autoinstalled Client terminals consist of a 1-character prefix and a 3-character suffix. The default prefix is '\'. The suffix can have the values 'AAA' through '999'. That is, each character in the suffix can have the value 'A' through 'Z' or '0' through '9'. The first suffix generated by CICS has the value 'AAA'. This is followed by 'AAB', 'AAC', ... 'AAZ', 'AA0', 'AA1', and so on, up to '999'. Each time a Client virtual terminal is autoinstalled, CICS generates a 3-character suffix that it has not recorded as being in use.

You can use the VTPREFIX system initialization parameter to override the default prefix assigned to CICS-generated TERMIDs. Use VTPREFIX to specify a different prefix, reserved for virtual terminals, on each TOR on which Client virtual terminals are to be installed. This ensures that the TERMIDs of Client terminals autoinstalled on each system are unique in your transaction routing network. This in turn prevents the conflicts that could occur if two or more terminal-owning regions ship definitions of Client virtual terminals to the same application-owning region.

If such a naming conflict does occur—that is, if a Client virtual terminal is shipped to an AOR on which a remote terminal of the same name is already installed—the autoinstall user program is invoked in the AOR. Your user program can resolve the conflict by allocating an alias terminal identifier to the shipped definition. (For details of writing an autoinstall user program to install shipped definitions, see your CICS on System/390 *Customization Guide*.)

installation of Client support

Notes:

- a. You can specify VTPREFIX as a system initialization override, or by coding an entry in the system initialization table (SIT).
- b. When specifying a prefix, ensure that TERMIDs generated by CICS for Client terminals do not conflict with those generated by your autoinstall user program for user terminals, or with the names of any other terminals or connections.
- c. Client terminal definitions are not recovered after a restart. Immediately after a restart, no Client terminals are in use, and so when CICS generates TERMIDs it begins again at the start of its sequence. This means that CICS does *not* always generate the same TERMID for any given Client terminal. This in turn means that server applications cannot assume that a particular CICS-generated TERMID always equates to a particular Client terminal.

CICS Transaction Server for OS/390 only

If your server programs do make this assumption, you can use your autoinstall user program to allocate alias TERMIDs, by which the virtual terminals will be known to CICS, in a consistent manner. For further details, see “EPI server programs” on page 113.

For definitive information about the VTPREFIX system initialization parameter, see your CICS on System/390 *System Definition Guide*.

CICS Transaction Server for OS/390 only

The autoinstall user program:

For brevity, the TERMID specified by the Client or the CICS-generated “VTPREFIX” name is referred to as the **supplied name**. The Client always knows the virtual terminal by the supplied name. However, your autoinstall user program can allocate an alias, by which the virtual terminal will be known to CICS.

One reason for using your autoinstall program to assign aliases to Client terminals might be to ensure that particular identifiers relate consistently to particular Client terminals. Whether this consistency is required depends on how your server applications are coded—see “EPI server programs” on page 113.

Your autoinstall user program cannot change the autoinstall model.

For further information about writing a user program to control the installation of Client virtual terminals, see your CICS on System/390 *Customization Guide*.

How CICS installs Client terminals

When an EPI program issues a **CICS_EpiAddTerminal** call, or the workstation user uses the **cicsterm** command to start an emulator session, the Client invokes the CTIN transaction to install or reserve the terminal definition on CICS on System/390.

Table 36 on page 103 relates the parameters passed by the Client to the way in which CICS installs the Client terminal.

Table 36. How CICS installs definitions of Client virtual terminals

CICS_EpiAddTerminal parameters or cicsterm specified		CICS actions			
NetName	DevType or Model-name	Static definition exists?	Result	TERMINID	Autoinstall model
Yes	No	Yes	Use static definition ¹³	Client-specified	-
Yes	No	No	Call fails	-	-
Yes	Yes	No	Autoinstall	Client-supplied ¹⁴	Client-specified
Yes	Yes	Yes	Use static definition ¹³	-	-
No	No	-	Autoinstall	CICS-supplied ¹⁴	DFHLU2
No	Yes	-	Autoinstall	CICS-supplied ¹⁴	Client-specified

Note: Definitions of Client virtual terminals are *not* deleted by the CICS on System/390 timeout delete mechanism that operates on shipped terminal definitions. ¹⁵

Defining a Client-attached printer

This section tells you how to define a printer that is locally attached to a Client workstation. As when defining any Client virtual terminal, you can either create a static definition, or use autoinstall. DFHLU3 is a suitable autoinstall model for a Client-attached printer.

Figure 30 on page 104 and Figure 31 on page 105 show example TERMINAL and TYPETERM definitions for a Client-attached printer.

Example static definition of a Client-attached printer

Note that:

- On the TERMINAL definition:
 - TERMINAL is the name by which the terminal is to be known to the Client. Note that the terminal names passed by Clients are case-sensitive. That is, if a Client passes a name that contains lowercase letters, CICS does not translate it into uppercase. CICS searches for a remote terminal definition whose TERMINAL name is the same combination of mixed or lowercase letters.
 - NETNAME should be allowed to take its default value.

13. A static definition is used only if its REMOTESYSTEM attribute points to a connection to the correct Client, and it is not currently in use. If the definition points to an incorrect Client, or is in use, the call fails.

14. In CICS Transaction Server for OS/390, the autoinstall user program may allocate an alias by which the terminal will be known to CICS. The Client knows the terminal by the supplied name.

15. That is, the timeout delete mechanism does not operate on the remote definitions of Client terminals installed on the CICS on System/390 system on which the CTIN transaction runs. It does operate on Client definitions that are shipped to a back-end CICS/ESA 4.1, CICS Transaction Server for OS/390, or CICS Transaction Server for VSE/ESA system.

installation of Client support

- REMOTESYSTEM must be the name of the connection to the Client.
- USERID must be left blank—you cannot use preset security with Client virtual terminals.
- The PRINTER and ALTPRINTER options are not supported. CICS on System/390 does not provide support for the CICS print key. Any screen print function must be implemented entirely within the Client in its role as a “TOR”; CICS on System/390 is unaware of any such support.
- On the TYPETERM definition, value 2 of the CGCSGID option should specify the code page to be used by the CICS on System/390 server for data conversion. You can use value 1 to specify the server character set. Alternatively, by leaving value 1 set to '00000', you can allow CICS to choose a default character set, deduced from the server code page—see page 110.
- On both definitions, some inessential options have been omitted. You can allow these to take their default values.

```

DEFINE TERMINAL(CPR1) GROUP(CLEPIVT)
OVERTYPE TO MODIFY
CEDA DEFINE
  Terminal    ==> CPR1
  Group       ==> CLEPIVT
  AUTINSTModel ==> No           No|Yes|Only
  AUTINSTName ==>
  TERMINAL IDENTIFIERS
  Typeterm    ==> CLIATPR
  Netname     ==>
  Console     ==> No           No|0-99
  REMOTESystem ==> CLIA
  REMOTESysnet ==>
  REMOTENAME ==>
  ASSOCIATED PRINTERS
  PRINTERCopy ==> No           No|Yes
  ALTPRINTCopy ==> No           No|Yes
  PIPELINE PROPERTIES
  Tasklimit   ==> No           No|1-32767
  OPERATOR DEFAULTS
  OPERPID     ==>
  OPERPriority ==> 000         0-255
  OPERRs1     ==> 0           0-24,...
  OPERSecurity ==> 1         1-64,...
  PRESET SECURITY
  Userid      ==>
  TERMINAL USAGES
  Termpriority ==> 000         0-255
  Inservice   ==> Yes         Yes|No
  SESSION SECURITY
  Attachsec   ==> Local       Local|Identify|Verify|
                                     Persistent|Mixidpe
  BINDPassword ==>           PASSWORD NOT SPECIFIED
  BINDSecurity ==> No         No|Yes

```

Figure 30. Example TERMINAL definition for a Client-attached printer

Here is an example TYPETERM definition to accompany the preceding TERMINAL definition.

```

DEFINE TYPETERM(CLIATPR) GROUP(CLEPIVT)
OVERTYPE TO MODIFY
CEDA DEFINE
  TYPeterm    ==> CLIATPR
  Group       ==> CLEPIVT
RESOURCE TYPE
  DEvice      ==> LUTYPE3
  TERmodel    ==> 2
  SESsiontype ==>
  LDclist     ==>
  SHippable   ==> Yes                No|Yes
MAPPING PROPERTIES
  PAGesize    ==> 024 , 080          0-999
  ALTPage     ==> 024 , 080          0-999
  ALTSuffix   ==>
  FMhparm     ==> No                No|Yes
  OBOperid    ==> No                No|Yes
PAGING PROPERTIES
  AUTOPage    ==> Yes                No|Yes
DEVICE PROPERTIES
  DEFscreen   ==> 024 , 080          0-999
  EXTendedds  ==> Yes                No|Yes
  Query       ==> All                No|Co1d|All
  SOSi        ==> No                No|Yes
  BAcKtrans   ==> No                No|Yes
  CGcsgid     ==> 00000 , 00037      0-65535
SESSION PROPERTIES
  AScii       ==> No                No|7|8
  SENdsizes   ==> 00256              0-30720
  RECEivesize ==> 00256              0-30720
  BRacket     ==> Yes                Yes|No
  LOGMode     ==>
OPERATIONAL PROPERTIES
  AUTOConnect ==> No                No|Yes|All
  ATi         ==> Yes                No|Yes
  TTi         ==> Yes                Yes|No
  CREatesess  ==> No                No|Yes
  RELreq      ==> Yes                No|Yes
  DIScreq     ==> Yes                Yes|No
MESSAGE RECEIVING PROPERTIES
  ROutedmsgs  ==> All                All|None|Specific
  LOGOnmsg    ==> No                No|Yes
APPLICATION FEATURES
  BUildchain  ==> No                No|Yes
  USerarealen ==> 000                0-255
  IOarealen   ==> 00512 , 00000      0-32767
  UCtran      ==> No                No|Yes|Tranid
  
```

Figure 31. Example TYPETERM definition for a Client-attached printer

Security

Important

In CICS Transaction Server for OS/390 and CICS/ESA Version 4.1, CICS intercommunication security is described in detail in the *CICS-RACF Security Guide*. In CICS Transaction Server for VSE/ESA, it is described in the *CICS Transaction Server for VSE/ESA Security Guide*. This section is intended to be read in conjunction with your CICS security manual; it describes security considerations that are specific to CICS Clients and CICS on System/390.

Users of external security managers (ESMs) other than the Resource Access Control Facility (RACF) or the CICS Transaction Server for VSE/ESA ESM should read this section in conjunction with the documentation for their own ESM.

Bind security

Bind-time security is not supported on CICS Client-CICS on System/390 links. Therefore, specify BINDSECURITY(NO) on the CONNECTION definitions that define Clients to CICS on System/390.

Link security

Link security provides the lowest level of resource security for intercommunication links. It defines the total set of resources that can be accessed across the connection.

To specify link security for a CICS Client-CICS on System/390 connection:

1. On the SECURITYNAME option of the CONNECTION definition, specify a user ID for the link.
2. Define a profile to your ESM for the link user ID, which must be a valid RACF user ID. Users of the connection will be able to access only those resources that the link user ID is authorized to access.

If you do not specify a user ID on SECURITYNAME, the authority of the link is that of the CICS default user.

User security

User (attach-time) security:

- Defines how individual users of an intercommunication link are to be checked.
- Affects the resources that individual users are able to access. Unless you specify LOCAL user security (in which case all potential users share the authority of the link user ID), you must define user profiles to your ESM.

Specify the level of user-security for a CICS Client-CICS server link on the ATTACHSEC option of the CONNECTION definition that defines the Client to CICS on System/390. The valid values of ATTACHSEC for CICS Client-System/390 links are LOCAL and VERIFY.

If you specify ATTACHSEC(VERIFY), you must also specify USEDFLTUSER(YES). If you do not, the first time the Client tries to initialize the connection to CICS on System/390 you see security violation messages DFHZN2701 and DFHZC2047 and

an SDUMP is taken. (This is because, when trying to attach the CCIN transaction, the Client does not include the password and user ID required by CICS on System/390.)

If a Client does not support VERIFY attach-time security, you must specify ATTACHSEC(LOCAL) on the CONNECTION definition, and rely on link security.

Note: Do not specify preset security when defining Client virtual terminals or models used for autoinstalling virtual terminals. Preset security is not supported for virtual terminals because attach-time security is used to verify users of CICS Client-CICS on System/390 links.

CICS-supplied transactions

CCIN is a category 3 transaction—that is, it is exempt from security checking.

CTIN is a category 2 transaction—that is, it is always associated with a terminal.

You should specify:

To CICS

For CCIN

RESSEC(NO) and CMDSEC(NO) on the transaction resource definition.

For CTIN

RESSEC(YES) and CMDSEC(YES) on the transaction resource definition.

The supplied definitions in the DFHCLNT CSD group specify these values.

To your ESM

For CTIN

If your external security manager is RACF, a transaction profile that specifies UACC(NONE); and an access list that contains the user IDs (or groups containing user IDs) of users who access CICS from Client workstations. For example:

```
RDEFINE GCICSTRN INTERCOM UACC(NONE)
      ADDMEM(CEHP,CEHS,CPMI, . . . ,CTIN, . . . )
      NOTIFY(security_admin_userid)
      OWNER(userid or groupid)
PERMIT INTERCOM CLASS(GCICSTRN) ID(intrgrp1,..,intrgrpz)
      ACCESS(READ)
```

System initialization parameters

To activate security on CICS Client-CICS on System/390 links, you need to specify the following system initialization parameters:

DFLTUSER=name,	To specify the CICS default userid	*
SEC=YES,	To turn on security checking	*
XTRAN=YES,	To turn on transaction security	*

For detailed information about these parameters, see your CICS on System/390 *System Definition Guide*.

Chapter 9. Data conversion for Clients

Important

Information in this chapter about data conversion for the EPI and the Client terminal emulator function applies only to:

- CICS Transaction Server for OS/390
- CICS Transaction Server for VSE/ESA
- CICS/ESA Version 4.1.

For detailed information about data conversion between CICS workstation products and CICS on System/390 systems, see “Chapter 6. Data conversion” on page 43. This chapter describes considerations specific to conversion between CICS Clients and CICS on System/390.

The CICS on System/390 server must translate between its EBCDIC encoding of character and numeric data, and the ASCII encoding used by the attached Clients.

For the EPI:

- Data conversion is done by the terminal-owning region (TOR)—that is, the CICS on System/390 region on which the Client virtual terminals are installed.
- No data conversion is performed for TCTUAs or COMMAREAs; these are not returned to Client programs.
- Only standard conversion—that is, conversion handled by the CICS on System/390 conversion program, DFHCCNV, using supported code pages—is possible. Non-standard conversion—that is, conversion that relies on user-code in the user-replaceable program, DFHUCNV—is not supported.

For the ECI:

- Data conversion is done by the application-owning region (AOR)—that is, the region in which the server programs run.
- Both standard and non-standard conversion is supported.

The client code page

This section describes the client code pages supported for the ECI, EPI, and Client terminal emulator.

The ECI

The client code pages supported for the ECI are the same as those supported by CICS on System/390 for distributed program link (DPL) requests. They are listed in “CICS-supported conversions” on page 45.

The client code page is specified by the Client. For example, for OS/2 Clients the client code page is that specified in the workstation’s CONFIG.SYS file, or as redefined by a CHCP command. For DOS or Windows Clients, it is the code page specified in the workstation’s AUTOEXEC.BAT file, or as redefined by a CHCP command.

The EPI and terminal emulator

The client code pages supported for the EPI and the terminal emulator function are shown in Table 37 on page 110. They are a subset of those supported for the ECI.

data conversion

The client code page is specified by the Client, as described for the ECI.

For the EPI and terminal emulator, if the client code page is invalid, CICS uses code page 850.

Table 37. EPI code pages. The client code pages supported by CICS on System/390 for the EPI, and the default server code page for each one.

National language group	Supported client code pages for EPI	Default server code page
Arabic	864 1089 1256	420 420 420
Baltic Rim	921 1257	1112 1112
Cyrillic	855 866 915 1251	1025 1025 1025 1025
Estonia	922 1257	1112 1112
Greek	813 869 1253	875 875 875
Hebrew	856 916 1255	424 424 424
Japanese	932 942 943 954	930 930 930 930
Korean	934 944 949 970 1363	933 933 933 933 933
Latin-1	437 819 850 1252	500 500 500 500
Latin-1 including euro support	858 5348	1148 1148
Latin-2	852 912 1250	870 870 870
Latin-5	857 920 1254	1026 1026 1026
Latin-9	923	924
Simplified Chinese	946 1381 1383 1386	935 935 935 935
Traditional Chinese	938 948 950 964	937 937 937 937

The server code page and character set

This section describes the server code pages and character sets supported for the ECI, EPI, and Client terminal emulator.

The ECI

The server code pages supported for the ECI are the same as those supported by CICS on System/390 for DPL requests. They are listed in “CICS-supported conversions” on page 45.

The server code page is determined from the conversion table, DFHCNV, installed on CICS on System/390.

The EPI and terminal emulator

The server code pages and character sets supported for the EPI and the terminal emulator function are shown in Table 38 on page 111. The code pages are a subset of those supported for the ECI.

The server code page is determined from value 2 of the CGCSGID option of the TYPETERM used to install the virtual terminal definition. If the server code page, or the combination of client and server code pages, is invalid, the installation of the virtual terminal is rejected. If no server code page is specified, CICS uses a default, which it deduces from the value of the client code page, as shown in Table 37.

You can use value 1 of the CGCSGID field of the TYPETERM definition to specify the server character set for the EPI. If you specify an invalid character set, the installation of the virtual terminal is rejected. Alternatively, you can use the default character set supplied by CICS on System/390. Table 38 shows the server character sets and code pages supported for the EPI, and the default character set deduced from each server code page.

Table 38. EPI character sets and server code pages. The character sets and server code pages supported by CICS on System/390 for the EPI, and the default character set for each server code page.

National language group	Supported character sets for EPI and default	Supported server code pages for EPI	Character set used by CICS for DBCS	Code page used by CICS for DBCS
Arabic	235	420	—	—
Baltic Rim	1305	1112	—	—
Cyrillic	1150	1025	—	—
Estonia	1307	1122	—	—
Greek	925	875	—	—
Hebrew	941	424	—	—
Japanese	1172 1172 1172	930 931 939	1001 1001 1001	300 300 300
Korean	1173	933	934	834
Latin-1	697 697 697 697 697 697 697 697 697 697 697	037 273 277 278 280 284 285 297 500 871 1047	—	—
Latin-1 including euro support	695 695 695 695 695 695 695 695 695 695	1140 1141 1142 1143 1144 1145 1146 1147 1148 1149	—	—
Latin-2	959	870	—	—
Latin-5	1152	1026	—	—
Latin-9	1353	924	—	—
Simplified Chinese	1174	935	937	837
Traditional Chinese	1175	937	935	835

Binary data conversion

For the ECI, binary data is converted, or not, as specified by the Client.

For the EPI, there is no binary data that needs to be converted.

Defining code pages to CICS on System/390

Bearing in mind the rules outlined above, this is what you code in your DFHCNV table, in the region or regions in which your server programs run, for each program invoked by the ECI:

- A DFHCNV TYPE=ENTRY entry, on which the SRVERCP operand specifies the server code page to be used to translate the communications area (COMMAREA).

data conversion

- DFHCNV TYPE=FIELD entries, on which the DATALEN and DATATYP operands specify, respectively, the length and type of each data field in the COMMAREA.

Note: It is **not** necessary to code, on the CLINTCP and SRVERCP operands of the DFHCNV TYPE=INITIAL macro, the code pages used with CICS Clients.

For definitive information about coding DFHCNV macros to specify code pages, see “Defining the conversion table” on page 59.

Chapter 10. Application programming for Clients

Important

Information in this chapter about EPI server programs and Client-attached printers applies only to:

- CICS Transaction Server for OS/390
- CICS Transaction Server for VSE/ESA
- CICS/ESA Version 4.1.

This chapter gives advice on writing CICS on System/390 server programs to be invoked from CICS Clients.

ECI server programs

Writing a server program to be invoked by the Client External Call Interface (ECI) is similar to writing one to be invoked by a distributed program link (DPL) request. ECI server programs can issue the same subset of EXEC CICS commands as DPL server programs (except that they cannot issue syncpoints, whereas DPL servers can do so if the client program specifies SYNCONRETURN on the LINK command). Thus, the commands an ECI server program *cannot* issue are:

- Terminal-control commands referring to its principal facility
- Commands that set or inquire on terminal attributes
- BMS commands
- Signon and signoff commands
- Batch data interchange commands
- Commands addressing the TCTUA
- Syncpoint commands.

For further information about writing DPL server programs, see your CICS on System/390 *Intercommunication Guide* and *Application Programming Reference* manual.

EPI server programs

The following restrictions apply to CICS on System/390 server programs that are invoked from the Client External Presentation Interface (EPI):

- They must not use:
 - A terminal control (TC) data stream that includes structured fields
 - The terminal control commands EXEC CICS ISSUE DISCONNECT or EXEC CICS ISSUE PASS
 - Basic mapping support (BMS) partition support
 - BMS paging.
- A Client virtual terminal cannot be a target of an EXEC CICS ROUTE command.
- A Client virtual terminal cannot be the target of the CMSG message-routing transaction.
- If you are using CICS-generated TERMIDs, your server programs must not rely on TERMIDs being allocated consistently to particular Client terminals. (If, on the other hand, TERMIDs are always nominated, in a consistent way, by your Client EPI programs, this restriction may not apply.)

A Client terminal can be deleted by the Client sending a **CICS_EpiDelTerminal** request, by an end user shutting down a Client terminal emulator or the Client

application programming for Clients

itself, or if a connection failure occurs. When it is reinstalled, CICS does not necessarily generate the same TERMID as it had previously.

This has implications for the way in which your server programs are written. For example:

- Your server programs derive temporary storage queue names from the TERMID (to associate each queue with a particular end user). Problems of data mismatch could occur if the queue is not deleted by transaction end (possibly due to a failure).

The best solution is for your application programs always to check before creating a temporary storage queue whether a queue of the same name already exists, and, if so, to delete it. However, if you have a large number of server applications, it may not be possible to check or change them all.

- Your server programs record TERMIDs for later use. For example, an application might issue an EXEC CICS START TERMID command, with a time interval after which the transaction is to be initiated against the named terminal. If, during the delay interval, the terminal definition is deleted, and reinstalled with a different TERMID, the started transaction could fail because the TERMID no longer exists.

CICS Transaction Server for OS/390 only

If your server programs cannot be rewritten, your autoinstall user program could allocate aliases to the CICS-generated TERMIDs. It could, for example, use a mapping file to relate particular aliases to particular Client workstations (identified by connection name).

If your server programs are located on a back-end AOR, the autoinstall user program is invoked in the AOR when a virtual terminal is shipped in, just as for any other shipped definition. It could, if necessary, allocate an alias terminal identifier to the shipped definition.

For information about writing an autoinstall user program to control the installation of Client virtual terminals and shipped definitions, see your CICS on System/390 *Customization Guide*.

Client-attached printers

A CICS on System/390 application can communicate with a Client-attached printer by starting a transaction against it, using an EXEC CICS START TERMID command.

A Client-attached printer accepts a 3270 data stream which contains set buffer address (SBA) commands, and a 3270 write control character (WCC) with the print bit set on. An application can use any of the following command sequences, followed by EXEC CICS PRINT, to print data successfully:

- BMS SEND MAP or SEND TEXT with the PRINT option specified
- BMS SEND MAP or SEND TEXT without the PRINT option, followed by BMS SEND CONTROL with the PRINT option
- TC SEND of an appropriate data stream, with the CTLCHAR option.

application programming for Clients

There are some restrictions:

- It is not possible for an application to initiate printing indirectly by starting the basic mapping support (BMS) paging transaction against the printer by a BMS ROUTE command, followed by BMS SEND MAP or SEND TEXT commands with the PRINT and PAGING options, followed by a BMS SEND PAGE command.
- A Client-attached printer does not support the NLEOM data stream (generated by BMS SEND MAP or SEND TEXT with the NLEOM option).

application programming for Clients

Chapter 11. Problem determination for Clients

Important

This chapter applies only to:

- CICS Transaction Server for OS/390
- CICS Transaction Server for VSE/ESA
- CICS/ESA Version 4.1.

This chapter tells you where to find information to help you diagnose Client-related problems.

Trace points

Client-related trace points are in the range AP 3000-AP 3075; they are listed in your CICS on System/390 *Diagnosis Reference* manual.

Notes:

1. Turning on TC level 2 tracing causes the amount of trace information produced to increase significantly.
2. Turning on TC level 2 tracing when you are using DBCS code pages with Client virtual terminals causes the amount of trace information to increase substantially.

For advice about how to use CICS trace, see your CICS on System/390 *Problem Determination Guide*.

Messages

Messages relating to Client support are in the range DFH3C3202-DFH3C3249. They are written to the CSCC transient data queue and are listed in your CICS on System/390 *Messages and Codes* manual.

Abend codes

Abend codes relating to Client support are in the ranges AXTP-AXTR and AZAD-AZAK. They are listed in your CICS on System/390 *Messages and Codes* manual.

Chapter 12. Recovery after a restart of CICS

Important

Information in this chapter about the EPI and the Client terminal emulator function applies only to:

- CICS Transaction Server for OS/390
- CICS Transaction Server for VSE/ESA
- CICS/ESA Version 4.1.

This chapter describes how users of attached Client workstations are affected if CICS on System/390 is restarted.

After a restart of CICS on System/390, sessions to CICS Clients are recovered automatically.

Note: If CICS on System/390 is using VTAM persistent sessions support, VTAM holds on to the sessions to the Client until the restart occurs; they are then unbound by CICS. (Using persistent sessions causes a restart to take slightly longer than it otherwise would.)

There are some restrictions on the use of persistent sessions with CICS Clients—see “Chapter 13. Restrictions on Client support” on page 121.

The Client terminal emulator

If CICS on System/390 is restarted it is not necessary to restart the Client terminal emulator. The user should:

1. Hit Enter after CICS on System/390 has crashed.

The following is displayed on the emulator screen:

```
CCL7045E Connection lost with server 'CICSSNA'  
CCL7020I Press Clear to continue
```

2. Hit the Clear key.
CCL7045E is displayed on the operator information line.
3. Wait for a minute or so after CICS has come back up while CICS reacquires the connection.
4. If the restart interrupted a Client transaction, rerun the transaction, if necessary. To discover whether a rerun is required, you may have to investigate whether any updates to resources were completed successfully.

Client EPI and ECI programs

In the event of a failure and restart of CICS on System/390, Client EPI and ECI programs are responsible for:

- Displaying appropriate messages on the workstation
- Taking any recovery actions that may be necessary.

For information about writing Client EPI and ECI programs, see the *CICS Clients: Client/Server Programming* manual.

If a restart interrupts a Client EPI or ECI program, the end user may need to rerun the associated transaction. Note that CICS Clients do not support synchronization

recovery after CICS restart

level (synclevel) 2 conversations. Therefore, to discover whether a rerun is required, you may have to investigate whether any updates to resources were completed successfully.

Chapter 13. Restrictions on Client support

Important

Information in this chapter about the CCIN and CTIN transactions, the EPI, Client virtual terminals, and the Client terminal emulator function, applies only to:

- CICS Transaction Server for OS/390
- CICS Transaction Server for VSE/ESA
- CICS/ESA Version 4.1.

There are some restrictions on CICS on System/390 support for CICS Clients.

The following general restrictions apply:

- The CCIN and CTIN transactions (invoked by the Client to exchange connection details with the server and to install Client terminals) can run only on a CICS system that is directly connected to the Client. A CICS system that is directly connected to the Client can, however, use normal transaction routing and function shipping flows to communicate with any other CICS system.

If an EPI request is routed to a back-end AOR, the AOR can be any currently-supported release of CICS.

If an ECI request is shipped to a back-end AOR, the AOR must be one of the following:

- CICS/ESA 3.3
- CICS/ESA 4.1
- CICS/VSE 2.3
- CICS Transaction Server for OS/390
- CICS Transaction Server for VSE/ESA.

(This is because data conversion for the ECI is done in the AOR.) Also, the Client must support the use of APPC to pass information, such as the client code page, between CICS on System/390 systems. (This is because the information passed by the Client to CICS on System/390 by means of the CCIN transaction is not passed by CICS on System/390 to a back-end AOR.)

- APPC connections to Clients support data synchronization levels (synclevels) 0 and 1 only.
- When acting as a server to a CICS Client using the EPI, CICS on System/390:
 - Does not support the use of a PF key specified on the TASKREQ option of a TRANSACTION definition to start the transaction from a Client virtual terminal.
 - Does not support use of the print key specified on the PRINT system initialization parameter from a Client virtual terminal.
 - Does not display the good morning panel when a Client terminal is installed. Note that you can use the CICSTERM command to specify the initial transaction for a CICS Client.
- You can make only limited use of VTAM persistent sessions support to recover connections to Clients automatically after a failure. If you define your Client connections to use persistent sessions, the only sessions that are recovered are those on:
 - Statically defined connections
 - Autoinstalled single-session connections.

These sessions are recovered only if:

restrictions

- No virtual terminals were installed on the connection at the time of the failure.
- For autoinstalled connections, the value of the AIRDELAY system initialization parameter is greater than zero.
- You can use the execution diagnostic facility (EDF) in single-terminal mode from a Client emulator, to test a CICS on System/390 server transaction.
EDF in two-terminal mode is supported only when both of the terminals and the transaction reside on the same CICS system; that is, when no Client terminal is involved.
- When running a Client emulator, you cannot use the CESN or CESF transactions to sign on to (or off from) the CICS region to which the Client is directly attached.
- EPI server transactions running in the CICS region to which the Client is directly attached can issue EXEC CICS SIGNON or SIGNOFF commands only if the virtual terminal has been installed as signon capable.

#

#

Note: Transactions started at a virtual terminal installed as signon capable are executed with the authorities assigned to either the default user defined by the DFLTUSER system initialization parameter or the user currently signed on at the terminal.

|
|
|
|

- You cannot install more than 512 virtual terminals per Client connection. This limit is necessary to prevent service attacks. An attempt to install more than 512 virtual terminals results in message DFHZC3206; the request to install the virtual terminal is rejected. A response code of DISASTER with a reason code of INVALIDREQUEST is sent to the Client.
- You cannot use preset security with Client virtual terminals.
- You cannot use the EXEC CICS or CECI ISSUE PASS command with Client virtual terminals. (ISSUE PASS is used to disconnect a VTAM terminal from CICS, and transfer it to another CICS terminal-owning region.)
- On CICS System/390 products other than CICS Transaction Server for OS/390 Release 3, the CEMT INQUIRE TERMINAL command does not return information about Client virtual terminals. However, the EXEC CICS INQUIRE TERMINAL command does.
- If you discard or reinstall a (statically defined or autoinstalled) connection to a Client workstation, and the connection is in use, the workstation end user must shut down and restart the Client before being able to continue.
- If you discard or reinstall a (statically defined or autoinstalled) Client virtual terminal, and the CTIN transaction has been run to install or reserve the terminal but not to release it, the workstation end user must restart the EPI program or terminal emulator that uses the virtual terminal, if the user wants to continue to use it.

|
|
|
|

As well as these general restrictions, there are some restrictions on server programs invoked from the ECI and EPI—see “Chapter 10. Application programming for Clients” on page 113.

Chapter 14. Migration considerations

Important

Information in this chapter about the EPI and the Client terminal emulator function applies only to:

- CICS Transaction Server for OS/390
- CICS Transaction Server for VSE/ESA
- CICS/ESA Version 4.1.

This chapter describes the effects that CICS on System/390 support for CICS Clients could have on an existing CICS on System/390 system.

Moving to a client/server environment

When you first install Client support, it is likely that most, if not all, of your existing CICS on System/390 transactions use 3270 data streams. To migrate to client/server solutions in a staged manner, you could use the following sequence:

1. Use the Client 3270 emulator to run, unchanged, existing CICS on System/390 3270 applications.
2. Use the EPI to add graphical user interface (GUI) front ends to existing CICS on System/390 3270 applications.
3. Use the ECI to develop new client/server applications in which the display and processing logic is appropriately split between the client and the server.

Server applications

You need to decide which of your CICS on System/390 applications are suitable for use as servers to CICS Clients. See “Chapter 10. Application programming for Clients” on page 113.

Data conversion

You should review your use of CICS on System/390 data conversion facilities.

The External Call Interface

If the ECI is used by a Client, consider the following:

- Data conversion is done in the application-owning region.
- The *client* code page is *always* specified by the Client. Any client code page specified in the CICS conversion table, DFHCNV, is overridden by the Client. (The code page overridden is the value of the CLINTCP option on the DFHCNV TYPE=ENTRY macro for the server program.)
- The *server* code page is that specified in the DFHCNV conversion table (on the DFHCNV TYPE=ENTRY macro for the server program).
- For binary data conversion, the default binary format defined on DFHCNV TYPE=FIELD entries is overridden by the Client.

The External Presentation Interface

If the EPI is used by a Client, consider the following:

- Data conversion is done in the terminal-owning region.

migration

- The *client* code page is specified by the Client. If the Client specifies an invalid code page, code page 850 is used.
- The *server* code page is determined from CICS on System/390's remote definition of the virtual Client terminal. The second value of the CGCSGID option of the TYPETERM definition is used, if specified. If a value is specified, it must be a code page that is supported by CICS on System/390. If no value is specified, a default code page is used, as shown in Table 38 on page 111.
- No binary data conversion is required for the EPI.

For more information about defining code pages and conversion tables for data conversion between CICS on System/390 and non-System/390 systems, see "Chapter 6. Data conversion" on page 43.

Part 3. Appendixes

Glossary of Client-related terms

This glossary defines the CICS Client-related terms used in this book.

A

APPC. See *Advanced Program-to-Program Communication*.

Advanced Program-to-Program Communication (APPC). An implementation of the SNA/SDLC LU 6.2 protocol that allows interconnected systems to communicate and share the processing of programs. The APPC protocol is used for communication between IBM CICS Clients and CICS on System/390 systems.

attach. In SNA, the request unit that flows on a session to initiate a conversation.

autoinstall. A facility that enables terminal definitions to be created automatically in a CICS server on request. The definition is deleted when the Client terminal logs off.

On CICS on System/390 systems, APPC connections to Clients can be autoinstalled too.

autoinstall model. On CICS on System/390, a pair of TERMINAL and TYPETERM (or CONNECTION and SESSIONS) definitions that can be used as a template to install terminals (or connections) dynamically, when they log on to CICS.

B

BIND. In SNA, a request to activate a session between two logical units (LUs).

business logic. The part of a distributed application that is concerned with the application logic rather than the user interface of the application. See *presentation logic*.

C

CCIN. A CICS on System/390 internal transaction, invoked when a CICS Client initiates or ends communication with CICS on System/390. It is used to update the connection definition on CICS on System/390.

CICS Clients. A family of CICS workstation products that provide a standard set of functions for *client/server* computing. Each CICS client is designed to run on a particular operating system. Each can attach to a range of CICS *server* systems, and provides access to resources owned by the servers. See *external call interface* and *external presentation interface*.

CICS on Open Systems. A term used to refer generically to the products:

- Transaction Server for AIX Version 4.1 (which contains CICS for AIX Version 2.1.1)
- TX Series Version 4.2 for AIX (which contains CICS for AIX Version 4.2)
- CICS for HP-UX Version 2.1.1
- TX Series Version 4.2 for HP-UX (which contains CICS for HP-UX Version 4.2)
- CICS for Solaris Version 2.1.1
- TX Series Version 4.2 for Solaris (which contains CICS for Solaris Version 4.2)

CICS on System/390. A term used to refer generically to the products: CICS Transaction Server for OS/390, CICS Transaction Server for VSE/ESA, CICS/ESA, and CICS/VSE.

client. See *client/server* and *CICS Clients*.

client/server. A distributed application design model, in which the client sends a request to its partner, the server, which executes the request and returns the results to the client. This design model is associated typically with cooperative processing.

client initialization file. A file containing configuration information used to inform the CICS Client of the CICS servers it can connect to, and the communication protocols to be used.

COMMAREA. A communication area that is used for passing data both between programs within a transaction and between transactions.

connection. In data communication, an association established between functional units for conveying information.

control table. In CICS, a storage area used to describe or define the configuration or operation of the system.

conversation. The communication between a CICS Client and a CICS server. In APPC, a connection between two transaction programs over an LU-LU session that allows them to communicate with each other while processing a transaction. See also *session*.

cooperative processing. A subset of *distributed processing*, in which at least the user interface (presentation) aspect of the application runs on a programmable workstation.

CTIN. A CICS on System/390 internal transaction, invoked as a result of a Client EPI program issuing a **CICS_EPIAddTerminal** or **CICS_EPIDelTerminal** request, or a workstation user starting or shutting down

glossary of Client-related terms

a Client terminal emulator. It is used to install, reserve, or delete a virtual terminal definition on CICS on System/390.

D

DBCS. See *double-byte character set*.

distributed application. An application for which the component application programs are distributed between two or more interconnected processors.

distributed processing. The processing of different parts of the same application in different systems, on one or more processors.

double-byte character set (DBCS). A set of characters in which each character is represented by 2 bytes. Languages such as Japanese, Chinese, and Korean, which contain more symbols than can be represented by 256 code points, require double-byte character sets. Because each character requires 2 bytes, the typing, display, and printing of DBCS characters requires hardware and programs that support DBCS. Contrast with *single-byte character set (SBCS)*.

E

ECl. See *external call interface*.

emulator, emulation program. A program that allows a host system to communicate with a workstation in the same way as it would with the emulated terminal. In IBM CICS Clients, the terminal emulation function allows client workstations to run CICS transactions that use 3270 data flows.

EPI. See *external presentation interface*.

external call interface (ECI). An application programming interface that allows a non-CICS program to link to a CICS program. Data is exchanged in a COMMAREA as for normal CICS interprogram communication.

external presentation interface (EPI). An application programming interface that allows a non-CICS program to appear to CICS as one or more standard 3270 terminals. 3270 data can be presented to the user by emulating a 3270 terminal or by using a graphical user interface.

K

keyboard mapping file. A file used by IBM CICS Clients to customize the keyboard mapping of client workstations.

L

logical unit (LU). In SNA, a port through which an end user accesses the SNA network to communicate with another end user, and through which the end user accesses the functions provided by system services control points (SSCP). An LU can support at least two sessions, one with an SSCP and one with another LU, and may be capable of supporting many sessions with other logical units. See *network addressable unit (NAU)*.

LU 6.2. See *APPC*.

M

mode. In SNA, a set of parameters that defines the characteristics of a session between two LUs.

N

NAU. See *network addressable unit*.

network address. In SNA, an address, consisting of subarea and element fields, that identifies a link, link station, or network addressable unit (NAU). Subarea nodes use network addresses; peripheral nodes use local addresses. The boundary function in the subarea node to which a peripheral node is attached transforms local addresses to network addresses and vice versa. See also *network name*.

network addressable unit (NAU). In SNA, a logical unit, a physical unit, or a system services control point. The NAU is the origin or the destination of information transmitted by the path control network. See also *logical unit, network address, network name*.

network name. In SNA, the symbolic identifier by which end users refer to a network addressable unit (NAU), link station, or link. See also *network address*.

P

parallel session. In SNA, two or more concurrently active sessions between the same two LUs using different pairs of network addresses. Each session can have independent session parameters.

PIP. See *program initialization parameters (PIP)*.

presentation logic. The part of a distributed application that is concerned with the user interface of the application. See *business logic*.

principal facility. To a transaction, the session that activates it.

program initialization parameters. In CICS distributed transaction programming, specially formatted

data passed to a back-end transaction on the CONNECT PROCESS command.

S

SAA. See *Systems Application Architecture*.

SBCS. See *single-byte character set*.

server. The member of the *client/server* partnership that executes the request and returns the results to the client.

session. In SNA, a logical connection between two network addressable units (NAUs) that can be activated, tailored to provide various protocols, and deactivated as requested. The session activation request and response can determine options for such things as rate and concurrency of data exchange, control of contention and error recovery, and characteristics of the data stream. Sessions compete for network resources, such as links within the path control network. For routing purposes, each session is identified by the network or local addresses of the session partners.

shippable terminal . A terminal whose definition can be shipped to another CICS system as and when the other system requires a remote definition of that terminal.

single-byte character set. A character set in which each character is represented by 1 byte.

SIT. See *system initialization table*.

SNA. See *Systems Network Architecture*.

synchronization level. The level of synchronization (0, 1, or 2) established for an APPC session.

synchronization point. In the processing of a transaction, a point at which resources are declared to be in a consistent state.

synclevel. Abbreviation of *synchronization level*.

syncpoint. Abbreviation of *synchronization point*.

system initialization table (SIT). In CICS on System/390, a table of user-specified data that controls how CICS is initialized.

Systems Application Architecture (SAA). A set of IBM software interfaces, conventions, and protocols that provide a framework for designing and developing applications across multiple computing environments.

Systems Network Architecture (SNA). The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through, and controlling the configuration and operation of, networks.

T

TCP/IP. Transport Control Protocol/Internet Protocol.

terminal emulation. The capability of a microcomputer or personal computer to operate as if it were a particular type of terminal linked to a processing unit, and to access data.

trace. A record of data that provides a history of events that occurred in a system. The process of recording the sequence in which the statements in a program are executed and, optionally, are the values of the program variables used in the statements.

transaction routing. A type of CICS intercommunication in which terminal operators of a CICS system can run transactions in a connected CICS system.

V

Virtual Telecommunications Access Method (VTAM). A set of programs that maintains control of the communication between terminals and applications running under an operating system.

virtual terminal. A remote terminal definition installed on CICS on System/390 to support a Client EPI program or terminal emulator. Virtual terminals can be statically defined to CICS on System/390, or autoinstalled on request from a Client.

VTAM. See *Virtual Telecommunications Access Method*.

W

Windows NT. An operating system from Microsoft Corporation.

Windows 95. An operating system from Microsoft Corporation for IBM-compatible personal computers.

| **Windows 98.** An operating system from Microsoft
| Corporation for IBM-compatible personal computers.

workstation. A personal computer or intelligent terminal with some local processing capabilities.

glossary of Client-related terms

Index

Numerics

3270 Terminal Emulator 93

A

APPC protocol, for Client-CICS on System/390
links 93

application programming

ECI server programs 113

EPI server programs 113

ASCII, translation to EBCDIC 94, 109

asynchronous processing 8

ATI (automatic transaction initiation) 8

ATTACHSEC option, CEDA DEFINE

CONNECTION 106

effect on signon attributes 36

specification 35

autoinstall

Client terminals 100

connections to Clients 96

automatic transaction initiation (ATI) 8

B

benefits of Client support 93

bibliography 94

binary data, conversion of 111

binary integers (INTEL format), conversion of 67

C

C programming language, integer datatype
conversion 67

CCIN transaction 95

CGCSGID option, CEDA DEFINE TYPETERM 98, 101

CICS Clients

benefits of support for 93

defining connections on CICS on System/390 95

defining virtual terminals to CICS on System/390 98

for DOS 91

for Macintosh 91

for Microsoft Windows 91

for OS/2 91

functions provided

External Call Interface 91

External Presentation Interface 92

terminal emulation 93

glossary of terms 127

overview 91

protocols for CICS on System/390 links 93

servers supported 91

CICS_EpiAddTerminal function 98, 101

CICS/ESA 9

CICS-supplied resource definition group, DFHCLNT 95

CICS-supplied transactions

CCIN 95

CTIN 95

CICS Transaction Server for OS/390 9

CICS Transaction Server for VSE/ESA 9

CICS/VSE 9

cicsterm command 98, 101

Client-attached printer, defining to CICS 103

client code page, how determined 109

client/server computing 91

CLINTCP operand, DFHCNV TYPE=INITIAL
macro 112

code pages

client, how determined 109

defining to CICS on System/390 111

server, how determined 110

connection

definition (for parallel sessions) 26

security 35

conversion templates 60, 61, 80, 83

field conversion records 80, 81, 82, 83

CSCC, transient data queue 95

CTIN transaction 95

D

data conversion

Arabic conversions 47

assembling/link-editing the conversion programs 76

Baltic Rim conversions 47

binary data 111

binary integers (INTEL format) 67

C programming language, integer datatype 67

character data 45

client code page 109

conversion process 55

conversion templates 61

Cyrillic conversions 47

defining code pages to CICS on System/390 111

defining the conversion table 59, 76

DSECT for data conversion template 82

Estonian conversions 48

Greek conversions 48

Hebrew conversions 48

IVP (initial program verification) 60

Japanese conversions 52

key templates 61

Korean conversions 53

Latin-1 conversions 49, 50

Latin-2 conversions 50

Latin-5 conversions 51

Latin-9 conversions 51

migration considerations 123

nonstandard conversion 56

program 22

resource definition 59, 76

sequence of conversion processing 57

server code page 110

simplified Chinese conversions 53

standard conversion 56

support for 94

table 22

data conversion (*continued*)
 traditional Chinese conversions 54
 types of conversion 44
 where conversion takes place 43

data synchronization 121

DATALEN operand, DFHCNV TYPE=FIELD macro 112

DATATYP operand, DFHCNV TYPE=FIELD macro 112

DB2 database access 6, 8

DBCS (double-byte character set)
 defining DBCS data fields 67
 included in standard conversion 44
 invalid and undefined characters 71
 mixed strings, SBCS/DBCS 67
 user-defined conversion tables 68, 72

DevType parameter
 of CICS_EpiAddTerminal call 101

DFHCCNV, standard conversion program 56

DFHCICSA, supplied profile 20

DFHCLNT resource definition group 95

DFHCNV, resource definition macro 59, 76
 coding examples 72
 coding hints 68
 example sequence 60
 macro types 59
 TYPE=ENTRY 63
 TYPE=FIELD 66
 TYPE=FINAL 68
 TYPE=INITIAL 61
 TYPE=IVP 60
 TYPE=KEY 64
 TYPE=SELECT 65

DFHCNV TYPE=DSECT macro 76

DFHCNV TYPE=ENTRY macro
 SRVERCP operand 111

DFHCNV TYPE=FIELD macro
 DATALEN operand 112
 DATATYP operand 112

DFHCNV TYPE=INITIAL macro
 CLINTCP operand 112
 SRVERCP operand 112

DFHCNVDS, DSECT for field conversion records 82

DFHSG PROGRAM=TCP macro 18

DFHUCNV, user-replaceable conversion program 76
 conversion template 80
 DFHCNV TYPE=DSECT macro 76
 DSECT for data conversion template 82
 DSECT for parameter list 77
 in conversion process 56, 59
 parameter list, DFHUCNV 77
 resource definition 22
 supplied version 83

DFHUNVDS, DSECT for DFHUCNV parameter list 77

distributed transaction processing (DTP) 8

DL/I database access 4, 6, 8

DPL (distributed program link) 6, 113
 COMMAREA 7
 performance optimization 7
 syncpointing 16

DTP (distributed transaction processing) 8

E

EBCDIC, translation to ASCII 94, 109

ECI (External Call Interface)
 overview 91
 restrictions on server programs 113

EDF (execution diagnostic facility) 122

EPI (External Presentation Interface)
 CICS_EpiAddTerminal function 98, 101
 defining Client-attached printer to CICS 103
 defining Client terminals to CICS 98
 overview 92
 restrictions on server programs 113

ESM (external security manager) 106

execution diagnostic facility (EDF) 122

external security manager (ESM) 106

EXTSEC transaction attribute 35

F

field conversion records 80, 83

file security 36

function shipping 3
 from CICS for OS/2, syncpointing 16

G

generation, system 18

glossary of Client-related terms 127

I

INBFMH operand, CEDA DEFINE PROFILE 20

initialization, system 18

installation checklist for Client support 95

invalid DBCS characters 71

IVP (initial program verification), data conversion table 60

L

LU 6.2
 monitoring restriction 6
 supported functions 3, 11

LUTYPE6.2 protocol, for Client-CICS on System/390
 links 93

M

messages 117

migration
 data conversion 123
 moving to a client/server environment 123
 server programs 123

mirror program and transaction
 definition 20
 security 34

monitoring restriction, CICS for OS/2 network name 6

moving to a client/server environment 123

N

NetName parameter
 of CICS_EpiAddTerminal call 98, 101

O

OPERRSL, terminalsession attribute 35
OPERSECURITY, terminalsession attribute 35

P

performance optimization, DPL 7
persistent sessions, restriction on 121
planning 11, 17
 resource definition 11
 syncpointing 15
problem determination, for CICS Clients
 abend codes 117
 messages 117
 trace points 117
protocols for Client-CICS on System/390 links
 APPC 93
 TCP/IP 93

R

RACF (Resource Access Control Facility) 106
remote definitions
 CICS for OS/2 terminals 28
 terminals 5
 transactions 5
Resource Access Control Facility (RACF) 106
resource definition 17
 Client terminals
 using autoinstall 100
 using static definitions 98
 connections to Clients
 using autoinstall 96
 using static definitions 95
 CSCC transient data queue 95
 data conversion 59, 76
 defining code pages to CICS on System/390 111
 defining the conversion table 59
 DFHCLNT resource group 95
 LU 6.2 19
 CONNECTION for parallel sessions 26
 data conversion program 22
 mirror program and transaction 20
 remote terminal 28
 remote TYPETERM 30
 SESSIONS for parallel sessions 27
 workstation TERMINAL 23
 workstation TYPETERM 25
 planning 11
restrictions
 DPL 6
 on ECI server programs 113
 on EPI server programs 113
 sync levels supported 121
 transaction routing 6
 VTAM persistent sessions 121
RSL attribute
 file 36
 program 34
 transaction 34, 35
RSLC, transaction attribute 34, 36

S

security 31
 attach-time 106
 bind-time 106
 CICS for OS/2, CICS on Open Systems, CICS/400
 specifications 32
 connection 35
 file 36
 link 106
 mirror program and transaction 34
 of CICS-supplied transactions 107
 session 35
 signon table 36
 system initialization parameters 107
 temporary storage 36
 transaction 35
 transient data 36
 user 36, 106
server code page, how determined 110
server programs
 for use with ECI 113
 for use with EPI 113
server support
 data conversion
 binary data 111
 client code page 109
 server code page 110
 migration 123
 problem determination
 messages 117
 trace points 117
 resource definition
 Client-attached printer 103
 Client terminals 98
 connections to Clients 95
 CSCC transient data queue 95
 DFHCLNT resource group 95
 restrictions
 on ECI server programs 113
 on EPI server programs 113
 sync levels supported 121
 VTAM persistent sessions 121
 security 106, 107
server support for Clients
 installation checklist 95
sessions
 definition (for parallel sessions) 27
 security 35
shippable terminal 129
signon table 36
SQL database access 6
SRVERCP operand, DFHCNV TYPE=ENTRY
 macro 111
SRVERCP operand, DFHCNV TYPE=INITIAL
 macro 112
sync level support 121
syncpointing 15, 17
 DPL 16
 function shipping from CICS for OS/2 16
 function shipping from CICS on System/390 15
 synchronization level (synclevel) 15

- system generation 18
- system initialization 18
- system initialization parameters
 - DFLTUSER 107
 - SEC 107
 - security-related 107
 - VTPREFIX 101
 - XTRAN 107

T

- TCP/IP protocol, for Client-CICS on System/390
 - links 93
- temporary storage, security 36
- terminal emulation 93
- Terminal Emulator
 - cicsterm command 98, 101
- trace points 117
- transaction routing 4
- transaction security 35
- TRANSEC transaction attribute 34, 35
- transient data, security 36
- transient data queue, CSCC 95

U

- undefined DBCS characters 71
- user security 36

V

- virtual terminals, for use by Clients 98
- VTAM/NCP definitions 39
- VTAM persistent sessions 121
- VTPREFIX, system initialization parameter 101

W

- workstation, programmable
 - LU 6.2 TERMINAL definition on CICS on System/390 23
 - LU 6.2 TYPETERM definition on CICS on System/390 25
- writing server programs
 - for use with the ECI 113
 - for use with the EPI 113

Sending your comments to IBM

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, to this address:
Information Development Department (MP095)
IBM United Kingdom Laboratories
Hursley Park
WINCHESTER,
Hampshire
United Kingdom
- By fax:
 - From outside the U.K., after your international access code use 44-1962-870229
 - From within the U.K., use 01962-870229
- Electronically, use the appropriate network ID:
 - IBM Mail Exchange: GBIBM2Q9 at IBMMAIL
 - IBMLink™ : HURSLEY(IDRCF)
 - Internet: idrcf@hursley.ibm.com

Whichever you use, ensure that you include:

- The publication number and title
- The topic to which your comment applies
- Your name and address/telephone number/fax number/network ID.



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

SC33-1697-03



Spine information:



CICS® Family:

Communicating from CICS on System/390