CICS Family

Communicating from CICS on System/390

SC34-6474-04

CICS Family

Communicating from CICS on System/390

SC34-6474-04

Note!
Before using this information and the product it supports, be sure to read the general information under “Notices” on page 157.

Tenth Edition (July 2010)

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 z/OS Version 3
CICS Transaction Server for z/OS Version 2, program number 5697–E93
CICS Transaction Server for OS/390 Version 1, program number 5655-147
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 eigth edition of the *CICS Family: Communicating from CICS on System/390* manual, SC34-6474-02. 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.

Contents

Preface

What this book is about

This book is about setting up a CICS® System/390® product—CICS Transaction Server for z/OS®, CICS Transaction Server for OS/390®, CICS Transaction Server for VSE/ESA, or CICS/VSE—to communicate with a non-System/390 CICS product—CICS on Open Systems, CICS Transaction Server for Windows®, 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 chapters 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 Transaction Server for Windows, 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 Guide*s 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 149.

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, System/390, or zSeries® family, and support communication with CICS products that run on other hardware platforms. (Not all of these products run on all of these computers—for example, CICS Transaction Server for z/OS Version 2 does not run on System/370.)

- CICS Transaction Server for z/OS Version 3
- CICS Transaction Server for z/OS Version 2, program number 5697–E93
- CICS Transaction Server for OS/390 Version 1, program number 5655-147
- CICS Transaction Server for VSE/ESA, program number 5648-054
- CICS/VSE Version 2, program number 5686-026

In this book, the term **System/390** is used to refer to any System/370, System/390, or zSeries 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—for example, iSeries® (used by CICS/400), IBM-compatible personal computers (used by CICS Transaction Server for Windows), 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 on System/390 product.

The term *CICS Transaction Server for z/OS*, without a qualifying Version number, is used as a generic term for:

- CICS Transaction Server for z/OS Version 3 Release 1
- CICS Transaction Server for z/OS Version 2 Release 3
- CICS Transaction Server for z/OS Version 2 Release 2

The term *CICS Transaction Server for OS/390*, without a qualifying Version number, refers to CICS Transaction Server for OS/390 Release 3.

The term *CICS Transaction Server for Windows*, without a qualifier, means CICS Transaction Server for Windows, Version 5.0.

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

- TXSeries for Multiplatforms Version 5.1, which contains:
  - CICS for AIX®
  - CICS for HP-UX
  - CICS for Sun Solaris
  - CICS for Windows NT®
- TXSeries Version 4.3 for AIX (which contains CICS for AIX)
- TXSeries Version 4.3 for Sun Solaris (which contains CICS for Sun Solaris)

- TXSeries Version 4.3 for Windows NT (which contains CICS for Windows NT)
- TXSeries Version 4.2 for HP-UX (which contains CICS for HP-UX)

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

The term *CICS Transaction Server for VSE/ESA* means CICS Transaction Server for VSE/ESA Release 1.1.1.

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

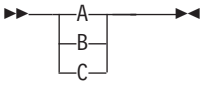
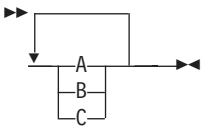
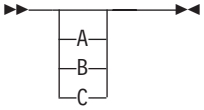
The term *CICS/400* is used as a generic name for:
- CICS/400 Version 4 Release 5
- CICS Transaction Server for iSeries

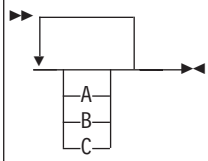
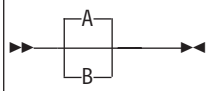
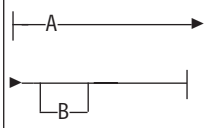
The term *CICS Clients* is used as a generic term for:
- The CICS Universal Client (for Windows NT, Windows 2000, and Windows XP)
- The CICS Client elements of the CICS Transaction Gateway products
- The client daemons of the CICS Transaction Gateway products

The notation **CICS–CICS TS for Windows** 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 63.

Symbol	Action
	A set of alternatives—one of which you ***must*** code.
	A set of alternatives—one of which you ***must*** code. You ***may*** code more than one of them, in any sequence.
	A set of alternatives—one of which you ***may*** code.

Symbol	Action
	A set of alternatives — any number (including none) of which you may code once, in any sequence.
	Alternatives where **A** is the default.
 **Name:** 	Use with the named section in place of its name.
Punctuation and uppercase characters	Code exactly as shown.
Lowercase characters	Code your own text, as appropriate (for example, `name`).

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

Chapter 2, "Planning for CICS System/390–non-System/390 intercommunication"
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 for communication with non-System/390 systems"
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 for communication with non-System/390 systems"
describes the CICS on System/390 resource definitions needed to enable data conversion for communication with CICS Transaction Server for Windows 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 103
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, "Installing 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"

Summary of changes

This book is based on the eighth edition of the *Communicating from CICS on System/390* manual, SC34-6474-02. Changes from that edition are marked by vertical bars in the left margin.

This part lists briefly the changes that have been made for the following editions:

Changes for the seventh edition

The major changes for this edition are:

- The book has been revised to take account of the following new products:
  - CICS Transaction Server for z/OS Version 3
- The DFHCNV macro, used to define data conversion templates, has been updated to allow you to specify client and server code page defaults in the system initialization table. This can simplify CICSplex management by reducing the number of conversion tables needed. See "Defaults for client and server code pages" on page 64.

Changes for the sixth edition

The major changes for this edition were:

- The book was revised to take account of the following new products:
  - CICS Transaction Server for z/OS Version 2 Release 3
  - CICS Transaction Server for Windows, Version 5.0
- References to the following CICS products, which are no longer supported, were removed:
  - CICS Transaction Server for z/OS Version 2 Release 1
  - CICS Transaction Server for OS/390 Release 2
  - CICS Transaction Server for OS/390 Release 1
  - CICS Transaction Server for VSE/ESA Release 1.0
  - CICS/ESA Version 4.1
  - CICS Transaction Server for OS/2 Warp Version 4.1
  - CICS for OS/2 Version 3.1
- The tables of client and server code pages in Chapter 6, "Data conversion for communication with non-System/390 systems," on page 43 were updated.
- The tables of client and server code pages in Chapter 9, "Data conversion for Clients," on page 127 were updated.
- In CICS TS for z/OS Version 2.3 and later, the DFHCNV macro, used to define data conversion templates, is extended. Whereas previously you had to define a separate template for each resource to which data conversion was to be applied, it is now possible to define generic templates that apply to multiple resources. The new support for generic templates was described in "Defining the conversion table" on page 63.

Changes for the fifth edition

The major changes for this edition were:

- The book was revised to take account of the following new product:
  - CICS Transaction Server for z/OS Version 2 Release 2
- CICS TS for z/OS Version 2.2's support for user-named data conversion programs was described in "User-named conversion programs" on page 82.
- The new ECI over TCP/IP function was described in "Using ECI over TCP/IP" on page 108 and "Using ECI over TCP/IP" on page 114.

Changes for the fourth edition

The more significant changes for this edition were:

- The book was revised to take account of the following new product:
  - CICS Transaction Server for z/OS Version 2 Release 1
- The tables of client and server code pages in Chapter 6, "Data conversion for communication with non-System/390 systems," on page 43 were updated.
- The tables of client and server code pages in Chapter 9, "Data conversion for Clients," on page 127 were updated.
- References to CICS/ESA Version 3 were removed, because this product is no longer supported.

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. It contains the following topics:

Note: CICS on System/390’s support for the CICS Clients products is described in Part 2, “Server Support for CICS Clients,” on page 103.

Chapter 1. Overview of CICS System/390–non-System/390 intercommunication

> **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. Every CICS on System/390 product supports all these facilities.

Function shipping (see "Function shipping")
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 "Transaction routing" on 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 "Distributed program link (DPL)" on 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 "Asynchronous processing" on 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 "Distributed transaction processing (DTP)" on 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.

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 and CICS/400. It is *not* supported by CICS Transaction Server for Windows. 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 14.

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 recoverable temporary storage queue and then write to it, without issuing a syncpoint between the two commands.

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 Transaction Server for Windows, CICS on Open Systems, or CICS/400, use distributed transaction processing or the distributed program link function.

Data conversion

CICS Transaction Server for Windows and CICS on Open Systems use ASCII[1] data representation and CICS/400 and CICS on System/390 systems use EBCDIC[2]. 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 TS for Windows→CICS function shipping, CICS converts the user data (see Table 2 on page 43). For CICS→CICS TS for Windows function shipping, CICS TS for Windows 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. American National Standard Code for Information Interchange
2. Extended Binary-Coded Decimal Interchange Code

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. The definition could be statically-defined or autoinstalled.

The non-System/390 CICS requires a remote definition of the terminal. This remote terminal definition could be statically-defined to the non-System/390 CICS or shipped from CICS on System/390.

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

For transaction routing from CICS Transaction Server for Windows, 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.

CICS on System/390 requires a remote definition of the terminal. This remote terminal definition could be statically-defined to CICS on System/390 or shipped from the non-System/390 CICS.

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 architecture (ASCII-7) 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. [3]

Dynamic transaction routing

Dynamic transaction routing allows a user-written program (the “dynamic transaction routing program”) to select the system to which a transaction routing request is to be directed. Dynamic transaction routing is supported by CICS on System/390, CICS on Open Systems, and CICS Transaction Server for Windows, but not by CICS/400.

The terminal-owning region (TOR) that receives the transaction request and the application-owning region (AOR) to which the request is routed do not have to be

3. In a network of CICS Transaction Server for Windows 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.

the same CICS product-type. For example, it is possible for a dynamic transaction routing program running on a CICS on System/390 TOR to route a transaction request to a CICS/400 AOR.

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.

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 Transaction Server for Windows is not available in the CICS monitoring record for LU 6.2 links.
- CICS Transaction Server for Windows 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 “Distributed transaction processing (DTP)” on 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.
- 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
- SIGNON and SIGNOFF (CICS Transaction Server for z/OS and CICS Transaction Server for OS/390 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.

A CICS program linked by a program in a remote CICS system can issue commands that address the TWA or the TCTUA. However, if the linked-to program addresses the TWA it is given access to the TWA of the *local* transaction. If it addresses the TCTUA a null pointer is returned.

Note for DB2

When DB2® data is accessed from CICS Transaction Server for Windows, 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. For communication between all CICS servers, the recommended maximum length of a communications area is 32500 bytes.

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 and CICS Transaction Server for z/OS allow you to route DPL requests dynamically. In these products, 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 for OS/390 Release 3 and later 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 for OS/390 Release 3 or later systems. This means that other CICS products can benefit from the enhanced workload balancing capabilities of CICS TS for OS/390 Release 3 and later.

- Any type of program-link request received from outside CICS. For example, CICS Transaction Server for OS/390 Release 3 and later can dynamically route any of the following:
  - Calls received from:
    - CICS Web support
    - The CICS Transaction Gateway
  - 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.

For definitive information about how to route program-link requests dynamically, see your *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 Transaction Server for Windows 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 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 103.

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

summary

Chapter 2. Planning for CICS System/390–non-System/390 intercommunication

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 Transaction Server for Windows or CICS on Open Systems—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 Transaction Server for Windows 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.

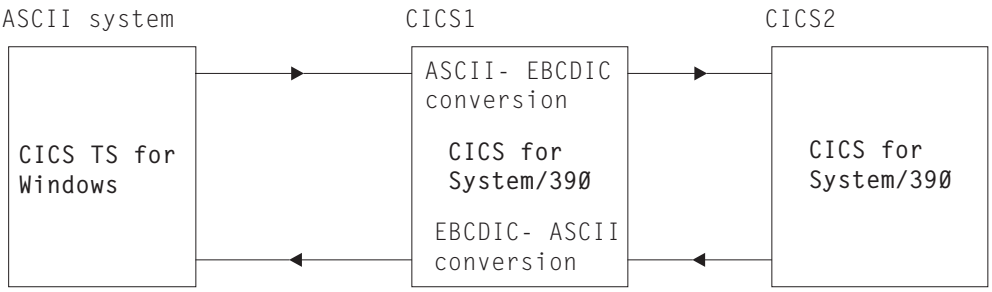


Figure 1. Where data conversion takes place

Figure 2 on page 12 shows an ASCII system, CICS Transaction Server for Windows 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:
   - 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.

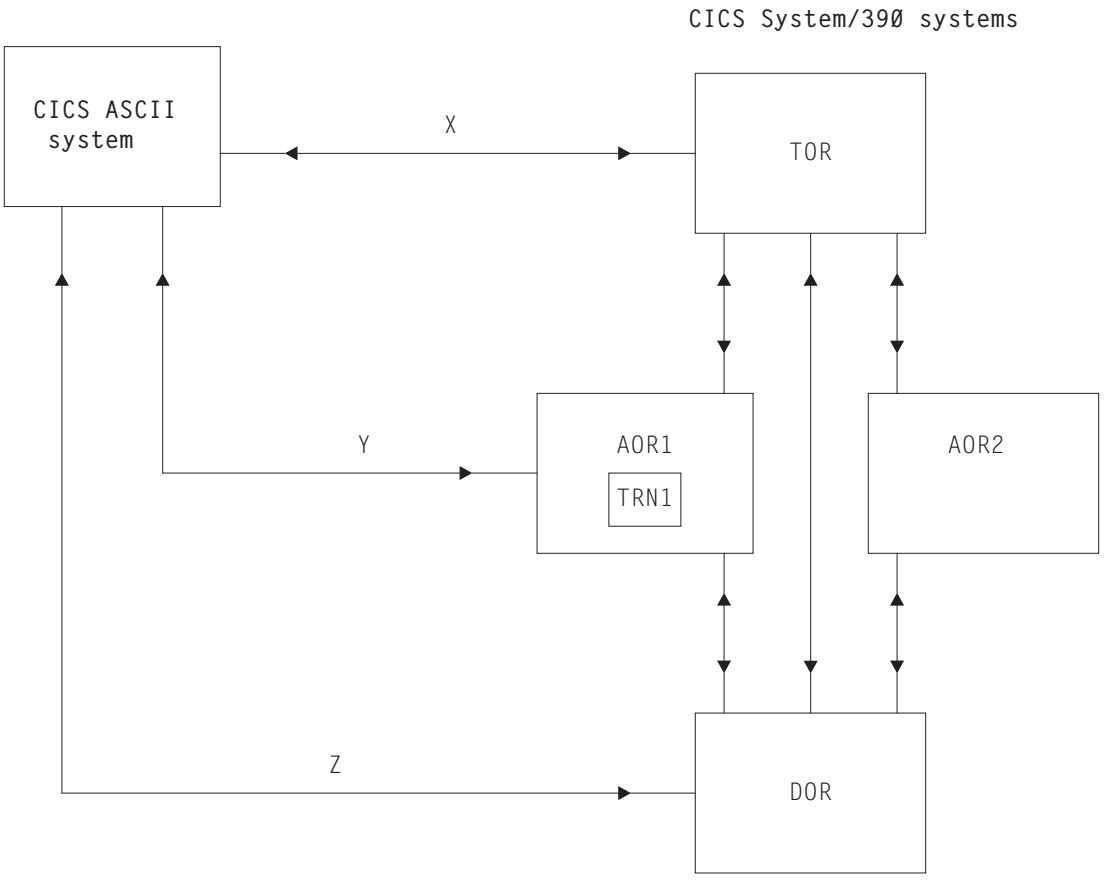


Figure 2. Sample configuration of System/390 and non-System/390 CICS systems

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:
1. The standard data conversion program
2. The data conversion table
3. The user-replaceable data conversion program.

All three of these items are required for function shipping and DPL, but only the first two for transaction routing.

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 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 uses its system services to perform the data conversion from ASCII to EBCDIC.

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.

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 and CICS/400. It is *not* supported by CICS Transaction Server for Windows. LU 6.2 sessions between CICS Transaction Server for Windows 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 Transaction Server for Windows

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 Transaction Server for Windows systems, X, Y, and Z.

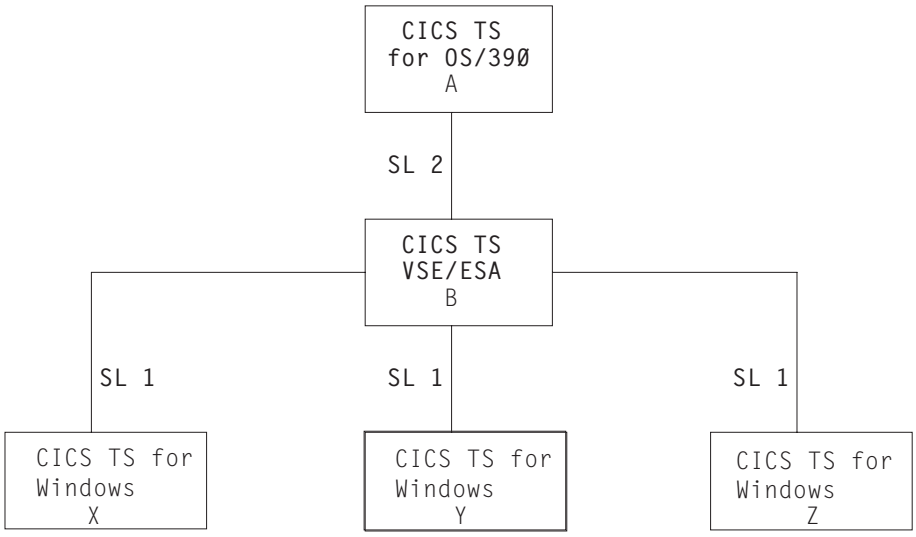


Figure 3. Synchronization between CICS on System/390 and CICS Transaction Server for Windows. CICS Transaction Server for Windows systems do not support synclevel 2 over LU 6.2 links.

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.

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 Transaction Server for Windows, 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 Transaction Server for Windows to CICS on System/390

DPL and function shipping incorporate synclevel 1 logic. The CICS Transaction Server for Windows system initiates the commit procedure by requesting CICS on System/390 to commit data changes. The CICS Transaction Server for Windows 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 Transaction Server for Windows system. If you take a syncpoint in the linked-to System/390 program, the syncpoint request is not propagated back to the CICS Transaction Server for Windows system.

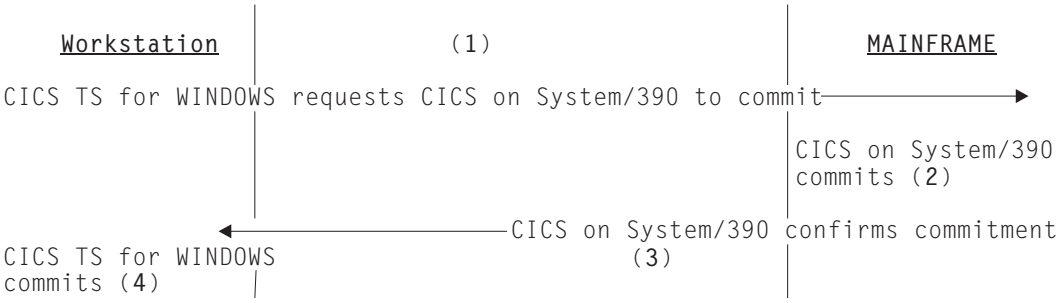


Figure 4. Sync level 1 logic in a CICS Transaction Server for Windows—CICS on System/390 link

Chapter 3. Resource definition for communication with non-System/390 systems

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* [4] 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 `CICSWIN`.

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

System generation and initialization

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 Transaction Server for Windows, CICS/400, and CICS on Open Systems, like CICS TS for z/OS, CICS TS for OS/390, and CICS Transaction Server for VSE/ESA, do not support macro-level programs).

For CICS/VSE Version 2 systems only, 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.

Setting up LU 6.2 links

You must use an LU 6.2 link for all the facilities described in Chapter 1, “Overview of CICS System/390–non-System/390 intercommunication,” 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 “Remote system TERMINAL definition (single session)” on page 23 and “Remote system TYPETERM definition” on page 25.
- For parallel sessions, CONNECTION and SESSIONS definitions, see “CONNECTION definition for parallel sessions” on page 26 and “SESSIONS definition for parallel sessions” on page 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 “CICS on System/390 mirror transaction definition” on page 20.
- The mirror program (the name varies with product and release); see “CICS on System/390 mirror program definition” on page 21.
- If the remote system is CICS on Open Systems or CICS Transaction Server for Windows, 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. Unless you can rely on a definition of this terminal being shipped from the non-System/390 system, you should create a remote definition of it to CICS on System/390.

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 transaction, unless the LINK or START command includes the SYSID option.

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 “Remote terminal TERMINAL definition” on 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(CICSWIN)
OVERTYPE TO MODIFY
 CEDA DEFine
  TRansaction    : CPMI
  Group          : CICSWIN
  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(CICSWIN)
OVERTYPE TO MODIFY
 CEDA DEFine
  PROGram      ==> DFHMIRS
  Group        ==> CICSWIN
  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 for z/OS, CICS TS for OS/390, or CICS Transaction Server for VSE/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.

Data conversion program definition

Enter all values as shown.

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

Data conversion table definition

Enter all values as shown.

```
DEFINE PROGRAM (DFHCNV)
OVERTYPE TO MODIFY
 CEDA DEFine
  PROGram      ==> DFHCNV
  Group        ==> CICSWIN
  Language     ==> Assembler          Cobol|Assembler|Le370|C|Pli|Rpg
  RELoad       ==> No                 No|Yes
  RESident     ==> No                 No|Yes
  RSl          ==> 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        ==> CICSWIN
  Language     ==> Assembler          Cobol|Assembler|Le370|C|Pli|Rpg
  RELoad       ==> No                 No|Yes
  RESident     ==> No                 No|Yes
  RSl          ==> 00                 0-24|Public
  Status       ==> Enabled            Enabled|Disabled
```

For all program definition screens, see “the notes on program definition screens” 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 Transaction Server for Windows, CICS/400, or CICS on Open Systems. This example is a definition of an LU 6.2 connection to CICS Transaction Server for Windows.

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.

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 “Remote system TYPETERM definition” on page 25.

```
DEFINE TERMINAL(PSO2) GROUP(CICSWIN)
OVERTYPE TO MODIFY
 CEDA  DEFine
  TErminal      ==> PSO2
  Group         ==> CICSWIN
  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
  OPERRsl       ==> 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.

Notes:

1. The OPERID, OPERPRIORITY, OPERRSL, and OPERSECURITY options are obsolete in CICS TS for z/OS, CICS TS for OS/390, and CICS Transaction Server for VSE/ESA, but are retained for compatibility with earlier releases.
2. The REMOTESYSNET option does not apply to CICS/VSE Version 2.
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 z/OS, CICS Transaction Server for OS/390 or CICS Transaction Server for VSE/ESA.

Remote system TYPETERM definition

This definition relates to the “Remote system TERMINAL definition (single session)” on page 23.

Most of the values below must be entered as shown.

```
DEFINE TYPETERM(APPC) GROUP(CICSWIN)
OVERTYPE TO MODIFY
 CEDA DEFine
  TYpeterm       : APPC
  Group          : CICSWIN
 RESOURCE TYPE
  DEVice       ==> APPC
  TERmmodel    ==>
  SESsiontype  ==>
  LDclist      ==>
  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
  ERRHilight   ==> No                 No|Blink|Reverse|Underline
 OPERATIONAL PROPERTIES
  AUTOConnect  ==> Yes                No|Yes|All
 RECOVERY
  RECOVOption  ==> Sysdefault         Sysdefault|Clearconv|
                                      Releasesess|Uncondrel|None
  RECOVNotify  ==> None               None|Message|Transaction
```

Note: In CICS TS for OS/390, CICS TS for z/OS, 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*.

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
  ACcessmethod ==> 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 ==>                   PASSWORD NOT SPECIFIED
  BINDSecurity ==> No                No|Yes
 RECOVERY
  Psrecovery   ==> Sysdefault        Sysdefault|None
```

Notes:

1. QUEUELIMIT and MAXQTIME do not apply to CICS/VSE Version 2. These options enable you to control the queuing of requests for free sessions on the connection. See your *Intercommunication Guide* for more details.
2. PSRECOVERY does not apply to CICS/VSE Version 2. With the RECOVOPTION attribute of DEFINE SESSIONS, it enables you to use VTAM persistent sessions on LU 6.2 links. See your *Intercommunication Guide* for more details.
3. BINDSECURITY is not applicable to CICS/VSE Version 2. BINDPASSWORD is not applicable to CICS Transaction Server for z/OS, 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     ==> CICSISC0
 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|Uncondrel|No
  RECOVNotify  ==> None              None|Message|Transaction
```

Note: In CICS TS for OS/390, CICS TS for z/OS, 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*.

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, 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 TS for OS/390, CICS TS for z/OS, 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(CICSWIN)
OVERTYPE TO MODIFY
 CEDA DEFine
  TErminal     ==> R123
  Group        ==> CICSWIN
  AUTINSTModel ==> No                 No|Yes|Only
 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
  OPERRsl      ==> 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 terminal definition screens” on page 23.

Remote terminal TYPETERM definition

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

```
DEFINE TYPETERM(LU62TR) GROUP(CICSWIN)
OVERTYPE TO MODIFY
CEDA DEFINE
  TYpeterm     ==> LU62TR
  Group        ==> CICSWIN
 RESOURCE TYPE
  DEVice       ==> 3270
  TERmmodel    ==> 2
  SESsiontype  ==>
  LDclist      ==>
  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|All
  ATi          ==> Yes                No|Yes
 RECOVERY
  RECOVOption  ==> Sysdefault         Sysdefault|Clearconv|
                                      Releasesess|Uncondrel|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 for z/OS, CICS TS for OS/390, and CICS Transaction Server for VSE/ESA do not support CICS internal security. CICS/VSE Version 2 supports either method.[5]

This chapter describes only CICS/VSE Version 2.3 internal security. CICS TS for OS/390 and CICS TS for z/OS users of RACF should refer to the *CICS-RACF Security Guide* for their System/390 product. CICS TS for OS/390 and CICS TS for z/OS 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 CONNECTION[6] 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.

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

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

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 CONNECTION[6] command.

Required specifications in remote systems

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

CICS Transaction Server for Windows specifications

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

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.[7]The level of security on a link depends on the ATTACHSEC option of the CEDA DEFINE CONNECTION[8] 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 Transaction Server for Windows, 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.

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

8. 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(CICSWIN)
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(CICSWIN)
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 Transaction Server for Windows—CICS/VSE links do not specify*** **BINDPASSWORD*.***

```
ALTER CONNECTION (APPC) GR(CICSWIN)
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(CICSWIN)
OVERTYPE TO MODIFY
 CEDA ALTer
 ................................
 ................................
 OPERRsl          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.

RSLC(YES)
specifies that security checking is required for resources accessed by this transaction.

```
ALTER TRANSACTION (RTED) GR(CICSWIN)
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 on page 37. 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 Transaction Server for Windows user with a CICS Transaction Server for Windows 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

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 Transaction Server for Windows, 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

```
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.

```
* 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=MODELU62,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=MODELU62,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=MODELU62,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=MODELU62,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=MODELU62,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=MODELU62,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

```
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'060200000000000000122F00'
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* LU62 REQUIRES SNASVCMG TO RUN PROPERLY for //ll. 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'060200000000000000122F00'
         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 for communication with non-System/390 systems

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.

Important

If you are using a **channel** to perform data conversion, read "Data conversion using channels" in the *CICS Application Programming Guide* instead of this topic. Channels are available in CICS Transaction Server for z/OS Version 3 Release 1 onwards.

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
TS	FROM area	As specified in DFHCNV table	Receiving system
TD	Queue name	Character	ASCII system
TD	INTO area	As specified in DFHCNV table	Receiving system
FC	File name	Character	ASCII system
FC	SET area	As specified in DFHCNV table	Receiving system
FC	Key	As specified in DFHCNV table	Receiving system
IC	Transaction ID	Character	ASCII system
IC	FROM area	As specified in DFHCNV table	Receiving system
IC	RTERMID, RTRANSID, REQID	Character	ASCII system
PC	Program name	Character	ASCII system
PC	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).

where conversion takes place

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

Conversion of application data is done field-by-field. Thus, ensure that the size of each field in the application data is sufficient to hold the result of the conversion applied to it. (This is particularly relevant where a field in the application data may contain both SBCS and DBCS characters).

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.

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 Transaction Server for Windows 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 apply nonstandard data conversion by writing your own version of the user-replaceable conversion program. 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.

For all CICS on System/390 products *other than* CICS TS for z/OS, Version 2.2 and later, to take advantage of nonstandard conversion you must provide a single data conversion program named DFHUCNV. This will probably be a customized version of the DFHUCNV program supplied with CICS.

For CICS TS for z/OS, Version 2.2 and later, you can provide *either*:

1. Your own, customized, version of DFHUCNV, *or*
2. One or more differently-named conversion programs

If the nonstandard conversion applies only to character data, you may not need to write your own data conversion program. Instead, you could create your own conversion tables for use with the standard conversion program, DFHCCNV. See “User-defined conversion tables” on page 75.

Important

Your user-supplied conversion program 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 conversion program must convert only fields defined as DATATYP=USERDATA (see “the DATATYP option of the DFHCNV TYPE=FIELD macro” on page 73).

Character data

Character data is described by a character set identifier and a code page identifier. The latter defines how each character is to be encoded; for example “A” is encoded as X'41' in ASCII and as X'C1' in EBCDIC.

The SRVERCP keyword on the DFHCNV TYPE=ENTRY macro determines the “server” code page in which character data associated with the specified resource is encoded in the System/390 server. Such data is assumed to be encoded in EBCDIC.

The CLINTCP keyword on the DFHCNV TYPE=ENTRY macro determines the default “client” code page in which the character data associated with the specified resource is encoded when it is received by or sent from the System/390 server. In general, such data is assumed to be encoded in ASCII. However, the data may be encoded in EBCDIC. (In this case, the client and server code pages are likely to be different, even though both are EBCDIC.)

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 Transaction Server for Windows and CICS on Open Systems, the default client code page must be set to the code page used by CICS Transaction Server for Windows. For example, specifying CLINTCP=932 allows CICS Transaction Server for Windows 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 75.

For nonstandard conversions, you must supply your own conversion program—see “User/CICS conversion” on page 60.

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:

CICS on System/390 is usually able to convert character data between ASCII and
EBCDIC if the client and server CCSIDs belong to the same group. However, there
are some limitations on conversions, even within the same group: for example,
when new CCSIDs are defined to extend the character set, conversions between
new equivalent ASCII and EBCDIC CCSIDs will be supported, but conversions
mixing old and new ASCII and EBCDIC CCSIDs may not.

Arabic

Baltic Rim
Latvia, Lithuania, Estonia

Cyrillic
Eastern Europe; Bulgaria, Russia, Yugoslavia

Devanagari (Hindi)
India

Farsi (Persian)
Iran

Greek
Greece

Hebrew
Israel

Japanese
Japan

Korean
Korea

Lao
Laos

Latin-1 and Latin-9
USA, Western Europe, and many other countries

Latin-2
Eastern Europe; Albania, Czech Republic, Hungary, Poland, Romania, Slovakia, Yugoslavia, Former Yugoslavia

Latin-5
Turkey

Simplified Chinese
Peoples’ Republic of China

Thai
Thailand

Traditional Chinese
Taiwan

Urdu
Pakistan

Vietnamese
Vietnam

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 current CICS on System/390 products that support the CCSID. Four levels of support are defined—"Base", "T01", "T02", and "T03".

Base
- CICS Transaction Server for z/OS (all releases)
- CICS Transaction Server for OS/390 Release 3
- CICS Transaction Server for VSE/ESA
- CICS/VSE Version 2 Release 3

T01
- CICS Transaction Server for z/OS (all releases)
- CICS Transaction Server for OS/390 Release 3
- CICS Transaction Server for VSE/ESA
- CICS/VSE Version 2 Release 3 plus APAR PQ19019

T02
- CICS Transaction Server for z/OS (all releases)

T03
- CICS Transaction Server for z/OS Version 3
- CICS Transaction Server for z/OS Version 2 Release 3

Arabic

Table 3. Arabic, Client CCSIDs

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
5352	T02	05352	01256	MS Windows: Arabic, version 2 with euro
9448	T03	09448	09448	MS Windows: Arabic, 2001
17248	T02	17248	00864	PC Data: Arabic with euro

Table 4. Arabic, Server CCSIDs

SRVERCP	in	CCSID	CPGID	Comments
420	Base	00420	00420	Host: Arabic
16804	T02	16804	00420	Host: Arabic with euro

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

Baltic Rim

Table 5. Baltic Rim, Client CCSIDs

	CLINTCP	in	CCSID	CPGID	Comments
	901	T02	00901	00901	PC data: Latvia, Lithuania; with euro
#	902	T02	00902	00902	PC data: Estonia with euro
	921	T01	00921	00921	PC data: Latvia, Lithuania
#	922	T01	00922	00922	PC data: Estonia
	1257	T01	01257	01257	MS Windows: Baltic Rim
	5353	T02	05353	01257	MS Windows: Baltic Rim, version 2 with euro

Table 6. Baltic Rim, Server CCSIDs

	SRVERCP	in	CCSID	CPGID	Comments
	1112	T01	01112	01112	Host: Latvia, Lithuania
#	1122	T01	01122	01122	Host: Estonia
	1156	T02	01156	01156	Host: Latvia, Lithuania; with euro
#	1157	T01	01157	01157	Host: Estonia;, with euro

Cyrillic

Table 7. Cyrillic, Client CCSIDs

	CLINTCP	in	CCSID	CPGID	Comments
	808	T02	00808	00808	PC data: Cyrillic, Russia; with euro
	848	T02	00848	00848	PC data: Cyrillic, Ukraine; with euro
	849	T02	00849	00849	PC data: Cyrillic, Belarus; with euro
#	855	Base	00855	00855	PC data: Cyrillic
	866	Base	00866	00866	PC data: Cyrillic, Russia
	872	T02	00872	00872	PC data: Cyrillic with euro
	915 8859-5	Base	00915	00915	ISO 8859-5: Cyrillic
	1124	T02	01124	01124	8-bit: Cyrillic, Belarus
	1125	T02	01125	01125	PC Data: Cyrillic, Ukraine
	1131	T02	01131	01131	PC Data: Cyrillic, Belarus
	1251	T01	01251	01251	MS Windows: Cyrillic
	5347	T02	05347	01251	MS Windows: Cyrillic, version 2 with euro

Table 8. Cyrillic, Server CCSIDs

SRVERCP	in	CCSID	CPGID	Comments
1025	Base	01025	01025	Host: Cyrillic multilingual
1123	T02	01123	01123	Host: Cyrillic Ukraine
1154	T02	01154	01154	Host: Cyrillic multilingual; with euro
1158	T02	01158	01158	Host: Cyrillic Ukraine; wtih euro

Devanagari

Table 9. Devanagari, Client CCSIDs

CLINTCP	in	CCSID	CPGID	Comments
806	T03	00806	00806	PC data: ISCII-91, Devanagari script code

Table 10. Devanagari, Server CCSIDs

SRVERCP	in	CCSID	CPGID	Comments
1137	T03	01137	01137	Host: Devanagari

Note: These Devanagari CCSIDs may also be used to encode the identical Devanagari character repertoire used by Marathi.

Farsi

Table 11. Farsi, Client CCSIDs

CLINTCP	in	CCSID	CPGID	Comments
1098	T03	01098	01098	PC data: Farsi

Table 12. Farsi, Server CCSIDs

SRVERCP	in	CCSID	CPGID	Comments
1097	T03	01097	01097	Host: Farsi

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

Greek

Table 13. Greek, Client CCSIDs

CLINTCP	in	CCSID	CPGID	Comments
813 8859-7	Base	00813	00813	ISO 8859-7: Greece
869	Base	00869	00869	PC data: Greece
1253	T01	01253	01253	MS Windows: Greece
4909	T02	04909	00813	ISO 8859-7: Greece with euro
5349	T02	05349	01253	MS Windows: Greece, version 2 with euro
9061	T02	09061	00869	PC Data: Greece with euro

Table 14. Greek, Server CCSIDs

SRVERCP	in	CCSID	CPGID	Comments
875	Base	00875	00875	Host: Greece
4971	T02	04971	00875	Host: Greece with euro

Hebrew

Table 15. Hebrew, Client CCSIDs

CLINTCP	in	CCSID	CPGID	Comments
856	Base	00856	00856	PC data: Hebrew
862	T02	00862	00862	PC data: Hebrew (migration)
867	T02	00867	00867	PC Data: Hebrew with euro
916 8859-8	Base	00916	00916	ISO 8859-8: Hebrew
1255	T01	01255	01255	MS Windows: Hebrew
5351	T02	05351	01255	MS Windows: Hebrew, version 2 with euro
9447	T03	09447	01255	MS Windows: Hebrew, version 2 with euro and new sheqel

Table 16. Hebrew, Server CCSIDs

SRVERCP	in	CCSID	CPGID	Comments
424	Base	00424	00424	Host: Hebrew
803	T02	00803	00803	Host: Hebrew (Character Set A)
4899	T02	04899	00803	Host: Hebrew (Character Set A) with euro
12712	T02	12712	00424	Host: Hebrew with euro and new sheqel

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

Japanese

Table 17. Japanese, Client CCSIDs

CLINTCP	in	CCSID	CPGID	Comments
932	Base	00932	1. 00897 2. 00301	1. PC data: SBCS 2. PC data: DBCS including 1880 user-defined characters
942	Base	00942	1. 01041 2. 00301	1. PC data: Extended SBCS 2. PC data: DBCS including 1880 user-defined characters
943	T01	00943	1. 00897 2. 00941	1. PC data: SBCS 2. PC data: DBCS for Open environment including 1880 IBM® user-defined characters
954 EUCJP	Base	00954	1. 00895 2. 00952 3. 00896 4. 00953	1. G0: JIS X201 Roman 2. G1: JIS X208-1990 3. G1: JIS X201 Katakana 4. G1: JIS X212
5050	T02	05050	1. 00895 2. 00952 3. 00896 4. 00953	1. G0: JIS X201 Roman 2. G1: JIS X208-1990 3. G1: JIS X201 Katakana 4. G1: JIS X212

Table 18. Japanese, Server CCSIDs

SRVERCP		CCSID	CPGID	Comments
930	Base	00930	1. 00290 2. 00300 3. 00290 4. 00300	1. Katakana Host: extended SBCS 2. Kanji Host: DBCS including 4370 user-defined characters 3. Katakana Host: extended SBCS 4. Kanji Host: DBCS including 1880 user-defined characters
931	Base	00931	1. 00037 2. 00300	1. Latin Host: SBCS 2. Kanji Host: DBCS including 4370 user-defined characters
939	Base	00939	1. 01027 2. 00300 3. 01027 4. 00300	1. Latin Host: extended SBCS 2. Kanji Host: DBCS including 4370 user-defined characters 3. Latin Host: extended SBCS 4. Kanji Host: DBCS including 1880 user-defined characters
1390	T02	01390	1. 00290 2. 00300	1. Katakana Host: extended SBCS; with euro 2. Kanji Host: DBCS including 6205 user-defined characters
1399	T02	01399	1. 01027 2. 00300	1. Latin Host: extended SBCS; with euro 2. Kanji Host: DBCS including 4370 user-defined characters; with euro

Korean

Table 19. Korean, Client CCSIDs

CLINTCP	in	CCSID	CPGID	Comments
934	Base	00934	1. 00891 2. 00926	1. PC data: SBCS 2. PC data: DBCS including 1880 user-defined characters
944	Base	00944	1. 01040 2. 00926	1. PC data: Extended SBCS 2. PC data: DBCS including 1880 user-defined characters
949	Base	00949	1. 01088 2. 00951	1. IBM KS Code - PC data: SBCS 2. IBM KS code - PC data: DBCS including 1880 user-defined characters
970 EUCKR	Base	00970	1. 00367 2. 00971	1. G0: ASCII 2. G1: KSC X5601-1989 including 1880 user-defined characters
1363	T01	01363	1. 01126 2. 01362	1. PC data: MS Windows Korean SBCS 2. PC data: MS Windows Koran DBCS including 11172 full Hangul

Table 20. Korean, Server CCSIDs

SRVERCP	in	CCSID	CPGID	Comments
933	Base	00933	1. 00833 2. 00834	1. Host: Extended SBCS 2. Host: DBCS including 1880 user-defined characters and 11172 full Hangul characters
1364	T02	01364	1. 00833 2. 00834	1. Host: Extended SBCS 2. Host: DBCS including 1880 user-defined characters and 11172 full Hangul characters

Lao

Table 21. Lao, Client CCSIDs

CLINTCP	in	CCSID	CPGID	Comments
1133	T03	01133	01133	ISO-8: Lao

Table 22. Lao, Server CCSIDs

SRVERCP	in	CCSID	CPGID	Comments
1132	T03	01132	01132	Host: Lao

Latin-1 and Latin-9

Table 23. Latin-1, Client CCSIDs

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
858	T01	00858	00858	PC data: Latin-1 countries; with euro
923	T01	00923	00923	ISO 8859-15: Latin-9
924	T02	00924	00924	ISO 8859-15: Latin-9
1047	T02	01047	01047	Host: Latin-1
1252	T01	01252	01252	MS Windows: Latin-1 countries
5348	T01	05348	01252	MS Windows: Latin-1 countries, version 2 with euro

Table 24. Latin-1 and Latin-9, Server CCSIDs

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
924	T01	00924	00924	Host: Latin-9
1047	T01	01047	01047	Host: Latin-1
1140	T01	01140	01140	Host: USA, Canada (ESA), Netherlands, Portugal, Brazil, Australia, New Zealand; with euro
1141	T01	01141	01141	Host: Austria, Germany; with euro
1142	T01	01142	01142	Host: Denmark, Norway; with euro
1143	T01	01143	01143	Host: Finland, Sweden; with euro
1144	T01	01144	01144	Host: Italy; with euro
1145	T01	01145	01145	Host: Spain, Latin America (Spanish); with euro
1146	T01	01146	01146	Host: United Kingdom; with euro
1147	T01	01147	01147	Host: France; with euro
1148	T01	01148	01148	Host: Belgium, Canada (AS/400), Switzerland, International Latin-1; with euro
1149	T01	01149	01149	Host: Iceland; with euro

Note: Conversions are supported between non euro-supported CCSIDs and euro-supported CCSIDs. These should be used with care because:

- The international currency symbol in each non euro-supported EBCDIC CCSID (for example, 00500) has been replaced by the euro symbol in the equivalent euro-supported EBCDIC CCSID (for example, 01148).
- The dotless *i* in non euro-supported ASCII CCSID 00850 has been replaced by the euro symbol in the equivalent euro-supported ASCII CCSID 00858.

Latin-2

Table 25. Latin-2, Client CCSIDs

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
5346	T02	05346	01250	MS Windows: Latin-2, version 2 with euro
9044	T02	09044	00852	PC data: Latin-2 multilingual with euro

Table 26. Latin-2, Server CCSIDs

	SRVERCP	in	CCSID	CPGID	Comments
	500	T02	00500	00500	Host: International Latin-1
	870	Base	00870	00870	Host: Latin-2 multilingual
#	924	T01	00924	00924	Host: Latin-9
# #	1140	T01	01140	01140	Host: USA, Canada (ESA), Netherlands, Portugal, Brazil, Australia, New Zealand; with euro
#	1141	T01	01141	01141	Host: Austria, Germany; with euro
#	1142	T01	01142	01142	Host: Denmark, Norway; with euro
#	1143	T01	01143	01143	Host: Finland, Sweden; with euro
#	1144	T01	01144	01144	Host: Italy; with euro
#	1145	T01	01145	01145	Host: Spain, Latin America (Spanish); with euro
#	1146	T01	01146	01146	Host: United Kingdom; with euro
#	1147	T01	01147	01147	Host: France; with euro
	1148	T02	01148	01148	Host: International Latin-1 with euro
#	1149	T01	01149	01149	Host: Iceland; with euro
	1153	T02	01153	01153	Host: Latin-2 multilingual with euro

Note: Conversions are supported for some combinations of Latin-2 ASCII CCSIDs and Latin-1 EBCDIC CCSIDs.

Latin-5

Table 27. Latin-5, Client CCSIDs

CLINTCP	in	CCSID	CPGID	Comments
857	Base	00857	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
5350	T02	05350	01254	MS Windows: Turkey, version 2 with euro
9049	T02	09049	00857	PC data: Latin-5 (Turkey) with euro

Table 28. Latin-5, Server CCSIDs

SRVERCP	in	CCSID	CPGID	Comments
1026	Base	01026	01026	Host: Latin-5 (Turkey)
1155	T02	01155	01155	Host: Latin-5 (Turkey) with euro

Simplified Chinese

Table 29. Simplified Chinese, Client CCSIDs

CLINTCP	in	CCSID	CPGID	Comments
946	Base	00946	1. 01042 2. 00928	1. PC data: Extended SBCS 2. PC data: DBCS including 1880 user-defined characters
1381	Base	01381	1. 01115 2. 01380	1. PC data: Extended SBCS (IBM GB) 2. PC data: DBCS (IBM GB) including 31 IBM-selected, 1880 user-defined characters
1383 EUCCN	T01	01383	1. 00367 2. 01382	1. G0: ASCII 2. G1: GB 2312-80 set
1386	T01	01386	1. 01114 2. 01385	1. PC data: S-Chinese GBK and T-Chinese IBM BIG-5 2. PC data: S-Chinese GBK
5488	T03	05488	1. 01252 2. 01385 3. 01391	1. GB18030, 1-byte data 2. GB18030, 2-byte data 3. GB18030, 4-byte data

Table 30. Simplified Chinese, Server CCSIDs

SRVERCP	in	CCSID	CPGID	Comments
935	Base	00935	1. 00836 2. 00837	1. Host: Extended SBCS 2. Host: DBCS including 1880 user-defined characters
1388	T02	01388	1. 00836 2. 00837	1. Host: Extended SBCS 2. Host: DBCS including 1880 user-defined characters
9127	T02	09127	1. 00836 2. 00837	1. Host: Extended SBCS 2. Host: DBCS including 1880 user-defined characters

Thai

Table 31. Thai, Client CCSIDs

CLINTCP	in	CCSID	CPGID	Comments
1161	T03	01161	01161	PC data: Thai with euro
1162	T03	01162	01162	MS Windows: Thai with euro
9066	T03	09066	00874	PC data: Thai extended SBCS

Table 32. Thai, Server CCSIDs

SRVERCP	in	CCSID	CPGID	Comments
1160	T03	01160	01160	Host: Thai with euro
9030	T03	09030	00838	Host: Thai extended SBCS

Traditional Chinese

Table 33. Traditional Chinese, Client CCSIDs

CLINTCP	in	CCSID	CPGID	Comments
938	Base	00938	1. 00904 2. 00927	1. PC data: SBCS 2. PC data: DBCS including 6204 user-defined characters
948	Base	00948	1. 01043 2. 00927	1. PC data: Extended SBCS 2. PC data: DBCS including 6204 user-defined characters
950 BIG5	Base	00950	1. 01114 2. 00947	1. PC data: SBCS (IBM BIG5) 2. PC data: DBCS including 13493 CNS, 566 IBM selected, 6204 user-defined characters
964 EUCTW	Base	00964	1. 00367 2. 00960 3. 00961	1. G0: ASCII 2. G1: CNS 11643 plane 1 3. G1: CNS 11643 plane 2
1370	T02	01370	1. 01114 2. 00947	1. PC data: Extended SBCS; with euro 2. PC data: DBCS including 6204 user-defined characters; with euro

Table 34. Traditional Chinese, Server CCSIDs

SRVERCP	in	CCSID	CPGID	Comments
937	Base	00937	1. 00037 2. 00835	1. Host: Extended SBCS 2. Host: DBCS including 6204 user-defined characters
1371	T02	01371	1. 01159 2. 00835	1. Host: Extended SBCS; with euro 2. Host: DBCS including 6204 user-defined characters; with euro

Urdu

Table 35. Urdu, Client CCSIDs

CLINTCP	in	CCSID	CPGID	Comments
868	T03	00868	00868	PC data: Urdu
1006	T03	01006	01006	ISO-8: Urdu

Table 36. Urdu, Server CCSIDs

SRVERCP	in	CCSID	CPGID	Comments
918	T03	00918	00918	Host: Urdu

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

Vietnamese

Table 37. Vietnamese, Client CCSIDs

CLINTCP	in	CCSID	CPGID	Comments
1129	T02	01129	01129	ISO-8: Vietnamese
1163	T02	01163	01163	ISO-8: Vietnamese with euro
1258	T02	01258	01258	MS Windows: Vietnamese
5354	T02	05354	01258	MS Windows: Vietnamese, version 2 with euro

Table 38. Vietnamese, Server CCSIDs

SRVERCP	in	CCSID	CPGID	Comments
1130	T02	01130	01130	Host: Vietnamese
1164	T02	01164	01164	Host: Vietnamese with euro

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.

Note: More extensive support for conversion to and from Unicode data is available in CICS if you use channels to communicate your data. See the *CICS Application Programming Guide*. Channels are available in CICS Transaction Server for z/OS Version 3 Release 1 onwards.

Table 39. Unicode

CLINTCP SRVERCP	in	CCSID	CPGID	Comments
1200 UCS-2	T01	01200	01400	Unicode with character set 65535 (the “growing” character set, see Note beneath table). In the absence of a byte-order mark (BOM), assumed to be UTF-16 BE (big-endian).
1208 UTF-8	T01	01208	01400	Unicode with character set 65535 (the “growing” character set, see Note beneath table). UTF-8.

Table 39. Unicode (continued)

CLINTCP SRVERCP	in	CCSID	CPGID	Comments
13488	T01	13488	01400	Unicode with character set 3001 (fixed at Unicode 2.0 character repertoire). In the absence of a byte-order mark, assumed to be UTF16-BE (big-endian).
17584		17584	01400	Unicode with character set 3004 (fixed at Unicode 3.0 character repertoire). in the absence of a byte-order mark, assumed to be UTF16-BE (big-endian).

Note: Character set 65535 is known as the "growing character set" because it allows more characters to be added to the set from time to time. This allows a product that supports Unicode to avoid having to change the CCSID value every time more characters are added to Unicode.

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 big-endian; that is, multibyte numerical values have the most significant byte values first (in the lower machine address).

DATATYP=NUMERIC
The default format for binary data received by CICS on System/390 is little-endian; that is, multibyte numerical values have the least significant byte values first.

The default binary format can be overridden. It is therefore important that you code a DFHCNV TYPE=FIELD macro for every binary field. If the resource can be accessed from CICS Transaction Server for Windows, you must set the default binary format to that used by CICS Transaction Server for Windows.

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 in "Defining the conversion table" on page 63.

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.

The user-replaceable conversion program, DFHUCNV
A user-replaceable program that allows you to override the standard conversions applied by CICS. You can use it to apply your own conversion logic to specific data fields. (How to do this is described in "User/CICS conversion" on page 60.)

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

For all CICS on System/390 products *other than* CICS TS for z/OS, Version 2.2 and later, to take advantage of nonstandard conversion you must provide a single data conversion program named DFHUCNV. This will probably be a customized version of the DFHUCNV program supplied with CICS.

For CICS TS for z/OS, Version 2.2 and later, you can provide *either*:

1. Your own, customized, version of DFHUCNV, *or*
2. One or more differently-named conversion programs

In the following sections, the generic term "DFHUCNV" represents both the (possibly customized) IBM-supplied conversion program and user-named conversion programs.

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 Transaction Server for Windows 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 Transaction Server for Windows or CICS on Open Systems.

The translation tables shipped with CICS 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 little-endian to big-endian format, on receipt of a request from CICS Transaction Server for Windows.
- From big-endian to little-endian format, before transmitting the response to CICS Transaction Server for Windows.

In CICS Transaction Server for Windows, 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 "the DATATYP option of the DFHCNV TYPE=FIELD macro" on page 73).

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 63. This enables DFHCCNV to handle the resource.
2. Specify USREXIT=NO 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 or (for CICS TS for z/OS Version 2.2 and later) USREXIT=*program* on the DFHCNV TYPE=ENTRY macro that defines the resource. If USREXIT=YES is specified, DFHUCNV is called. If USREXIT=*program* is specified, your conversion program named *program* 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 "the USRTYPE option of the DFHCNV TYPE=FIELD macro" on page 73). 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 or (for CICS TS for z/OS Version 2.2 and later) a differently-named conversion program to handle the nonstandard fields.

   "The user-replaceable conversion program" on page 82 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 or (for CICS TS for z/OS Version 2.2 and later) USREXIT=*program* on the DFHCNV TYPE=ENTRY macro that defines the resource. If USREXIT=YES is specified, DFHUCNV is called. If USREXIT=*program* is specified, your conversion program named *program* is called.
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 or (for CICS TS for z/OS Version 2.2 and later) a differently-named conversion program 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 83.
- It relies on the following:
  - That all workstations connected to the CICS on System/390 server share a common encoding (for example, 437) for character data, and a common format (for example, little-endian) for binary data
  - That your version of DFHUCNV knows the ASCII encoding and binary format, and knows the corresponding EBCDIC encoding
  - That DFHUCNV provides the appropriate ASCII/EBCDIC 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 83.

   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.

   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 Transaction Server for Windows should not use the TRANSID option to specify a transaction other than the default CPMI, which is required to trigger conversion.

Figure 12 summarizes System/390 conversion procedures.

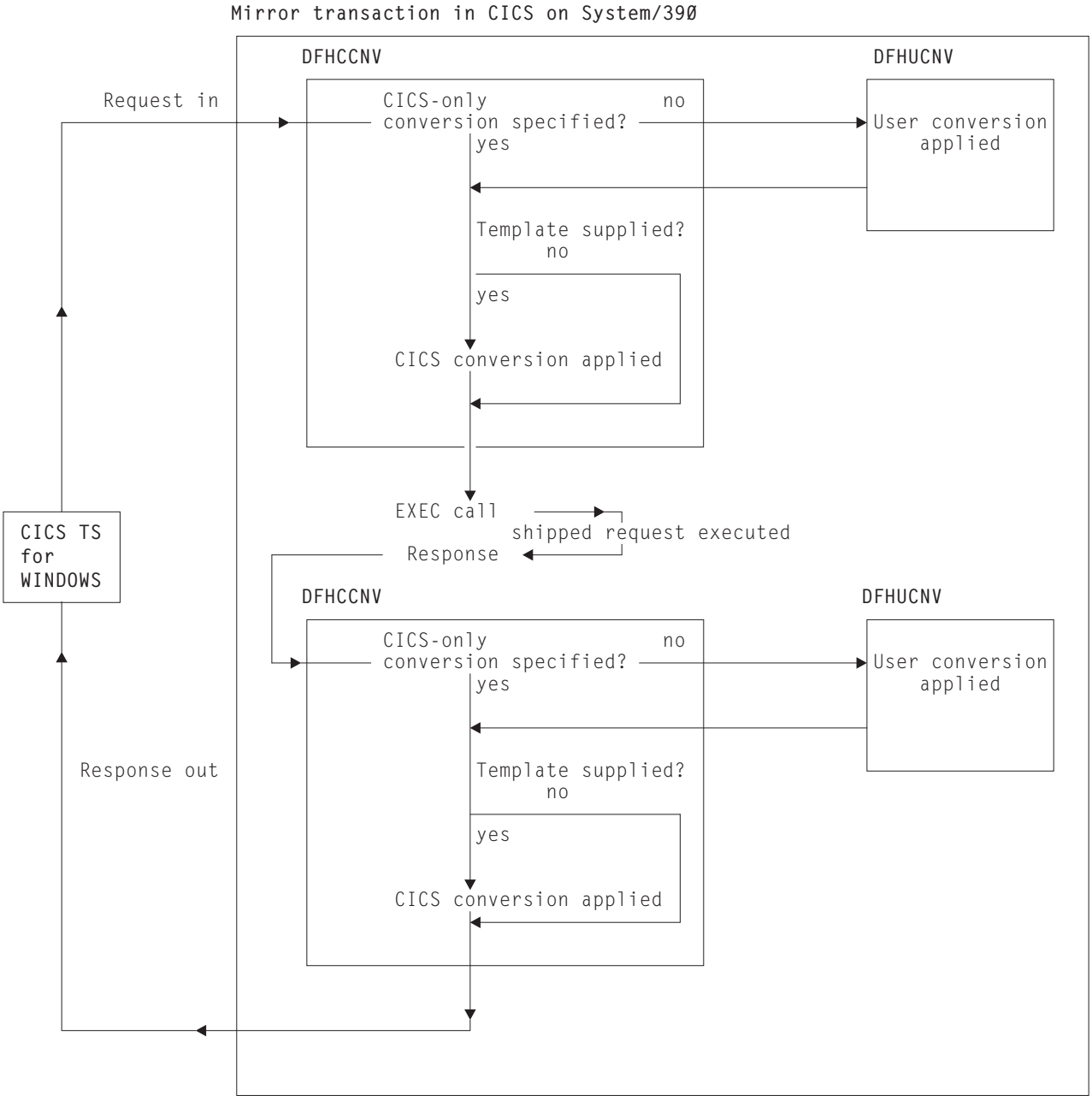


Figure 12. Data conversion in CICS on System/390

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, which are described below.

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.

If you are running CICS on System/390 in a z/OS environment, see “Assembling and link-editing the conversion programs” on page 81.

DFHCNV macro types

The following macros define the conversion table:

- DFHCNV TYPE=INITIAL defines the beginning of the conversion table. It defines the default client and server CCSIDs.
- 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 CCSID to be used is specified.

CICS TS for z/OS Version 2.3 and later

You can create generic templates that apply to multiple resources of the same resource type. You do this by using the RPFX or XRPFX parameters of the DFHCNV TYPE=ENTRY macro to specify a prefix that can be matched against multiple resource names, rather than using the full name of a specific resource.

Defining resources in this way means that order becomes important in the conversion table. For example, when specifying file resources, if prefix AB precedes prefix ABCD, the former entry is used to convert data for a file resource named ABCDEFGH. This example would give you an error when assembling the conversion table. To avoid errors, you should put the most specific resource names at the top of the conversion table, with the least specific prefix at the bottom.

When no resource name or prefix is specified, the default conversion template is used for that particular resource type. For an example of the DFHCNV TYPE=ENTRY macro, see “DFHCNV TYPE=ENTRY” on page 68.

- 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 [9] 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.

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 66 shows templates within a complete conversion table definition.

Defaults for client and server code pages

This section applies only to CICS Transaction Server for z/OS Version 3 Release 1 and later.

Certain distributed components of a CICSplex such as CICS Transaction Gateway for z/OS and CICS Transaction Server for Windows do not provide an override for the default client code page specified in the conversion table. As conversion tables do not have a suffix, a consequence of this is that two tables can be required, each residing on a different library and differing only in the default code page.

In order to reduce the number of conversion tables required, you can specify that the default client or server code page is defined in the system initialization table. For the client code page:

1. In the DFHCNV TYPE=ENTRY and TYPE=SELECT macros, specify the value SYSDEF for the CLINTCP parameter.
2. In the system initialization table, set a default client code page by specifying a value for the CLINTCP parameter. You can use any value supported for the CLINTCP parameter on the DFHCNV macro. The default is CLINTCP=437.

For the server code page:

1. In the DFHCNV TYPE=ENTRY and TYPE=SELECT macros, specify the value SYSDEF for the SRVERCP parameter.

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

2. In the system initialization table, set a server code page by specifying a value for the SRVERCP parameter. You can use any value supported for SRVERCP parameter on the DFHCNV macro. The default is SRVERCP=037.

Example sequence—DFHCNV macros

Figure 14 on page 66 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

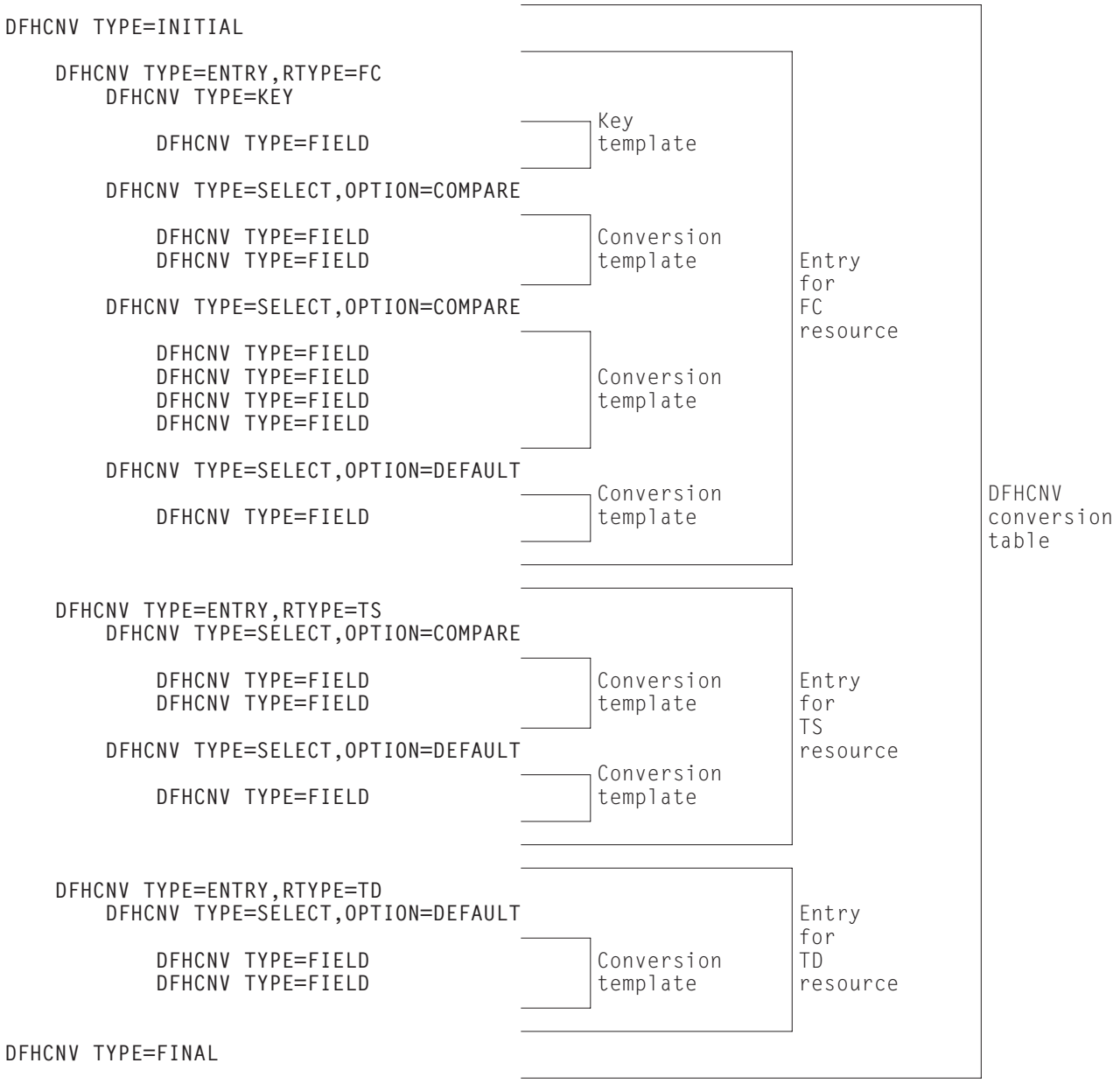
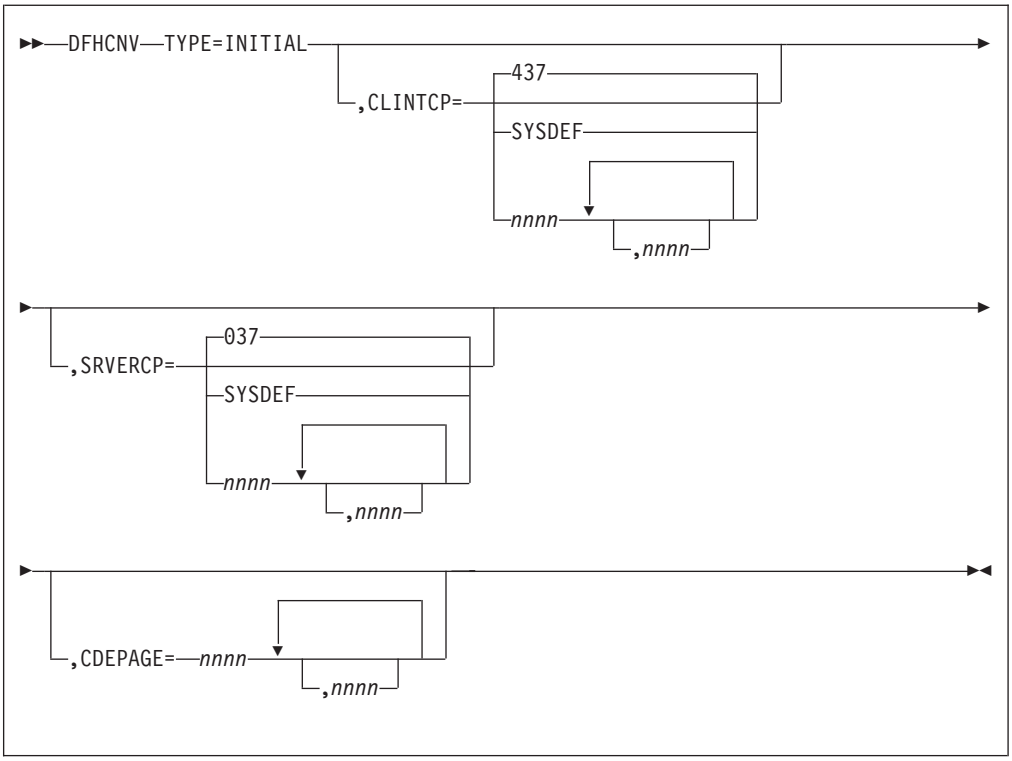


Figure 14. Example of DFHCNV macro sequence

DFHCNV TYPE=INITIAL

The DFHCNV TYPE=INITIAL macro has the following format:



TYPE=INITIAL
Defines the beginning of the conversion table.

CLINTCP={437|SYSDEF|nnnn[,nnnn, ...]}
The first operand defines the default client CCSID to be used when the CLINTCP and CDEPAGE operands are omitted from a DFHCNV TYPE=ENTRY macro.

SYSDEF specifies that the default client code page is determined by the system initialization table parameter CLINTCP. SYSDEF applies only to CICS Transaction Server for z/OS Version 3 Release 1 and later.

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

SRVERCP={037|SYSDEF|nnnn[,nnnn, ...]}
The first operand defines the server CCSID to be used when the SRVERCP and CDEPAGE operands are omitted from a DFHCNV TYPE=ENTRY macro.

SYSDEF specifies that the default server code page is determined by the system initialization table parameter SRVERCP. SYSDEF applies only to CICS Transaction Server for z/OS Version 3 Release 1 and later

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.*)

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

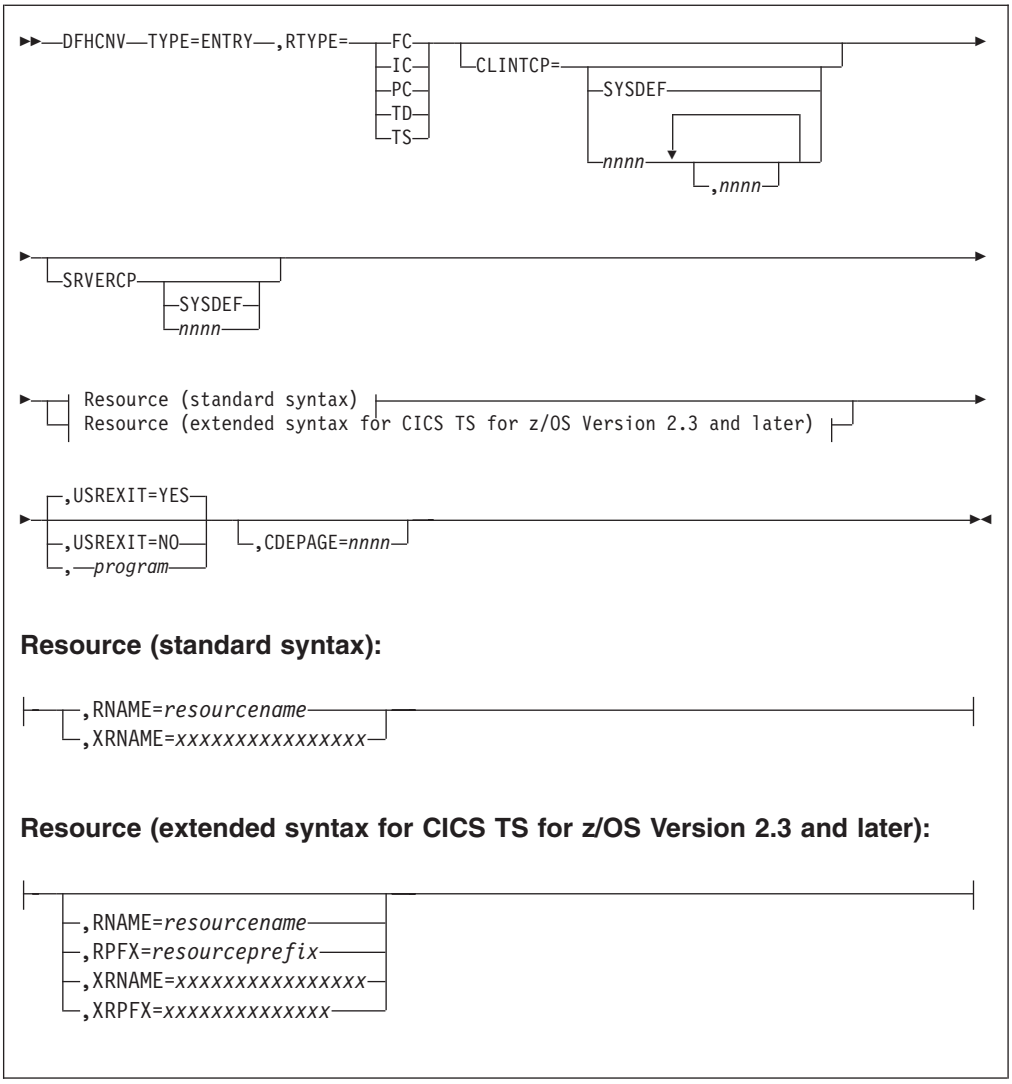
932
Is equivalent to:
CLINTCP=932
SRVERCP=931

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:



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 data
PC A program link with a COMMAREA.

CLINTCP={nnnn[,nnnn, ...]|SYSDEF}
The first operand defines the default client code page to be used.

SYSDEF specifies that the default client code page is determined by the system initialization table parameter CLINTCP. SYSDEF applies only to CICS Transaction Server for z/OS Version 3 Release 1 and later.

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

SRVERCP={nnnn|SYSDEF}
The operand defines the server code page to be used.

SYSDEF specifies that the server code page is determined by the system initialization table parameter SRVERCP. SYSDEF applies only to CICS Transaction Server for z/OS Version 3 Release 1 and later.

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).

  Note: Although CICS TS for OS/390 Release 3 and above support TS queue names of up to16 characters, DFHCNV only supports TS queue names of up to 8 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).

RPFX=resourceprefix
Applies only to CICS TS for z/OS Version 2.3 and later. Specifies a resource prefix of up to 7 characters for programs, TS queues and files; or 3 characters for TD queues and transactions. The resource prefix allows resources of a particular type to be grouped together using just one macro. All resources of the specified type and prefix will be treated in the same way. Order is important, so the most specific resource names should be at the top of the converstion table, with the least specific prefixes at the bottom. If none of the parameters are specified at this point in the macro, the default template is used for all resources within the specified resource type.

XRNAME=xxxxxxxxxxxxxxxx (RTYPE=TS only)
Specifies the resource name in hexadecimal notation. It can include up to 16 hexadecimal digits, padded with blanks if necessary.

XRPFX=xxxxxxxxxxxxxx (RTYPE=TS only)
Applies only to CICS TS for z/OS Version 2.3 and later. Specifies a resource prefix of up to 14 hexadecimal digits. The resource prefix allows resources of a particular type to be grouped together. All resources of the specified type and prefix will be treated in the same way. Order is important, so the most specific resource names should be at the top of the conversion table, with the least specific prefixes at the bottom.If none of the parameters are specified at this point in the macro, the default template is used for all resources within the specified resource type.

USREXIT={YES|NO|*program*}
Specifies whether the user data conversion exit is called.

YES
User-defined conversion is required for this resource. DFHUCNV is invoked. Code this if you need your customized version of DFHUCNV to convert some data for this resource.

NO
User-defined conversion is not required for this resource. The user-replaceable conversion program is not called. Code this to eliminate the overhead of calling the program unnecessarily.

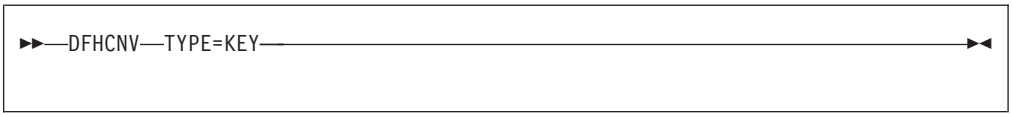
program **(applies only to CICS TS for z/OS Version 2.2 and later)**
User-defined conversion is required for this resource; *program* is invoked. Code this if you need your user-supplied program, *program*, to convert some data for this resource.

CDEPAGE=nnnn
(*Not for new definitions. Retained for compatibility with earlier releases.*)

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

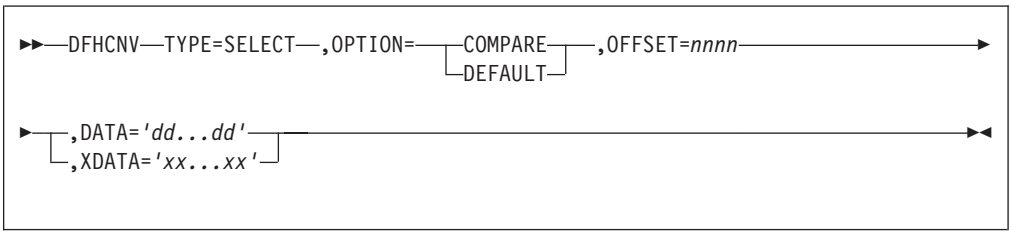
The DFHCNV TYPE=KEY macro is valid only for FC RTYPE requests, and, if coded, must immediately follow a DFHCNV TYPE=ENTRY macro. The macro has the following format:



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:



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*)

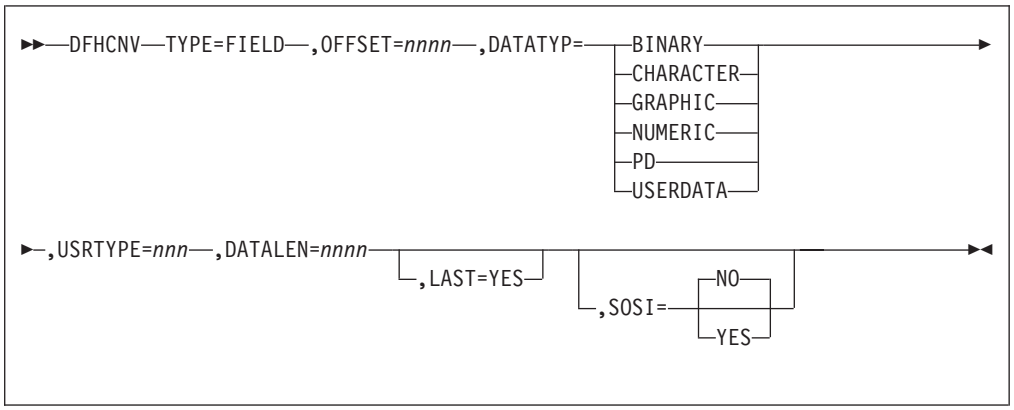
Specifies the comparison data as a character string, with a maximum length of
255 characters. Because you specify this data on the System/390 system, it is in EBCDIC. The system converts the incoming data from ASCII to EBCDIC before checking it against the comparison data, so that EBCDIC is compared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

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:



TYPE=FIELD
Specifies conversion specifications for a data field. There must be one such
statement for each field in a record. You cannot code a TYPE=FIELD macro
until you have coded 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 Transaction Server for Windows 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 big-endian format.

The CICS Transaction Server for Windows user can specify the IBMCOMP directive with the CICSCOMP command used to compile a CICS Transaction Server for Windows COBOL/2 application program that creates
binary fields in big-endian-compatible format.

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

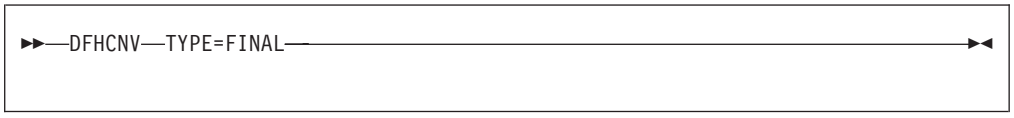
If DATATYP=NUMERIC, DATALEN must be 2 or 4.

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

SOSI=YES|NO
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.

DFHCNV TYPE=FINAL

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



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 Transaction Server for Windows 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 Transaction Server for Windows users.
5. CICS Transaction Server for Windows assumes 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

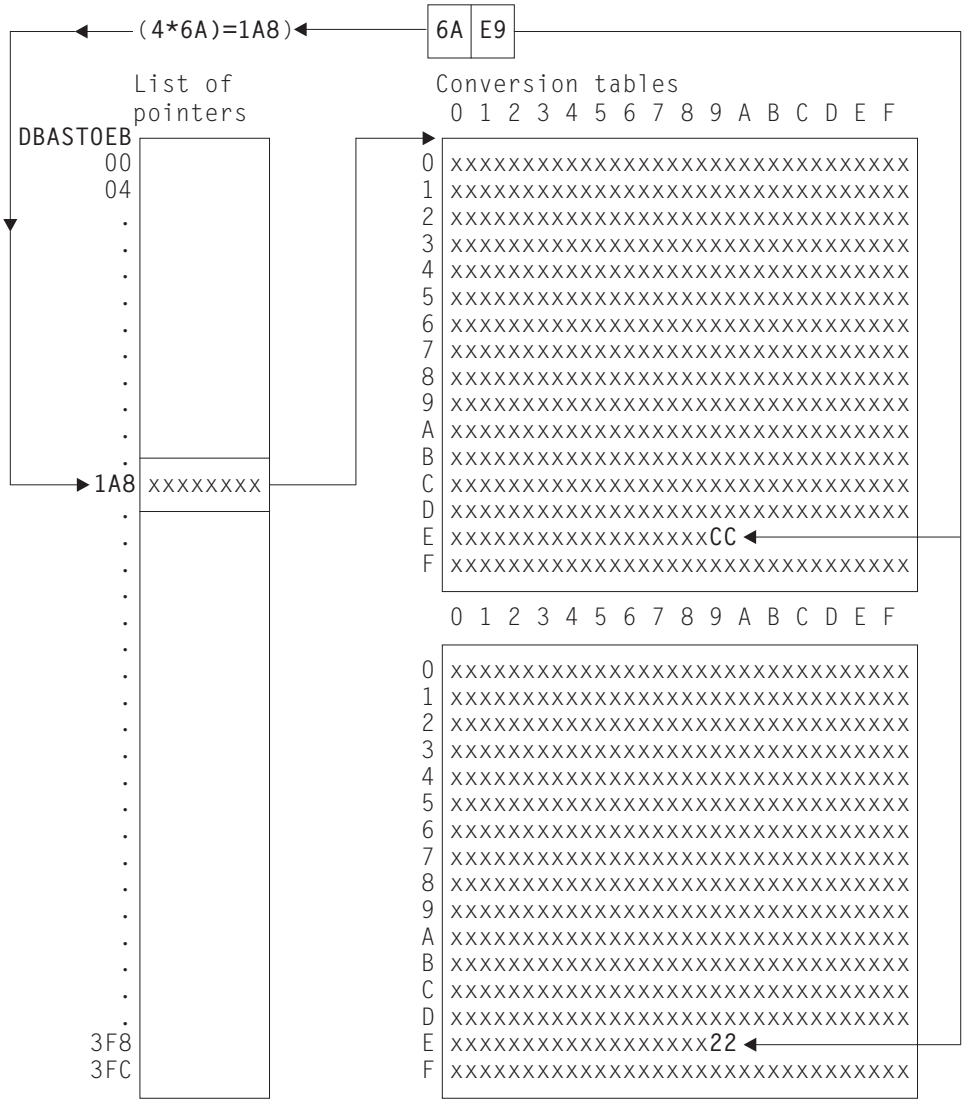
You must provide two character conversion tables, labelled ASTOEB and EBTOAS. Each table must be 256 bytes long. ASTOEB is used for ASCII to EBCDIC conversion and EBTOAS is used for EBCDIC to ASCII conversion. The hexadecimal value of a character byte is used as an offset in the conversion table to obtain the converted value of the character. Figure 15 on page 76 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 77 illustrates this process.

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

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 labeled 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 DEBUTS 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.

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  ADDRX        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,CLINTCP=437,SRVERCP=037
    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,CLINTCP=437,SRVERCP=037
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  ADDRX        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

```
   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 00 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.

```
*
LABL1    DFHCNV TYPE=INITIAL,CLINTCP=437,SRVERCP=037
*
         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'C0C1C2C3C4C5C6C7C8C9CACBCCCDCECF'
         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    XL16'80818283848586878889CACBCCCDCECF'
         DC    XL16'90919293949596979899A9B9C9D9E9F'
         DC    XL16'A0A1A2A3A4A5A6A7A8A9AAABACADAEAF'
         DC    XL16'B0B1B2B3B4B5B6B7B8B9BABBBCBDBEBF'
         DC    XL16'C081828384858687889CACBCCCDCECF'
         DC    XL16'D091929394959679899DADBDCDDDEDF'
         DC    XL16'E0E1A2A3A4A5A6A7A8A9EAEBECEDEEEF'
         DC    XL16'F0F1F2F3F4F5F6F7F8F9FAFBFCFDFEFF'
         END   DFHCNVBA
```

Figure 22. SBCS user-defined conversion table

Assembling and link-editing the conversion programs

In z/OS, you can use either of the standard procedures DFHAUPLE and DFHAUPLK to assemble the DFHCNV table.

assembling conversion programs

If your CICS product is CICS/VSE 2.3 or CICS Transaction Server for VSE/ESA, or
if your CICS product can run with z/OS, 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.

On VSE or VSE/ESA systems, 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
  ```

The user-replaceable conversion program

This section describes the user-replaceable data conversion program.

User-named conversion programs

Important
This section applies only to CICS Transaction Server for z/OS, Version 2 Release 2 and later.

Releases of CICS TS for z/OS from Version 2.2 onwards allow you to replace DFHUCNV, the default user-replaceable conversion program supplied with your CICS on System/390 product, by one or more user-named conversion programs.

DFHUCNV is invoked if:

- A conversion template is not defined for the resource, *or*
- A conversion template is defined for the resource and the template specifies `USREXIT=YES`.

A user-named conversion program is invoked if:

- A conversion template is defined for the resource and the template specifies `USREXIT=`*`userprogram`*

  where *userprogram* is the name of the user-supplied conversion program.

In the following sections, the generic term "DFHUCNV" represents both the (possibly customized) IBM-supplied conversion program and user-named conversion programs.

DFHUCNV

For an explanation of why you might need to amend or replace DFHUCNV, refer to "User/CICS conversion" on page 60. 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" on page 83:
  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 91:

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)") and the conversion template (see the general description of conversion and key templates in "Conversion and key templates" on page 64 and the reference information in "Conversion and key templates" on page 87).

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.
- An assembler DSECT for field conversion records (these are the basic components of a template; see Figure 25 on page 87).
- Equates for resource types and field types.

Parameter list (DFHUVNDS): Figure 23 on page 84 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
UNVCLIDP DS    AL4                A "client" CCSID
*                                   (for example, 00819)
UNVSRIDP DS    AL4                A "server" CCSID
*                                   (for example, 00285)
```

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.

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 conversion from client encoding to server encoding.
- CNVRPETA (X'04') indicates a response needing conversion from server encoding to client encoding.

UNVDTMP
Points to the start of the conversion template found by CICS to match this resource. If UNVDTMP is zero no template was found.

UNVDLNP
Points to a field that gives the length of the conversion template. The field is:
- A fullword for CICS TS for z/OS, Version 2.2 and later
- A half-word for all other CICS on System/390 products

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

UNVKLNP (file control requests only)
Points to a field that gives the length of the key conversion template. The field is:
- A fullword for CICS TS for z/OS, Version 2.2 and later
- A half-word for all other CICS on System/390 products

UNVATEP
Points to a 256-byte SBCS translation table used for converting character data from client encoding to server encoding.

UNVETAP
Points to a 256-byte SBCS translation table used for converting character data from server encoding to client encoding.

UNVATED
Points to a DBCS translation table used for converting character data from client encoding to server encoding.

UNVETAD
Points to a DBCS translation table used for converting character data from server encoding to client encoding.

The overlay section depends on resource type:

TS requests:

UNVTSDP
Points to the start of the TS record being read or written. The field is:
- A fullword for CICS TS for z/OS, Version 2.2 and later
- A half-word for all other CICS on System/390 products

UNVTSLNP
Points to a 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 field that gives the length of the TD record. The field is:
- A fullword for CICS TS for z/OS, Version 2.2 and later
- A half-word for all other CICS on System/390 products

IC requests:

UNVICDP
Points to the “from” area of an IC START request.

UNVICLNP
Points to a field that gives the length of the “from” area. The field is:
- A fullword for CICS TS for z/OS, Version 2.2 and later
- A half-word for all other CICS on System/390 products

PC requests:

UNVPCDP
Points to the start of the COMMAREA being supplied.

UNVPCLNP
Points to a field that gives the length of the COMMAREA. The field is:
- A fullword for CICS TS for z/OS, Version 2.2 and later
- A half-word for all other CICS on System/390 products

FC requests:

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

UNVFCLNP
Points to a field that gives the length of the file control record. The field is:
- A fullword for CICS TS for z/OS, Version 2.2 and later
- A half-word for all other CICS on System/390 products

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

UNVFCKLP
Points to a field that gives the length of the key. The field is:
- A fullword for CICS TS for z/OS, Version 2.2 and later
- A half-word for all other CICS on System/390 products

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
```

UNVCLIDP (applies only to CICS TS for z/OS Version 2.2 and later)
Points to a fullword field that gives the IBM-defined CCSID, for example 00819, corresponding to the “client” code page.

UNVSRIDP (applies only to CICS TS for z/OS Version 2.2 and later)
Points to a fullword field that gives the IBM-defined CCSID, for example 00285, corresponding to the “server” code page.

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.

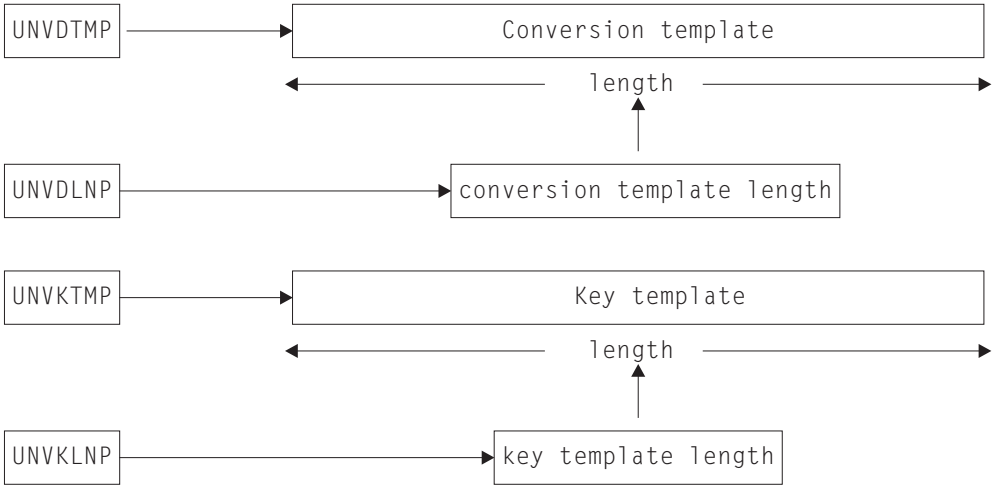


Figure 24. Parameter fields and the conversion templates

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 88, and mapped by a supplied DSECT, DFHCNVDS (see "DFHCNVDS, DSECT for field conversion records" on page 89). 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.

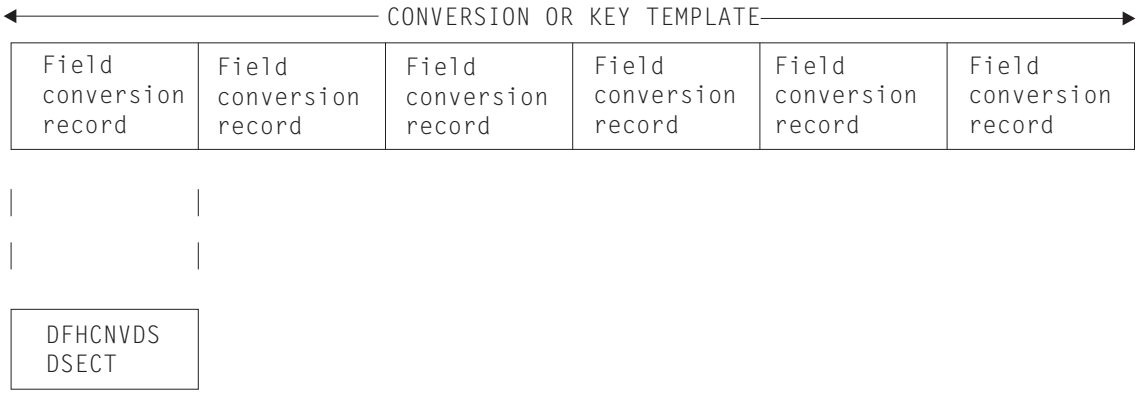


Figure 25. Field conversion records and a conversion or key template

Field conversion records: For CICS TS for z/OS, Version 2.2 and later, a field conversion record has the following layout:

Table 40. Layout of a field conversion record for CICS TS for z/OS, Version 2.2 and later

CNVRLEN	CNVRTYPE	Reserved	CNVDATTY	CNVDATAO	CNVDATAL
Record length	Record type	Reserved	Data type	Data offset	Data length
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5-8	Byte 9-12

For all CICS on System/390 products other than CICS TS for z/OS Version 2.2 and later, a field conversion record has the following layout:

Table 41. Layout of a field conversion record for CICS on System/390 products other than CICS TS for z/OS Version 2.2 and later

CNVRLEN	CNVRTYPE	Reserved	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 40 and Table 41, 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 89 and Figure 27 on page 90). 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 X'0C' for releases of CICS TS for z/OS from Version 2.2 onwards, and X'08' for all other CICS on System/390 products. 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 91.

DFHCNVDS, DSECT for field conversion records:

CICS TS for z/OS, Version 2.2 and later, version:

```
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
DTDBCS   EQU   X'05'               DBCS
DTNUM    EQU   X'06'               NUMERIC
DTUSRMIN EQU   X'50'               MINIMUM USER DATA TYPE
DTUSRMAX EQU   X'80'               MAXIMUM USER DATA TYPE
*
CNVDATAO DS    AL4                 DATA OFFSET
CNVDATAL DS    AL4                 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. ***This is the CICS TS for z/OS, Version 2.2 and later, version****.*

DFHCNVDS, DSECT for field conversion records:

Non-CICS TS for z/OS, Version 2.2 and later, version:

```
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
DTDBCS   EQU   X'05'               DBCS
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 27. DFHCNVDS, DSECT that maps conversion/key templates passed to DFHUCNV. ***This is the version for all CICS on System/390 products other than CICS TS for z/OS, Version 2.2 and later****.*

Supplied user-replaceable conversion program

Figure 28 lists the version of DFHUCNV supplied with CICS TS for z/OS, Version 2.2 and later. Figure 29 on page 97 lists the version of DFHUCNV supplied with all other CICS on System/390 products. Both versions are written in assembler.

The supplied version of DFHUCNV 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.

User-replaceable conversion program—CICS TS for z/OS, Version 2.2 and later, version:

```
*   MODULE NAME = DFHUCNV
*
* DESCRIPTIVE NAME = C.I.C.S./.......
**       CICS TS for Windows USER CONVERSION SAMPLE PROGRAM
**
*
* TRANSACTION NAME = Cxxx
**       NOT A TRANSACTION
*
* 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 TS for Windows 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 TS for Windows (ASCII TO EBCDIC) OR
*        OUTBOUND FROM CICS TS for Windows (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 28. DFHUCNV, user-replaceable conversion program for CICS on System/390—CICS Transaction Server for Windows link (Part 1 of 6). ***This is the CICS TS for z/OS, Version 2.2 and later, version****.*

```
* 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, SERIALLY 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
*         DFHCNV 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 28. DFHUCNV, user-replaceable conversion program for CICS on System/390—CICS Transaction Server for Windows link (Part 2 of 6). ***This is the CICS TS for z/OS, Version 2.2 and later, version****.*

```
*     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 TS for Windows 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 28. DFHUCNV, user-replaceable conversion program for CICS on System/390—CICS Transaction Server for Windows link (Part 3 of 6). ***This is the CICS TS for z/OS, Version 2.2 and later, version****.*

```
*        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
*        DFHCNV TABLE (CONTAINING THE TEMPLATES) WILL BE LOADED
*        ABOVE THE 16M LINE.
*
*    CAVEAT
*
*        FULLWORD VALUES ARE NOW PASSED IN THE FOLLOWING
*        FIELDS:
*
*          CNVDATAL
*          CNVDATAO
*
*          UNVFCLNP
*          UNVFCKLP
*          UNVICLNP
*          UNVPCLNP
*          UNVTDLNP
*          UNVTSLNP
*
*********************************************************************
```

Figure 28. DFHUCNV, user-replaceable conversion program for CICS on System/390—CICS Transaction Server for Windows link (Part 4 of 6). ***This is the CICS TS for z/OS, Version 2.2 and later, version****.*

```
         DFHCNV TYPE=DSECT
DFHUCNV  CSECT
         B     IDBYP
         DFHVM UCNV,ENTRY=DFHUCNV,RMODE=ANY
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
         L     R5,0(0,R4)          GET TOTAL TEMPLATE LENGTH
         AR    R5,R10              END OF TEMPLATE
PROCESS  DS    0H
         CR    R10,R5              HAVE WE REACHED THE END OF TEMPL
         BNL   RETURN              YES
         CLI   CNVRTYPE,CNVTFLD    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
*
*        GET LOWER VALUE OF ACTUAL LENGTH AND POTENTIAL LENGTH
*        INTO R4
         L     R4,UNVTSLNP
         L     R4,0(0,R4)          PICK UP ACTUAL TS DATA LENGTH
         LTR   R4,R4               JUST CHECK ITS POSITIVE
         BNP   RETURN              IF NOT RETURN
         L     R7,CNVDATAO         GET THE OFFSET FROM THE TEMPLATE
         CR    R7,R4               IS THE OFFSET PAST THE DATA
         BNL   NEXTREC             YES, TRY THE NEXT RECORD
         A     R7,CNVDATAL         ADD IN THE LENGTH(TEMPLATE)
         CR    R7,R4               COMPARE OFFSET+LEN WITH REAL DATA
         BH    LENOK
         LR    R4,R7
LENOK    DS    0H
```

Figure 28. DFHUCNV, user-replaceable conversion program for CICS on System/390—CICS Transaction Server for Windows link (Part 5 of 6). ***This is the CICS TS for z/OS, Version 2.2 and later, version****.*

```
*        HERE R4 SHOULD BE THE SMALLER OF THE 2 LENGTHS
*        NOW CALCULATE THE  REAL LENGTH FOR CONVERSION
         S     R4,CNVDATAO         SUBTRACT THE OFFSET
         L     R7,UNVTSDP          ADDRESS ACTUAL DATA
         A     R7,CNVDATAO         ....PLUS OFFSET
*
*  R7 POINTS AT THE START OF WHERE WE TRANSLATE AND R4
*  INDICATES THE LENGTH (ENSURING WE DONT 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,CNVRLEN          GET LENGTH OF THIS RECORD
         AR    R10,R4              AND ADDRESS THE NEXT ONE
         B     PROCESS
RETURN   DS    0H
         EXEC  CICS RETURN
        END   DFHUCNV
```

Figure 28. DFHUCNV, user-replaceable conversion program for CICS on System/390—CICS Transaction Server for Windows link (Part 6 of 6). ***This is the CICS TS for z/OS, Version 2.2 and later, version****.*

User-replaceable conversion program—non-CICS TS for z/OS, Version 2.2 and later, version:

```
*   MODULE NAME = DFHUCNV
* DESCRIPTIVE NAME = C.I.C.S./.......
**       CICS TS for Windows 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 TS for Windows 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 TS for Windows (ASCII TO EBCDIC) OR
*        OUTBOUND FROM CICS TS for Windows (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 29. DFHUCNV, user-replaceable conversion program for CICS on System/390—CICS Transaction Server for Windows link (Part 1 of 6). ***This is the version for all CICS on System/390 products other than CICS TS for z/OS, Version 2.2 and later****.*

```
* 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, SERIALLY 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
*         DFHCNV 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 29. DFHUCNV, user-replaceable conversion program for CICS on System/390—CICS Transaction Server for Windows link (Part 2 of 6). ***This is the version for all CICS on System/390 products other than CICS TS for z/OS, Version 2.2 and later****.*

```
*     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 TS for Windows 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 29. DFHUCNV, user-replaceable conversion program for CICS on System/390—CICS Transaction Server for Windows link (Part 3 of 6). ***This is the version for all CICS on System/390 products other than CICS TS for z/OS, Version 2.2 and later****.*

```
*        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
*        DFHCNV TABLE (CONTAINING THE TEMPLATES) WILL BE LOADED
*        ABOVE THE 16M LINE.
*
*------------------------------------------------------------------ *
*
* CHANGE ACTIVITY :
**
*        $MOD(DFHUCNV),COMP(ISC),PROD(CICS/MVS)
*
*     PN= REASON REL YYMMDD HDXIII : REMARKS
*     P0= REASON REL YYMMDD 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 29. DFHUCNV, user-replaceable conversion program for CICS on System/390—CICS Transaction Server for Windows link (Part 4 of 6). ***This is the version for all CICS on System/390 products other than CICS TS for z/OS, Version 2.2 and later****.*

```
         DFHCNV TYPE=DSECT
DFHUCNV  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,CNVTFLD    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 29. DFHUCNV, user-replaceable conversion program for CICS on System/390—CICS Transaction Server for Windows link (Part 5 of 6). ***This is the version for all CICS on System/390 products other than CICS TS for z/OS, Version 2.2 and later****.*

```
*        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,CNVRLEN          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 29. DFHUCNV, user-replaceable conversion program for CICS on System/390—CICS Transaction Server for Windows link (Part 6 of 6). ***This is the version for all CICS on System/390 products other than CICS TS for z/OS, Version 2.2 and later****.*

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 Universal Client, the CICS Client elements of the CICS Transaction Gateway products, and the CICS Transaction Gateway client daemons (hereafter all jointly referred to as *CICS Clients*).

Important

- This part of the book describes the support for CICS Clients provided by:
  - CICS Transaction Server for z/OS
  - CICS Transaction Server for OS/390
  - CICS Transaction Server for VSE/ESA

  The support for Clients provided by these products is identical, *except that*:
  - In CICS Transaction Server for VSE/ESA, 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.

This part of the book contains the following topics:

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?

Terminology

In this book, we use the term *CICS Clients* to mean all of the following:

- The CICS Universal Client, which runs on:
  - AIX Version 5.3
  - Linux® on Intel®, which comprises:
    - Red Hat Enterprise Linux Version 4 with kernel 2.6 and glibc 2.3.2
    - Novell LINUX Desktop 9
    - SUSE LINUX Desktop Version 1 with kernel 2.4 and glibc 2.3.2
    - SUSE LINUX Enterprise Server 9
  - Linux on POWER®, which comprises:
    - Red Hat Enterprise Linux Version 4 with kernel 2.6 and glibc 2.3.2
    - SUSE LINUX Enterprise Server Version 9 with 64-bit kernel 2.6 and glibc 2.3.2
  - Microsoft® Windows 2000 Professional with service pack 4
  - Microsoft Windows XP Professional with service pack 1
- The CICS Client elements of the CICS Transaction Gateway products (which are available on all the platforms listed below)
- The client daemons of the CICS Transaction Gateway products

A CICS Client is not a full-function CICS system, but 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
- Microsoft Windows 2000
- Microsoft Windows XP
- Solaris
- HP-UX
- Linux 390

Each Client can attach to any or all of the following CICS servers:
- CICS Transaction Server for z/OS
- CICS Transaction Server for OS/390
- CICS Transaction Server for VSE/ESA
- CICS/VSE Version 2.3
- CICS/400
- CICS Transaction Server for Windows
- CICS on Open Systems

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.

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 z/OS and CICS Transaction Server for OS/390 Release 3 allow you to route ECI calls dynamically. In these products, 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 for OS/390 Release 3 and later.

For definitive information about how to route program-link requests dynamically, see your *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 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
- Java™
- Visual Basic (on Windows client).

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 virtual terminal may attach.
- Attach a virtual terminal.
- Detach a virtual terminal.
- Start a transaction for a virtual terminal.
- Send data from a virtual terminal to a transaction.
- Obtain details of an “event” that has occurred for a virtual terminal. An example of an event is when the transaction is expecting a reply from the virtual terminal.
- Obtain detailed error information for the last error that occurred for a virtual terminal.

The External Security interface

The External Security Interface (ESI) is an API that allows a non-CICS Client program to verify and change the passwords used by Clients to connect to a CICS server.

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?

Any CICS Client can use the LU 6.2 (APPC) communication protocol to communicate with any CICS on System/390 server. Some CICS Client–CICS on System/390 combinations support the use of the Transport Control Protocol/Internet Protocol (TCP/IP). Support for TCP/IP is described in “TCP/IP support” on page 108.

APPC

CICS on System/390-CICS Client applications can use the 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. You can autoinstall both single-session and parallel-session APPC connections. 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

Client–CICS on System/390 TCP/IP communication is supported in two ways:
1. Via TCP62 and AnyNet®
2. Natively, via ECI over TCP/IP.

Using TCP62 and AnyNet

Important
This section applies to:
- CICS Transaction Server for z/OS
- CICS Transaction Server for OS/390

This method 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. It extracts SNA definitions from the Client and uses them when communicating with CICS. The AnyNet feature of VTAM is also required.

The System/390 servers supported are:
- CICS Transaction Server for z/OS
- CICS Transaction Server for OS/390

The Clients supported are those for:
- Microsoft Windows NT
- Microsoft Windows 2000
- Microsoft Windows XP

Both the ECI and the EPI are supported.

For information about how to use TCP62 and AnyNet to set up a TCP/IP connection to a CICS Client, see “Using TCP62 and AnyNet” on page 112.

Using ECI over TCP/IP

Important
This section applies only to CICS Transaction Server for z/OS, Version 2 Release 2 and later.

This method allows Clients to use TCP/IP directly to CICS without any intervening products such as TCP62.

The only System/390 servers supported are releases of CICS Transaction Server for z/OS from Version 2.2 onwards.

All Clients are supported.

Only the ECI (not the EPI nor the ESI) is supported.

This is the preferred method of setting up TCP/IP connections between Clients and CICS on System/390, because:
- It does not require an intervening product to perform TCP/IP-to-SNA datastream conversion.
- It simplifies the administration of large networks of Clients, because TCP/IP is easier to configure than SNA.

- It reduces the cost of administration of large networks of Clients.
- To migrate to it, existing client and server programs that communicate via APPC do not have to be rewritten.

For information about how to set up an ECI over TCP/IP connection to a CICS Client, see “Using ECI over TCP/IP” on page 114.

Benefits of Client support

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

- 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 Transaction Gateway: Programming Guide* , SC34-6141, and the *CICS Transaction Gateway: Programming Reference* , SC34-6140. The remainder of this document deals with setting up and administering the CICS on System/390 side of the Client-server link.

Chapter 8. Installing server support for Clients

Important

This chapter applies to:

- CICS Transaction Server for z/OS
- CICS Transaction Server for OS/390
- CICS Transaction Server for VSE/ESA

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 for communication with non-System/390 systems," 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 groups, DFHCLNT and DFHIPECI—see "Installing the DFHCLNT and DFHIPECI resource groups"
- Define the CSCC and CIEO transient data queues—see "Defining the CSCC and CIEO transient data queues"
- Install connections to the Clients—see "Installing connections to Clients" on page 112
- Install some Client virtual terminals—see "Installing Client virtual terminals" on page 116
- Specify the level of security to be used for Client-CICS on System/390 links—see "Setting up security" on page 124
- Specify the code pages to be used for data conversion—see Chapter 9, "Data conversion for Clients," on page 127.

Installing the DFHCLNT and DFHIPECI resource groups

For general Client support, 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.

If you want to use ECI over TCP/IP, you must install the DFHIPECI CICS-supplied resource definition group, which includes definitions of the internal CICS transaction, CIEP, and CIEP's associated program, DFHIEP.

The DFHIPECI resource group is included in the default CICS startup group list, DFHLIST.

Defining the CSCC and CIEO transient data queues

Messages relating to APPC-connected Clients are written to the CSCC transient data queue. If you use APPC-connected Clients, 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.

Messages relating to Clients connected to CICS by native TCP/IP are written to the CIEO transient data queue. If your Clients use ECI over TCP/IP (see “Using ECI over TCP/IP” on page 108), you must define CIEO to CICS. There is a sample definition in the DFHDCTG resource group. The sample defines CIEO as an indirect extrapartition destination, pointing to CSSL.

The DFHDCTG resource definition group is included in the default CICS startup group list, DFHLIST.

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

Installing connections to Clients

Unless you are using ECI over TCP/IP exclusively (see “Using ECI over TCP/IP” on page 108), 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.

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

Autoinstalled APPC connections

The preferred method of installing APPC 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

TCP/IP communication between Clients and CICS on System/390 is supported in two ways:

1. Via TCP62 and AnyNet—see “Using TCP62 and AnyNet”
2. Via ECI over TCP/IP—see “Using ECI over TCP/IP” on page 114.

Using TCP62 and AnyNet

Important

This section applies to:

- CICS Transaction Server for z/OS
- CICS Transaction Server for OS/390

System/390 software required: TCP/IP via TCP62 requires the AnyNet feature of VTAM. The other System/390 software you require depends on the level of VTAM you are running. Table 42 shows the acceptable software combinations.

Table 42. System/390 software required for TCP/IP via TCP62. Each row lists an acceptable software combination.

• 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
• 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
• 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
• 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 via TCP62: Enabling CICS to communicate with a Client using TCP/IP via TCP62 requires actions on z/OS, CICS, VTAM, and the Client workstation.

On z/OS: On z/OS, you must:

1. Install a TCP major node. For example:

```
ABHTCP   VBUILD TYPE=TCP,
                CONTIMER=25,
                DGTIMER=40,
                DNSUFX=HURSLEY.IBM.COM,
                EXTIMER=5,
                IATIMER=60,
                PORT=397,
                TCB=10,
                TCPIPJOB=TCPIP
 ABH1GRP  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 library 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. Set the TCP/IP system initialization parameter to 'YES'.
3. 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.

Using ECI over TCP/IP

Important

This section applies only to CICS Transaction Server for z/OS, Version 2 Release 2 and later.

Enabling ECI over TCP/IP: Enabling ECI over TCP/IP requires actions on CICS and the Client workstation.

On CICS: On CICS, you must:

1. Create a TCPIPSERVICE definition for ECI over TCP/IP. You can use the definition in RDO group DFH$SOT as supplied, edit it to suit your requirements, or create your own definition.

   A TCPIPSERVICE for ECI over TCP/IP requires SOCKETCLOSE(NO) to be specified. The recommended method to achieve a timeout for a task initiated through ECI over TCP/IP is to specify an RTIMOUT value on the mirror transaction. The standard mirror transaction, CPMI, is defined with profile DFHCICSA, which has RTIMOUT(NO). This means that long-running mirrors will wait indefinitely for data unless you customize the RTIMOUT value for the mirror transaction.

   The supplied TCPIPSERVICE definition specifies an attach-time security level of VERIFY and a TCP/IP port number of 1435. If you want some of your Clients to use LOCAL security and others to use VERIFY, you must create two TCPIPSERVICE definitions. The two definitions should have different security attributes and must listen on different ports. In each Client initialization file, specify that the Client should use the port for the appropriate security level.
2. Install your TCPIPSERVICE definitions.
3. Set the TCPIP system initialization parameter to 'YES'.

On the Client workstation: On the Client workstation, you must:

1. Edit the Client initialization file. In the initialization file, code a server section like the following example:

```
SECTION SERVER = ECICLNT
  DESCRIPTION=TCP/IP Server
  UPPERCASESECURITY=N
  USENPI=N
  PROTOCOL=TCPIP
  NETNAME=hostname.example.com
  PORT=1435
  CONNECTTIMEOUT=0
  TCPKEEPALIVE=N
ENDSECTION
```

   Notes:

   a. On the PORT option, specify the number of the TCP/IP port to be used for conversations with CICS.

   b. On the NETNAME option, specify the TCP/IP name of the CICS System/390 server—for example, `winvmb.hursley.ibm.com`.

2. Optionally, uninstall TCP62 support.
3. When starting the client, use a command line input of `cicscli /s=ECICLNT`, so that the ECICLNT section of the initialization file is used when the client starts.

When setting up Clients to use ECI over TCP/IP, bear the following in mind:

- Only the ECI, not the EPI nor the ESI, is supported. You must ensure that applications that issue EPI or ESI calls are not run on an ECI over TCP/IP connection.
- The mirror transaction on CICS System/390 must not be defined as remote. It should use a profile that specifies a timeout—see “Enabling Ping support.”

Enabling Ping support: ECI over TCP/IP support includes support for conversation-level and connection-level ping. These ping flows are initiated from CICS if a Client becomes unresponsive, *but only if the mirror transaction is running with a profile that specifies a timeout*. The default mirror does not have a timeout. To enable ping support you must use a mirror profile that specifies a timeout. If you don’t, and a Client becomes disconnected due to a connection failure, any current conversations with the Client could be suspended indefinitely.

When a timeout occurs, CICS does the following:

1. If the Client supports conversation-level ping, CICS tries to ping a specific conversation. This attempts to confirm whether a conversation is still active on the connection.
2. If the Client does not support conversation-level ping, or the conversation-level ping request times out, CICS tries a connection-level ping. This queries whether a connection to the Client still exists in TCP/IP.
3. If the connection-level ping request times out, CICS uninstalls the Client from CICS. Any state associated with the Client is lost. Any suspended tasks associated with the Client are abended.

If a connection fails during an ECI extended conversation, CICS detects this when the ping timeout processing occurs. The conversation is abended.

Port sharing in a CICSplex: Port sharing, used as a method of load balancing in a CICSplex, is supported for TCP/IP-connected Clients.

installation of Client support

When a Client is installed into CICS, the install request is routed to a particular CICS region. The sockets domain in that region allocates an ephemeral port for use with the Client during the rest of the time for which it is installed. The ephemeral port is not shared, so an affinity between the Client and the CICS region on which it is installed is established. Until the client is uninstalled or the connection fails (forcing the Client to be re-installed), all subsequent work is routed to the correct CICS region through the unshared ephemeral port.

Installing 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 120). For information about the `CICS_EpiAddTerminal` function, see the *CICS Transaction Gateway: Programming Reference* 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.

Figure 30 on page 117 and Figure 31 on page 118 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.
- 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 “The EPI and terminal emulator” on page 131.
- 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
  AUTINSTModel ==> 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
  OPERRsl      ==> 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 statically defined Client virtual terminal

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
  TERmmodel    ==> 2
  SESsiontype  ==>
  LDclist      ==>
  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
  EXtendedds   ==> Yes                No|Yes
  Query        ==> All                No|Cold|All
  SOsi         ==> No                 No|Yes
  BAcktrans    ==> No                 No|Yes
  CGcsgid      ==> 00000 , 00037      0-65535
 SESSION PROPERTIES
  AScii        ==> No                 No|7|8
  SENdsize     ==> 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|All
  ATi          ==> Yes                No|Yes
  CReatesess   ==> No                 No|Yes
  RELreq       ==> Yes                No|Yes
  DIscreq      ==> Yes                Yes|No
 MESSAGE RECEIVING PROPERTIES
  ROutedmsgs   ==> All                All|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 31. 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 Transaction Gateway: Programming Reference* 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.
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*.)

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 TS for OS/390 and CICS TS for z/OS 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 "Writing EPI server programs" on page 135.

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

CICS TS for OS/390 and CICS TS for z/OS 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 "Writing EPI server programs" on page 135.

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 43 relates the parameters passed by the Client to the way in which CICS installs the Client terminal.

Table 43. 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**	**TERMID**	**Autoinstall model**
Yes	No	Yes	Use static definition [10]	Client-specified	-
Yes	No	No	Call fails	-	-
Yes	Yes	No	Autoinstall	Client-supplied [11]	Client-specified
Yes	Yes	Yes	Use static definition [10]	Client-specified	-
No	No	-	Autoinstall	CICS-supplied [11]	DFHLU2
No	Yes	-	Autoinstall	CICS- supplied [11]	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. [12]

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 32 on page 122 and Figure 33 on page 123 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

10. 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.
11. In CICS Transaction Server for OS/390 and CICS Transaction Server for z/OS, 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.
12. 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 Transaction Server for OS/390, CICS Transaction Server for z/OS, or CICS Transaction Server for VSE/ESA system.

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.
- 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 "The EPI and terminal emulator" on page 131.
- 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
  OPERRsl      ==> 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 32. 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
  TERmmodel    ==> 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|Cold|All
  SOsi         ==> No                 No|Yes
  BAcktrans    ==> No                 No|Yes
  CGcsgid      ==> 00000 , 00037      0-65535
 SESSION PROPERTIES
  AScii        ==> No                 No|7|8
  SENdsize     ==> 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 33. Example TYPETERM definition for a Client-attached printer

Setting up security

Important

In CICS Transaction Server for z/OS and CICS Transaction Server for OS/390, 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 (APPC only)

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

Link security (APPC only)

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 APPC 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.

If you are using APPC links, specify the level of user-security on the ATTACHSEC option of the CONNECTION definition that defines the Client to CICS on System/390.

If you are using ECI over TCP/IP, specify the level of user-security on the ATTACHSEC option of the TCPIPSERVICE definition for ECI over TCP/IP.

The valid values of ATTACHSEC for CICS Client-System/390 links are LOCAL and VERIFY.

For APPC links, 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) 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 and CIEP (which is used for ECI over TCP/IP) are category 3 transactions—that is, they are 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 and CIEP
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 and DFHIPECI CSD groups 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 to:

- CICS Transaction Server for z/OS
- CICS Transaction Server for OS/390
- CICS Transaction Server for VSE/ESA

If you are using a **channel** to perform data conversion, read "Data conversion using channels" in the *CICS Application Programming Guide* instead of this topic. Channels are available in CICS Transaction Server for z/OS Version 3 Release 1 onwards.

For detailed information about data conversion between CICS workstation products and CICS on System/390 systems, see Chapter 6, "Data conversion for communication with non-System/390 systems," 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 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-defined conversion tables or user-code in the user-replaceable program, DFHUCNV—is not supported.

For the ECI:

- Data conversion is done in the region to which the Client is connected.
- 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 the Universal Client for Windows the client code page is specified by running the Client configuration tool and doing one of the following:

- Checking the box labeled "Use OEM codepage". This sets the value of the USEOEMCP parameter to Y in the Client section of the ctg.ini initialization file.
- Specifying a value for the field labeled "Codepage identifier override". This sets the value of the CCSID parameter in the Client section of the ctg.ini initialization file.

If the CICS Universal Client finds that the CCSID parameter in the ctg.ini file has a value, it sends that value to the server as the client CCSID. Otherwise, it issues either a GetACP function call to obtain the ANSI code page or a GetOEMCP call to obtain the OEM code page, and sends that code page to the server.

The client code page can be 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 44.

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 44. 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	00864	00420
Arabic	01089	00420
Arabic	01256	00420
Arabic	05352	16804
Arabic	09448	16804
Arabic	17248	16804
Baltic Rim	00901	01156
Baltic Rim	00902	01157
Baltic Rim	00921	01112
Baltic Rim	00922	01122
Baltic Rim	01257	01112
Baltic Rim	05353	01156
Cyrillic	00808	01154
Cyrillic	00848	01158
Cyrillic	00849	01154
Cyrillic	00855	01025
Cyrillic	00866	01025
Cyrillic	00872	01154
Cyrillic	00915	01025
Cyrillic	01124	01123
Cyrillic	01125	01123
Cyrillic	01131	01025

Table 44. EPI code pages (continued). 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
Cyrillic	01251	01025
Cyrillic	05347	01154
Devanagari	00806	01137
Farsi	01098	01097
Greek	00813	00875
Greek	00869	00875
Greek	01253	00875
Greek	04909	04971
Greek	05349	04971
Greek	09061	04971
Hebrew	00856	00424
Hebrew	00862	00424
Hebrew	00867	12712
Hebrew	00916	00424
Hebrew	01255	00424
Hebrew	05351	12712
Hebrew	09447	12712
Japanese	00932	00930
Japanese	00942	00930
Japanese	00943	00930
Japanese	00954	00930
Japanese	05050	01390
Korean	00934	00933
Korean	00944	00933
Korean	00949	00933
Korean	00970	00933
Korean	01363	00933
Lao	01133	01132
Latin-1 and Latin-9	00437	00500
Latin-1 and Latin-9	00819	00500
Latin-1 and Latin-9	00850	00500
Latin-1 and Latin-9	00858	00500
Latin-1 and Latin-9	00923	00924
Latin-1 and Latin-9	00924	00924 **Note:** EBCDIC/EBCDIC conversion not supported.
Latin-1 and Latin-9	01047	01047 **Note:** EBCDIC/EBCDIC conversion not supported.
Latin-1 and Latin-9	01252	00500

Table 44. EPI code pages (continued). 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
Latin-1 and Latin-9	05348	01148
Latin-2	00852	00870
Latin-2	00912	00870
Latin-2	01250	00870
Latin-2	05346	01153
Latin-2	09044	01153
Latin-5	00857	01026
Latin-5	00920	01026
Latin-5	01254	01026
Latin-5	05350	01155
Latin-5	09049	01155
Simplified Chinese	00946	00935
Simplified Chinese	01381	00935
Simplified Chinese	01383	00935
Simplified Chinese	01386	00935
Simplified Chinese	05488	01388
Thai	01161	01160
Thai	01162	01160
Thai	09066	09030
Traditional Chinese	00938	00937
Traditional Chinese	00948	00937
Traditional Chinese	00950	00937
Traditional Chinese	00964	00937
Traditional Chinese	01370	01371
Urdu	00868	00918
Urdu	01006	00918
Vietnamese	01129	01130
Vietnamese	01163	01164
Vietnamese	01258	01130
Vietnamese	05354	01164

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 45. 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 44 on page 128.

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 45 shows the server character sets and code pages supported for the EPI.

From CICS TS for z/OS Version 2.2 onwards, the supported character sets and codepages listed reflect the CGCSGID returned by a 3270 device in response to a Query Reply. Table 45 highlights all the differences between CICS TS for z/OS Version 2.2 (and later) and earlier CICS releases.

In CICS TS for z/OS Version 2.2 and later, if you want to use a unique National Language group use the code page and character set values.

Table 45. EPI character sets and server code pages. The character sets and server code pages supported by CICS on System/390 for the EPI.

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	00235	00420	—	—
Arabic	01461	00420	—	—
Baltic Rim	01305	01112	—	—
Baltic Rim	01393	01156	—	—
Cyrillic	01150	01025	—	—
Cyrillic	01326	01123	—	—
Cyrillic	01381	01154	—	—

Table 45. EPI character sets and server code pages (continued). The character sets and server code pages supported by CICS on System/390 for the EPI.

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
Cyrillic	01388	01158	—	—
Estonian	01307	01122	—	—
Estonian	01391	01157	—	—
Greek	00925	00875	—	—
Greek	01371	00875	—	—
Hebrew	00941	00424	—	—
Hebrew	01147	00803	—	—
Hebrew	01357	00803	—	—
Hebrew	1356	00424	—	—
Japanese	00101	00931 00037 [1]	01001	00300
Japanese	01172	00930 00290 [1]	01001	00300
Japanese	01172	00939 01027 [1]	01001	00300
Japanese	65535	01390 00290 [1]	65535	00300
Japanese	65535	01399 01027 [1]	65535	00300
Korean	01173	00933 00833 [1]	00934	00834
Korean	65535	01364 00833 [1]	65535	00834
Latin-1 and Latin-9	00695	01140	—	—
Latin-1 and Latin-9	00695	01141	—	—
Latin-1 and Latin-9	00695	01142	—	—
Latin-1 and Latin-9	00695	01143	—	—
Latin-1 and Latin-9	00695	01144	—	—
Latin-1 and Latin-9	00695	01145	—	—
Latin-1 and Latin-9	00695	01146	—	—
Latin-1 and Latin-9	00695	01147	—	—
Latin-1 and Latin-9	00695	01148	—	—
Latin-1 and Latin-9	00695	01149	—	—
Latin-1 and Latin-9	00697	00037	—	—
Latin-1 and Latin-9	00697	00273	—	—
Latin-1 and Latin-9	00697	00277	—	—
Latin-1 and Latin-9	00697	00278	—	—
Latin-1 and Latin-9	00697	00280	—	—
Latin-1 and Latin-9	00697	00284	—	—

Table 45. EPI character sets and server code pages (continued). The character sets and server code pages supported by CICS on System/390 for the EPI.

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
Latin-1 and Latin-9	00697	00285	—	—
Latin-1 and Latin-9	00697	00297	—	—
Latin-1 and Latin-9	00697	00500	—	—
Latin-1 and Latin-9	00697	00871	—	—
Latin-1 and Latin-9	00697	01047	—	—
Latin-1 and Latin-9	01353	00924	—	—
Latin-2	00695	01148	—	—
Latin-2	00697	00500	—	—
Latin-2	00959	00870	—	—
Latin-2	01375	01153	—	—
Latin-5	01152	01026	—	—
Latin-5	01378	01155	—	—
Simplified Chinese	00103	09127 00836 [1]	00937	00837
Simplified Chinese	01174	00935 00836 [1]	00937	00837
Simplified Chinese	65535	01388 00836 [1]	65535	00837
Traditional Chinese	01175	00937 00037 [1]	00935	00835
Traditional Chinese	65535	01371 01159 [1]	65535	00835
Vietnamese	01336	01130	—	—
Vietnamese	01397	01164	—	—

Notes:

1. In CICS TS for z/OS Version 2.2 and later, use this value for the server code page. It provides SBCS and DBCS translation.

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).

- 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 63.

Chapter 10. Application programming for Clients

Important

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

- CICS Transaction Server for z/OS
- CICS Transaction Server for OS/390
- CICS Transaction Server for VSE/ESA

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

Writing 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.

Writing 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 14- or 16-bit addresses or structured fields. This is because the CICS Transaction Gateway and CICS Universal Client support only the ASCII-7 subset of the 3270 data stream architecture. Only 12-bit SBA addressing is supported. Consequently, the maximum screen size for EPI terminals is 27 rows by 132 columns.
  - Basic mapping support (BMS) partition support.
  - BMS paging.
  - The purge function to cancel ATI requests queued against the terminal. If a CICS transaction uses EXEC CICS START with the DELAY option to schedule transactions to a terminal resource autoinstalled by a user application, the user application should ensure that delayed ATI requests are not lost when the terminal resource is deleted. See your server documentation to determine the effects of deleting a terminal resource when delayed ATI requests are outstanding.
  - The terminal control commands EXEC CICS ISSUE DISCONNECT or EXEC CICS ISSUE PASS.

- 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 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 TS for OS/390 and CICS TS for z/OS 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*.

Note also that an EPI **client** application cannot:

- Use basic mapping support (BMS) paging.
- Determine the alternate screen size of the terminal resource definition, although it can determine the default screen size.

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.

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).

Chapter 11. Problem determination for Clients

Important

This chapter applies to:

- CICS Transaction Server for z/OS
- CICS Transaction Server for OS/390
- CICS Transaction Server for VSE/ESA

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 DFHZC3202-DFHZC3249 and are written to the CSCC transient data queue (TDQ).

ECI over TCP/IP error messages are in the range DFHIE0001-DFHIE1213 and are written to the CIEO TDQ.

Both sets of messages are listed in your CICS on System/390 *Messages and Codes* manual.

Abend codes

Abend codes relating to APPC-attached Clients are in the ranges AXTP-AXTR and AZAD-AZAK. Abend codes relating to TCP/IP-attached Clients are AIEB and AITH.

All abend codes are described 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 to:

- CICS Transaction Server for z/OS
- CICS Transaction Server for OS/390
- CICS Transaction Server for VSE/ESA

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, APPC 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 143.

For TCP/IP connected Clients, when CICS restarts after a failure it has no
knowledge of any Clients that may have been installed. The Clients must be
re-installed.

Recovering 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 Transaction Gateway: Programming Guide*.

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 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 to:

- CICS Transaction Server for z/OS
- CICS Transaction Server for OS/390
- CICS Transaction Server for VSE/ESA

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 to a back-end AOR, the AOR must be one of the following:
  - CICS Transaction Server for z/OS
  - CICS Transaction Server for OS/390
  - CICS Transaction Server for VSE/ESA
  - CICS/VSE 2.3
  - CICS/400

  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 APPC 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 APPC connections
  - Autoinstalled single-session APPC connections.

  These sessions are recovered only if:
  - 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 that 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.
- # For APPC connections to Clients, 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 he or she 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 135.

Chapter 14. Migration considerations

Important

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

- CICS Transaction Server for z/OS
- CICS Transaction Server for OS/390
- CICS Transaction Server for VSE/ESA

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.

Using existing applications as servers

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 135.

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:

- 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.
- 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 44 on page 128.
- 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 for communication with non-System/390 systems," on page 43.

Part 3. Appendixes

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: Communicating from CICS on System/390, SC34-6474
CICS Family: Interproduct Communication, SC34-6473

CICS on System/390 intercommunication books

CICS Transaction Server for z/OS Version 3 Release 1

CICS Distributed Transaction Programming Guide, SC34-6438-00
CICS External Interfaces Guide, SC34-6449-00
CICS Front End Programming Interface User's Guide, SC34-6436-00
CICS Intercommunication Guide, SC34-6448-00
CICS Internet Guide, SC34-6450-00

CICS Transaction Server for z/OS Version 2 Release 3

CICS Distributed Transaction Programming Guide, SC34-6236-00
CICS External Interfaces Guide, SC34-6244-00
CICS Front End Programming Interface User's Guide, SC34-6234-00
CICS Intercommunication Guide, SC34-6243-00
CICS Internet Guide, SC34-6245-00

CICS Transaction Server for z/OS Version 2 Release 2

CICS Distributed Transaction Programming Guide, SC34-5998-00
CICS External Interfaces Guide, SC34-6006-00
CICS Front End Programming Interface User's Guide, SC34-5996-00
CICS Intercommunication Guide, SC34-6005-00
CICS Internet Guide, SC34-6007-00

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 VSE/ESA Release 1.1.1

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 TS for Windows, Intercommunication, SC34-6209
CICS on Open Systems Intercommunication Guide, SC33-1564
CICS/400 Intercommunication, SC33-1388

CICS Transaction Gateway and CICS Universal Client

CICS Transaction Gateway: Programming Guide, SC34-6141
CICS Transaction Gateway: Programming Reference, SC34-6140
CICS/VSE Version 2 Release 3 Server Support for CICS Clients, SC33-1712

Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully.

You can perform most tasks required to set up, run, and maintain your CICS system in one of these ways:

- using a 3270 emulator logged on to CICS
- using a 3270 emulator logged on to TSO
- using a 3270 emulator as an MVS system console

IBM Personal Communications provides 3270 emulation with accessibility features for people with disabilities. You can use this product to provide the accessibility features you need in your CICS system.

Index

Numerics

A

B

C

D

E

F

G

I

T

U

V

W

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 Transaction Server for Windows, 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

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at Copyright and trademark information at www.ibm.com/legal/copytrade.shtml.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

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

UNIX is a registered trademark of The Open Group in the United States and other countries.

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 ask questions, make comments about the functions of IBM products or systems, or to request additional publications, contact your IBM representative or 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:

  IBM United Kingdom Limited
  User Technologies Department (MP095)
  Hursley Park
  Winchester
  Hampshire
  SO21 2JN
  United Kingdom
- By fax:
  - From outside the U.K., after your international access code use 44–1962–816151
  - From within the U.K., use 01962–816151
- Electronically, use the appropriate network ID:
  - IBMLink: HURSLEY(IDRCF)
  - Internet: idrcf@hursley.ibm.com

Whichever you use, ensure that you include:

- The publication title and order number
- The topic to which your comment applies
- Your name and address/telephone number/fax number/network ID.

SC34-6474-04

Spine information:

CICS Family

Communicating from CICS on System/390