

CICS® Universal Client for OS/2®



# Administration

*Version 3.1*



CICS® Universal Client for OS/2®



# Administration

*Version 3.1*

**Note!**

Before using this information and the product it supports, be sure to read the general information under "Appendix C. Notices" on page 149.

**Second edition (September 1999)**

This edition applies to Version 3.1 of IBM CICS Universal Client program number 5648-B42. It will also apply to all subsequent versions, releases, and modifications until otherwise indicated in new editions.

This edition replaces SC34-5450-00.

© Copyright International Business Machines Corporation 1994, 1999. All rights reserved.

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

---

# Contents

<b>Figures</b> . . . . .	<b>vii</b>
--------------------------	------------

<b>Tables</b> . . . . .	<b>ix</b>
-------------------------	-----------

<b>Summary of changes</b> . . . . .	<b>xi</b>
-------------------------------------	-----------

Changes for IBM CICS Universal Clients	
--	--

Version 3.1 . . . . .	xi
-----------------------	----

<b>About this book</b> . . . . .	<b>xiii</b>
----------------------------------	-------------

Who should read this book. . . . .	xiii
------------------------------------	------

Conventions and terminology used in this book . . . . .	xiv
---	-----

Prerequisite and related information. . . . .	xvi
---	-----

CICS Universal Clients publications. . . . .	xvi
--	-----

CICS publications . . . . .	xvi
-----------------------------	-----

NetBIOS-related publications . . . . .	xvii
--	------

TCP/IP-related publications . . . . .	xvii
---------------------------------------	------

APPC-related publications. . . . .	xvii
------------------------------------	------

Configuration, distribution, installation publications . . . . .	xviii
--	-------

Novell NetWare documentation. . . . .	xviii
---------------------------------------	-------

How to send your comments . . . . .	xviii
-------------------------------------	-------

Obtaining books from IBM. . . . .	xix
-----------------------------------	-----

<b>Chapter 1. Overview</b> . . . . .	<b>1</b>
--------------------------------------	----------

An introduction to IBM CICS Universal Clients	1
---	---

CICS Universal Clients communication . . . . .	2
--	---

The benefits of using CICS Universal Clients	3
--	---

CICS Universal Clients functions . . . . .	3
--	---

3270 terminal emulation . . . . .	3
-----------------------------------	---

The External Call Interface . . . . .	4
---------------------------------------	---

The External Presentation Interface. . . . .	4
--	---

The External Security Interface . . . . .	5
---	---

3270 client printer support . . . . .	5
---------------------------------------	---

Client control . . . . .	5
--------------------------	---

CICS telnet support . . . . .	6
-------------------------------	---

<b>Chapter 2. Planning before installation</b> . . . . .	<b>7</b>
--	----------

CICS Universal Clients and your network . . . . .	7
---	---

Connections to CICS servers . . . . .	7
---------------------------------------	---

Accessing the CICS Universal Clients . . . . .	8
--	---

Multiple protocol configurations . . . . .	9
--	---

Mobile communication. . . . .	9
-------------------------------	---

Communication between CICS Universal Clients and CICS servers. . . . .	10
--	----

Restrictions on the use of CICS servers . . . . .	11
---	----

Communication protocols . . . . .	13
-----------------------------------	----

Which protocols can you use? . . . . .	13
--	----

Why use a particular protocol?. . . . .	13
---	----

Hardware requirements . . . . .	15
---------------------------------	----

Software requirements . . . . .	16
---------------------------------	----

CICS server PTF requirements . . . . .	17
--	----

Code page support. . . . .	17
----------------------------	----

## Chapter 3. Installing CICS Universal

<b>Clients</b> . . . . .	<b>19</b>
--------------------------	-----------

Installing CICS Universal Clients . . . . .	19
---	----

Installing CICS Universal Client on OS/2 . . . . .	20
--	----

The installation procedure . . . . .	20
--------------------------------------	----

Updating your CICS Universal Client system . . . . .	22
--	----

Deleting a CICS Universal Client system	24
---	----

Automatic installation of CICS Universal Client . . . . .	24
---	----

Other actions required in setting up a CICS Universal Client. . . . .	25
---	----

## Chapter 4. Setting up client/server

<b>communication</b> . . . . .	<b>27</b>
--------------------------------	-----------

Client/server definitions. . . . .	27
------------------------------------	----

Definition checklists . . . . .	28
---------------------------------	----

NetBIOS configuration . . . . .	29
---------------------------------	----

Definition checklists . . . . .	29
---------------------------------	----

NetBIOS resources . . . . .	30
-----------------------------	----

TCP/IP configuration . . . . .	30
--------------------------------	----

IBM TCP/IP for OS/2 . . . . .	31
-------------------------------	----

Definition checklists . . . . .	31
---------------------------------	----

Verifying the TCP/IP installation . . . . .	34
---	----

TCP62 configuration . . . . .	35
-------------------------------	----

On OS/390 . . . . .	35
---------------------	----

Firewall implications . . . . .	37
---------------------------------	----

APPC configuration . . . . .	37
------------------------------	----

Configuring APPC for CICS Universal Client for OS/2 . . . . .	38
---	----

CICS Universal Clients in Novell NetWare LANs . . . . .	39
---	----

An overview of the Open Data-Link Interface . . . . .	40
---	----

Coexistence with Novell's IPX . . . . .	41	CICSTERM and user exits . . . . .	93
<b>Chapter 5. Customization . . . . .</b>	<b>51</b>	CICSTERM and RETURN TRANSID IMMEDIATE. . . . .	94
Renaming the client configuration and mapping files . . . . .	51	CICSTERM command reference . . . . .	94
Using the configuration tool . . . . .	52	The CICSPRNT command . . . . .	97
The configuration tool interface . . . . .	52	Using CICSPRNT . . . . .	98
Configuring Client settings . . . . .	55	CICSPRNT and user exits . . . . .	99
Configuring Server settings . . . . .	58	CICSPRNT and RETURN TRANSID IMMEDIATE. . . . .	99
Trace settings . . . . .	65	CICSPRNT command reference . . . . .	99
The configuration conversion tool. . . . .	67	CICS Client telnet terminal support. . . . .	102
Using the conversion tool . . . . .	67	<b>Chapter 7. Problem determination . . . . .</b>	<b>105</b>
Editing the configuration file . . . . .	68	Preliminary checks and information. . . . .	105
CLIENT section. . . . .	68	Program support . . . . .	106
SERVER section. . . . .	69	Reporting problems . . . . .	106
DRIVER section. . . . .	69	Documentation of problems . . . . .	108
Customizing the keyboard . . . . .	69	Locating and compiling information . . . . .	109
Keyboard mapping file syntax . . . . .	70	Submitting the documentation . . . . .	111
The keyboard mapping file . . . . .	70	APARs and fixes . . . . .	111
Customizing the screen colors . . . . .	74	CICS Universal Client problem determination . . . . .	112
Color mapping syntax . . . . .	75	CICS Universal Clients trace . . . . .	112
The color mapping file . . . . .	75	CICS Universal Client error processing . . . . .	119
Testing the client/server link . . . . .	78	CICS server problem determination. . . . .	120
<b>Chapter 6. Operation . . . . .</b>	<b>79</b>	Communication problem determination . . . . .	120
An introduction to the CICS Universal Clients commands . . . . .	79	NetBIOS-providing products . . . . .	121
CICS Universal Client for OS/2 operation . . . . .	80	TCP/IP-providing products . . . . .	121
CICS Universal Clients icons . . . . .	80	APPC-providing products . . . . .	122
The CICSCLI command . . . . .	80	Other tools and utilities. . . . .	123
Starting the client control process . . . . .	81	Troubleshooting . . . . .	123
Starting connections with additional servers. . . . .	81	Starting clients and terminals . . . . .	124
Stopping the client control process . . . . .	81	NetBIOS communication problems . . . . .	124
Restarting the client . . . . .	82	TCP/IP communication problems . . . . .	125
Specifying which configuration file to use . . . . .	83	APPC communication problems . . . . .	126
Turning on client trace . . . . .	83	<b>Chapter 8. Distributing CICS Universal Clients in a network . . . . .</b>	<b>127</b>
Specifying the trace components . . . . .	83	Configuration, distribution, installation . . . . .	127
Turning off client trace . . . . .	83	Performing installation and maintenance . . . . .	128
Setting up security. . . . .	83	The installation and maintenance parameters . . . . .	128
Listing the connected servers . . . . .	87	Response files . . . . .	130
Disabling the display of messages. . . . .	87	Return codes . . . . .	134
Enabling and disabling the display of pop-up messages . . . . .	87	Using SystemView for OS/2 to install CICS Universal Clients . . . . .	135
Displaying the command parameters. . . . .	87	Overview of SystemView for OS/2 installation . . . . .	135
CICSCLI and applications . . . . .	88	Copying CICS Universal Clients to the code server . . . . .	136
CICSCLI command reference . . . . .	88		
The CICSTERM command . . . . .	92		
Using CICSTERM . . . . .	92		
Stopping a terminal emulator . . . . .	93		

<b>Chapter 9. Migration issues.</b>	<b>137</b>	Book filenames.	145
Application compatibility	137	Sample configuration documents.	145
ECI and EPI compatibility	137	Other publications	146
<b>Appendix A. CICS Universal Clients data conversion</b>	<b>139</b>	Viewing the online documentation	146
Supported conversions	139	Viewing PDF books	147
<b>Appendix B. The CICS Transaction Gateway and CICS Universal Clients library</b>	<b>143</b>	<b>Appendix C. Notices</b>	<b>149</b>
CICS Transaction Gateway books	143	Trademarks	150
CICS Universal Clients books	144	<b>Glossary</b>	<b>153</b>
CICS Family publications	144	<b>Index</b>	<b>161</b>





---

## Figures

- |  |    |   |     |
|--|----|---|-----|
| 1. CICS Universal Clients communicating over NetBIOS, TCP/IP, APPC . . . . . | 9  | 7. Statements added to CONFIG.SYS for NetWare Requester for OS/2 and LAPS . . | 48  |
| 2. Mobile communication to a CICS server                                     | 10 | 8. Example of PROTOCOL.INI after setting up NetBIOS emulation . . . . .       | 49  |
| 3. Example messages returned by ping command . . . . .                       | 34 | 9. The configuration tool . . . . .   | 53  |
| 4. Implementation of Novell's ODI layer and driver . . . . .                 | 41 | 10. Sample keyboard mapping file for CICS Universal Clients. . . . .          | 73  |
| 5. Statements added to CONFIG.SYS for coexistence. . . . .                   | 43 | 11. Sample color mapping file . . . . .                                       | 77  |
| 6. Example of PROTOCOL.INI after setting up coexistence . . . . .            | 44 | 12. Sample CICS Universal Clients trace                                       | 117 |
|  |    | 13. CICS Universal Clients trace: using an invalid port number . . . . .      | 118 |



---

## Tables

- |  |     |  |     |
|--|-----|--|-----|
| 1. Command syntax conventions . . . .  | xiv | 8. Matching definitions: CICS Universal Client for OS/2 to TXSeries for AIX using TCP/IP . . . . . | 33  |
| 2. Communication between CICS Universal Client for OS/2 and CICS servers and functions . . . . .         | 10  | 9. eNetwork Communications Server configuration files . . . . .                                    | 38  |
| 3. Hardware requirements for CICS Universal Client for OS/2. . . . .                                     | 15  | 10. CICS Universal Client keys that can be mapped. . . . .   | 71  |
| 4. Software requirements for CICS Universal Client for OS/2. . . . .                                     | 16  | 11. Sample problem reporting sheet   | 109 |
| 5. Matching definitions: CICS Universal Clients to CICS for OS/2 using NetBIOS. 28                       |     | 12. CICS Transaction Gateway and CICS Universal Clients books and file names .                     | 145 |
| 6. Matching definitions: CICS Universal Clients to CICS for OS/2 using NetBIOS. 29                       |     |  |     |
| 7. Matching definitions: CICS Universal Client for OS/2 to CICS for OS/2 using TCP/IP for OS/2 . . . . . | 32  |  |     |



---

## Summary of changes

---

### Changes for IBM CICS Universal Clients Version 3.1

The following are the main new functions added for CICS Universal Clients Version 3.1.

- A graphical user interface (GUI) utility that makes configuration of CICS Universal Clients easier.
- A conversion utility, CTGCONV to convert configuration files from existing CICS Universal Clients to the new format of the configuration file.
- Enhanced security functions.
- Improved trace facilities including a utility, CICSFTRC, that converts the binary trace file to ASCII format.



---

## About this book

This book contains the following chapters:

- Chapter 1 introduces the IBM CICS Universal Clients and summarizes the benefits of using them, and the functions they provide.
- Chapter 2 discusses the various ways in which CICS Universal Clients can be connected to CICS servers. It also lists the hardware, software, and communication protocols required.
- Chapter 3 describes how to install your CICS Universal Clients.
- Chapter 4 summarizes how to set up your CICS Universal Clients for communication with CICS servers.
- Chapter 5 describes how to configure your CICS Universal Clients using the configuration tool. This chapter also describes how to customize your CICS Universal Clients by remapping the keyboard and tailoring the 3270 screen color attributes.
- Chapter 6 describes how to use the facilities supplied with CICS Universal Clients to control the client process, 3270 terminal emulation, and client printer support.
- Chapter 7 describes problem determination for CICS Universal Clients. This chapter describes CICS Universal Clients trace and error processing, and discusses communication problems and how to deal with them.
- Chapter 8 describes how to install and distribute CICS Universal Client for OS/2 around your network.
- Chapter 9 discusses some migration issues relevant to using CICS Universal Client applications.
- Appendix A discusses data conversion between CICS Universal Clients and CICS servers.
- Appendix B describes how to view the online information in the CICS Universal Clients library, and how to print and order books.

---

## Who should read this book

This book is intended for anyone involved with the planning, installation, customization, or operation of a CICS Universal Clients system.

It is assumed that you are familiar with the operating system under which your CICS Universal Clients run.

---

## Conventions and terminology used in this book

In this book the syntax of commands is shown in a standard way. This syntax, commonly known as *railroad* syntax, is described in Table 1. You interpret the syntax by following the arrows from left to right, and from top to bottom, along the main path line.

Table 1. Command syntax conventions



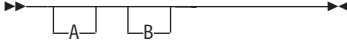


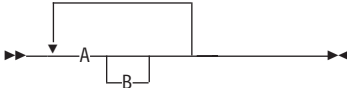
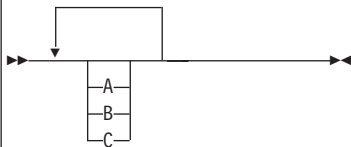
Symbol	Meaning
	Required items appear on the main path line.
	If there is more than one required item to choose from, the items are stacked vertically. This is a set of alternatives—one of which you <b>must</b> specify.
	Optional items appear below the main path line.
	If there is more than one optional item to choose from, the items are stacked vertically below the main path line. This is a set of alternatives—one of which you may specify.
	If one item in a set of alternatives is the default, this item appears above the main path line and all other items are stacked vertically below the line.
	An arrow returning to the left above items on the main path line means that the items can be repeated. Such items may be either required or optional.



Table 1. Command syntax conventions (continued)

Symbol	Meaning
	<p>An arrow returning to the left above a set of items means that more than one item can be selected or that a single item can be repeated.</p> <p><b>Note:</b> For CICS, unless otherwise stated, this representation means only that more than one item can be selected.</p>
Punctuation and uppercase characters	Code exactly as shown.
Lowercase characters appearing like <i>this</i>	Code your own text, as appropriate. For example, with FILE( <i>name</i> ) you must specify FILE and () unchanged, but are free to specify any valid text string for the name of your file.

A glossary is provided at the back of this book.

For simplicity, 'C' is used to refer to both the C and C++ programming languages throughout this book.

In this book, 'CICS on open systems' is used to refer to the following CICS server products, subject to availability:

- CICS for HP-UX
- CICS for Sun Solaris
- TXSeries for AIX
- TXSeries for HP-UX
- TXSeries for Solaris.

'CICS Client for Windows Family' refers to CICS Universal Client for Windows 98, and CICS Universal Client for Windows NT collectively. Similarly, 'Windows Family' refers to the Windows 95, Windows 98, and Windows NT<sup>®</sup> operating systems collectively.

---

## Prerequisite and related information

The following sections list books relevant to CICS Universal Clients.

### CICS Universal Clients publications

For information on the books available, refer to “Appendix B. The CICS Transaction Gateway and CICS Universal Clients library” on page 143. That chapter also gives details of how to view and print the softcopy books supplied with CICS Universal Clients and how to order printed copies from IBM.

### CICS publications

The books in this section are concerned with CICS server products.

#### CICS inter-product communication

The following books describe the intercommunication facilities of the CICS server products:

*CICS Family: Inter-Product Communication*, SC33-0824

*CICS for OS/2 Intercommunication*, SC33-1583

*Transaction Server for Windows NT: Intercommunication*, SC33-1882

*CICS Transaction Server for OS/390 CICS Intercommunication Guide*, SC33-1695

*CICS for VSE/ESA Intercommunication Guide*, SC33-0701

*Communicating from CICS for MVS/ESA<sup>®</sup> and CICS for VSE/ESA*, SC33-0825

*CICS for OS/400: Intercommunication*, SC33-1388

*TXSeries: CICS Intercommunication Guide*, SC09-3900

The first book above is a CICS family book containing a platform-independent overview of CICS inter-product communication.

#### CICS problem determination books

The following books describe the problem determination facilities of the CICS server products:

*CICS for OS/2 Problem Determination*, SC33-1584

*Transaction Server for Windows NT: Problem Determination*, SC33-1883

*CICS Transaction Server for OS/390<sup>®</sup> Problem Determination Guide*, GC33-1693

*CICS for VSE/ESA Problem Determination Guide*, SC33-0716

*CICS for OS/400:<sup>®</sup> Problem Determination*, SC33-1384

*TXSeries: CICS Problem Determination Guide*, SC33-1565.

You can find information on CICS products at the following Web site:

<http://www.ibm.com/software/ts/cics/>

## **NetBIOS-related publications**

The following books are concerned with products providing NetBIOS:

*Network Transport Services/2: LAN Adapter and Protocol Support Configuration Guide*, S96F-8489

*LAN Technical Reference IEEE 802.2 and NetBIOS Application Program Interfaces*, SC30-3587

## **TCP/IP-related publications**

The following books are concerned with products providing TCP/IP:

### **IBM products**

*IBM Transmission Control Protocol/Internet Protocol Version 2.0 for OS/2: Installation and Administration*, SC31-6075

*IBM Transmission Control Protocol/Internet Protocol Version 2.0 for OS/2: Installation and Interoperability*, GG24-3531

*TCP/IP for MVS, VM, OS/2 and DOS Troubleshooting Guide*, GG24-3852.

## **APPC-related publications**

The following books are concerned with products providing APPC support:

### **IBM products**

#### **IBM eNetwork Communications Server for OS/2 Warp:**

*IBM eNetwork Communications Server for OS/2 Warp Version 5.0: Quick Beginnings*, GC31-8189

*IBM eNetwork Communications Server for OS/2 Warp Version 5.0: Network Administration and Subsystem Management Guide*, SC31-8181

*IBM Communications Server for OS/2 Warp; Version 4.1: 32-Bit APPC Programming Guide and Reference*, SC31-8148

*IBM eNetwork Communications Server for OS/2 Warp Version 5.0: Problem Determination Guide*, SC31-8186

*IBM eNetwork Communications Server for OS/2 Warp Version 5.0: Response File Reference*, SC31-8187

### **Systems Network Architecture (SNA)**

*SNA Formats*, GA27-3136

*Systems Network Architecture Technical Overview*, GC30-3073

## Configuration, distribution, installation publications

The following books provide information on configuration, distribution, and installation (CID):

*NetView Distribution Manager/2: Concepts and Overview Version 2.0*,  
GH19-4009

*IBM Start/2 User's Guide*, S96F-8398

*SystemView for OS/2 Up and Running!*, SH19-4184

## Novell NetWare documentation

The use of Novell's NetBIOS Emulator in an OS/2 Requester is described in:

*NOVELL: NetWare Requester for OS/2*

*NOVELL: NetWare Version 3.11 Installation*

*NOVELL: NetWare Version 4.2 Installation*

*Coexistence Guide for OS/2 LAN Server and NetWare from IBM*, S96F-8311

---

## How to send your comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this book, or any other CICS documentation:

- Visit our Web site at:

<http://www.ibm.com/software/ts/cics/>

and follow the **Library** link to our feedback form.

Here you will find the feedback page where you can enter and submit your comments.

- Send your comments by e-mail to [idrcf@hursley.ibm.com](mailto:idrcf@hursley.ibm.com)
- Fax your comments to:

+44-1962-870229 (if you are outside the UK)  
01962-870229 (if you are in the UK)

- Mail your comments to:

Information Development  
Mail Point 095  
IBM United Kingdom Laboratories  
Hursley Park  
Winchester

Hampshire  
SO21 2JN  
United Kingdom

Whichever method you use, ensure that you include:

- The name of the book
- The form number of the book
- If applicable, the version of the product
- The specific location of the text you are commenting on, for example, a page number or table number.

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

---

## Obtaining books from IBM

For information on books you can download, visit our Web site at:

<http://www.ibm.com/software/ts/cics/>

and follow the **Library** link.

You can order hardcopy books:

- Through your IBM representative or the IBM branch office serving your locality.
- By calling 1-800-879-2755 in the United States.
- From the Web site at:

<http://www.elink.ibm.link.ibm.com/pbl/pbl>



---

## Chapter 1. Overview

**This chapter contains the following:**

- “An introduction to IBM CICS Universal Clients”
- “CICS Universal Clients communication” on page 2
- “The benefits of using CICS Universal Clients” on page 3
- “CICS Universal Clients functions” on page 3.

---

### An introduction to IBM CICS Universal Clients

The IBM CICS Universal Clients family comprises workstation products that bring the advantages of client/server operation to your transaction processing. CICS Universal Clients allow users to access transactions and programs on the entire family of CICS application servers.

The CICS Universal Clients family comprises:

- IBM CICS Universal Client for OS/2
- IBM CICS Universal Client for Windows 98
- IBM CICS Universal Client for Windows NT
- IBM CICS Universal Client for AIX®
- IBM CICS Universal Client for Solaris

CICS Universal Client for Windows 98 runs on Windows 95 and Windows 98.

The CICS Universal Clients can communicate with the following CICS server products, which are all Year 2000 ready:

- CICS Transaction Server for OS/2 Warp Version 4.0, and higher
- Transaction Server for Windows NT Version 4.0
- CICS Transaction Server for Windows NT Version 4.1
- ‘CICS on open systems’, that is:
  - CICS for HP-UX Version 2.1.1
  - CICS for Sun Solaris. Version 2.1.1
  - TXSeries for AIX Version 4.2, and higher
  - TXSeries for HP-UX Version 4.2, and higher
  - TXSeries for Solaris Version 4.2, and higher
- CICS for OS/400 Version 3 Release 1, and higher
- CICS for MVS/ESA Version 4 Release 1
- CICS Transaction Server for OS/390 Release 1 and higher
- CICS Transaction Server for VSE/ESA Release 1

## An introduction to IBM CICS Universal Clients

- CICS for VSE/ESA Version 2 Release 3, and higher

---

### CICS Universal Clients communication

Your CICS Universal Clients can communicate via the following protocols:

- Network Basic Input/Output System (NetBIOS)
- Transmission control protocol/Internet protocol (TCP/IP)
- Advanced program-to-program communication (APPC)

CICS Universal Client for OS/2, CICS Universal Client for Windows 98 and CICS Universal Client for Windows NT can also communicate with CICS for MVS/ESA via the IBM TCP62 protocol, which allows SNA communication over a TCP/IP network.

Table 2 on page 10 shows the protocols that CICS Universal Clients can use to communicate with each server, and the functions supported. Support for a protocol may be provided by one or more communication software products. You can therefore use the products best suited to your network environment.

CICS Universal Clients can communicate with multiple CICS servers. You use a configuration tool to determine the settings for client operation, and identify the associated servers and protocols used for communication (see “Using the configuration tool” on page 52).

The CICS Universal Clients can communicate with a server by the following mechanisms:

- The *External Call Interface* (ECI), which enables the design of new applications to be optimized for client/server operation, with the business logic on the server and the presentation logic on the client.
- The *External Presentation Interface* (EPI), which enables modern technologies, such as graphical or multimedia interfaces, to be used with traditional 3270 CICS applications.
- The *External Security Interface* (ESI), which allows non-CICS applications to verify that a password matches that recorded by an external security manager for a specified user ID. (The ESI also allows the changing of passwords.)
- *3270 terminal and printer emulation*. This provides CICS 3270 emulation for CICS servers to which the client is connected.
- *Telnet terminal* access to CICS servers.

Object oriented programming support is provided for the ECI, EPI, and ESI;



---

### The benefits of using CICS Universal Clients

You can use CICS Universal Clients for the *presentation logic* of applications. This offloads work from the CICS servers and allows the application to exploit modern graphical or multimedia interfaces, for increased usability and productivity. Depending on the form of implementation, the existing application need not be altered at all—a protection of your investment.

Data can be exchanged directly between CICS and applications at the workstation; for example, spreadsheets. Furthermore, the power of the client applications is considerably enhanced by their ability to access applications and databases on **multiple** CICS server platforms. This ability allows CICS users on every platform to write new modern applications that use client/server operation.

Using the ECI, applications can easily be written to interface non-terminal devices, for example, image scanners and bar-code readers, with a CICS server.

Local printers attached to desktop workstations can be used for printing output sent from CICS applications.

In a distributed network you can install multiple copies of CICS Universal Client for OS/2 onto many workstations using a distributed management product, for example, IBM SystemView for OS/2, or IBM NetView Distribution Manager (NetView DM). See “Chapter 8. Distributing CICS Universal Clients in a network” on page 127 for more information.

---

### CICS Universal Clients functions

This section summarizes the functions of CICS Universal Clients.

#### 3270 terminal emulation

CICS 3270 emulation enables a client workstation to function as a 3270 display or printer for CICS applications, without needing a separate 3270 emulator product. This provides flexible client workstation capabilities without the need for extensive software outlay.

Because each client can run multiple CICS 3270 emulation sessions, the hardware required is reduced, and you can use multiple 3270 emulator sessions from one or more CICS servers, all on one workstation.

## CICS Universal Clients functions

By means of mapping files, you can customize the client emulator's screen color attributes and keyboard settings. This allows users to tailor the machine to their own preferences or, for example, to comply with company standard keyboard layouts.

CICS Universal Clients terminal definitions (with some exceptions, see "Communication between CICS Universal Clients and CICS servers" on page 10) are autoinstalled at most CICS server systems, and do not have to be predefined at the server.

### The External Call Interface

The ECI enables a non-CICS client application to call a CICS program synchronously or asynchronously, as a subroutine. The client application communicates with the server CICS program using a data area called a COMMAREA. The COMMAREA is passed to the CICS server on the call, and the CICS program typically populates it with data accessed from files or databases, which is then returned to the client for manipulation or display.

CICS Universal Clients can support an unlimited number of concurrent ECI calls to a CICS server, with no restrictions on communication protocols, functions, or whether the calls are to the same or different CICS systems.

The ECI is the recommended interface for developing new client/server applications. Its call structure easily divides the presentation logic (usually in the client) from the business logic in the CICS application, offering application designers maximum flexibility. As an example, the ECI can be used with mainframe CICS applications that are already divided into business logic (in the application-owning region) and presentation logic (in the terminal-owning region). The business logic can remain unaltered when the presentation logic is developed.

### The External Presentation Interface

The EPI enables client applications to start and converse with a legacy 3270 CICS application running on the CICS server. The CICS application sends and receives 3270 data streams (for example, a CICS basic mapping support (BMS) transaction) to and from the client application as though it were conversing with a 3270 terminal. The client application captures these data streams and, typically, displays them with a non-3270 presentation product, such as graphical user interface (GUI) or multimedia software.

The EPI is therefore a method of enhancing an existing CICS application by adding a graphical or other modern interface. The CICS application itself does not need to be altered. There is no limit on the number of concurrent EPI calls.

## The External Security Interface

The ESI enables client applications to verify that a password matches the password recorded by an external security manager for a specified user ID. The ESI also allows passwords to be changed.

The ESI is described in the *CICS Family: Client/Server Programming* book.

## 3270 client printer support

CICS 3270 client printer support is the ability to define a printer terminal on the client workstation. This enables CICS applications running on the server to direct output to the client-attached printer.

You can direct the output to a physical printer attached, for example, to the LPT1 port, or you can specify a command to process the data into a format more suitable for special-purpose printers.

CICS 3270 client printer support uses CICS 3270 emulation functions. See Table 2 on page 10 for information on which CICS servers currently support CICS 3270 emulation and hence CICS 3270 client printer support.

## Client control

CICS Universal Clients provide commands or icons to:

- **Control the client process**

You can:

- Start or stop the client process
- Turn client trace on or off
- Specify the client components to be traced
- Specify the configuration file to be used
- Set up security by specifying user IDs and passwords for a CICS server
- List connected servers
- Enable and disable the display of messages.

- **Control terminal emulation**

You can:

- Start and stop the terminal emulator
- Specify the initial transaction
- Define the terminal characteristics
- Specify the name of the keyboard and screen color mapping files
- Define the command used to process print requests
- Specify the name of a file used for appending print requests.

- **Control client printer operation**

You can:

## Client control

- Start and stop the client printer emulator
- Specify the initial transaction to be run against the client printer
- Define the printer terminal characteristics
- Define the command used to process print requests
- Specify the name of a file used for appending print requests.

---

## CICS telnet support

CICS Universal Clients provides a command, CICSTELD, to start a CICS telnet daemon. You can then use any telnet 3270 client to access a CICS server through the CICS telnet daemon.

For more information, see “CICS Client telnet terminal support” on page 102.

---

## Chapter 2. Planning before installation

**This chapter describes the following:**

- “CICS Universal Clients and your network”
- “Communication protocols” on page 13
- “Hardware requirements” on page 15
- “Software requirements” on page 16

This chapter helps you plan the installation of your CICS Universal Clients. It discusses the ways in which CICS Universal Clients can be implemented in your network and the means by which they can connect with CICS servers. It also discusses the relevant communication protocols and lists the prerequisite hardware and software.

---

### CICS Universal Clients and your network

This section discusses the various ways in which CICS Universal Clients can communicate with CICS servers, and the ways in which you can access the CICS Universal Clients code within your network.

#### Connections to CICS servers

Table 2 on page 10 shows which CICS servers the CICS Universal Clients can connect to, the protocols used, and the functions supported.

CICS Universal Client for OS/2 and CICS Universal Client for Windows NT on a LAN can connect to CICS for OS/2 using the NetBIOS protocol.

The LAN could be either an IBM, Microsoft, or a Novell NetWare LAN. It can be a token-ring or Ethernet LAN.

Your CICS Universal Clients can connect to CICS for OS/2, CICS for Windows NT, and ‘CICS on open systems’ servers using the TCP/IP protocol.

Your CICS Universal Clients can communicate with CICS application servers via APPC. This is commonly achieved through a LAN and a workstation acting as a **SNA gateway**. However, CICS Universal Client for OS/2 can communicate directly with a CICS server without a gateway workstation. This facility is provided by, for example, eNetwork Communications Server.

## CICS Universal Clients and your network

### Gateways

A gateway is a device that connects two dissimilar LANs, or that connects a LAN to a wide area network (WAN), midrange computer, or mainframe computer. A gateway device has its own processor and memory and may perform protocol conversion.

The physical links from a gateway to CICS server machines can be made in a variety of ways; for example, via token ring, SDLC, or coaxial connections. For example, a gateway workstation on a LAN can be connected directly by token ring cable to an IBM 3745 controller, the front-end processor for a mainframe.

A SNA gateway workstation attaches to a mainframe CICS or CICS for OS/400 machine and handles multiple communication sessions simultaneously. Therefore, a separate connection is not required for each CICS Universal Clients workstation. To each CICS Universal Clients workstation, it appears that they are attached directly to the CICS server, but the communication data is actually routed through one of the gateway's sessions.

SNA gateway software is required in the gateway workstation to provide gateway server functions to the clients on the LAN: ***In an IBM LAN environment***, eNetwork Communications Server is used, though a product such as IBM AIX SNA Gateway/6000 Version 2.1 could be used.

### Accessing the CICS Universal Clients

It is possible to access CICS Universal Clients on your LAN in a number of ways:

1. You can have CICS Universal Clients code on each of the workstations from which you want to use CICS.
2. You can install CICS Universal Clients code on one workstation in the LAN, and use this machine as a LAN server to the other workstations in the LAN. The client code is therefore accessed via LAN redirection. This approach may be suitable if all of the clients use the same protocols and communicate with the same CICS servers.
3. You can have CICS Universal Clients code installed on a LAN server and have individual configuration files at the other workstations in the LAN. This provides more flexibility than approach 2.

**Note:** If your CICS Universal Clients code resides on a LAN server, then problems may occur if the network drive is later detached, as the code would no longer be available.

“Chapter 8. Distributing CICS Universal Clients in a network” on page 127 describes how you can distribute CICS Universal Client for OS/2 around your network.

### Multiple protocol configurations

CICS Universal Clients can communicate with multiple servers simultaneously using the same or different communication protocols. Figure 1 shows a CICS Universal Client for OS/2 communicating simultaneously with a ‘CICS on open systems’ server using TCP/IP, a CICS for OS/2 using NetBIOS, and a CICS mainframe using APPC.

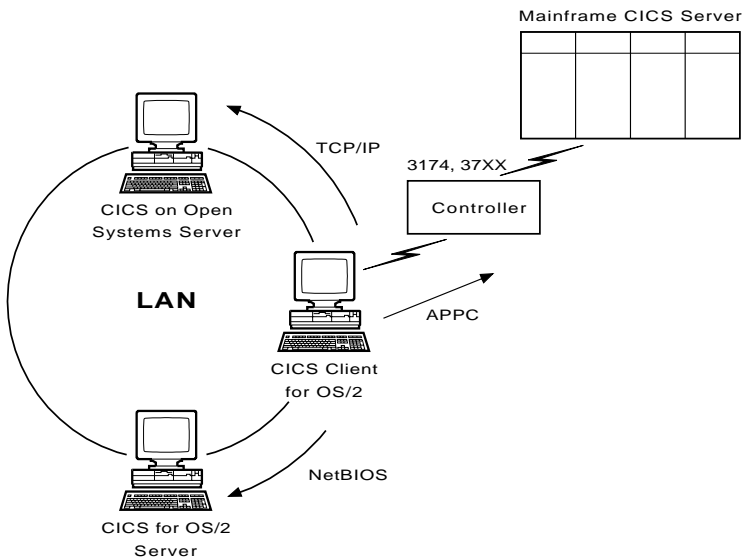


Figure 1. CICS Universal Clients communicating over NetBIOS, TCP/IP, APPC

### Mobile communication

CICS Universal Clients can communicate using mobile (wireless) communications in the same way that a CICS for OS/2 can (see Figure 2 on page 10). Mobile communication is achieved using a product such as LAN Distance<sup>®</sup> Remote for OS/2, or LAN Distance Remote for Windows on the client workstation. The LAN Distance Connection Server product is required on the gateway machine on the token-ring.

The connection from the client workstation and the connection to the server are by modem. The connection between the two modems can be either a cellular radio or a telephone connection.

## CICS Universal Clients and your network

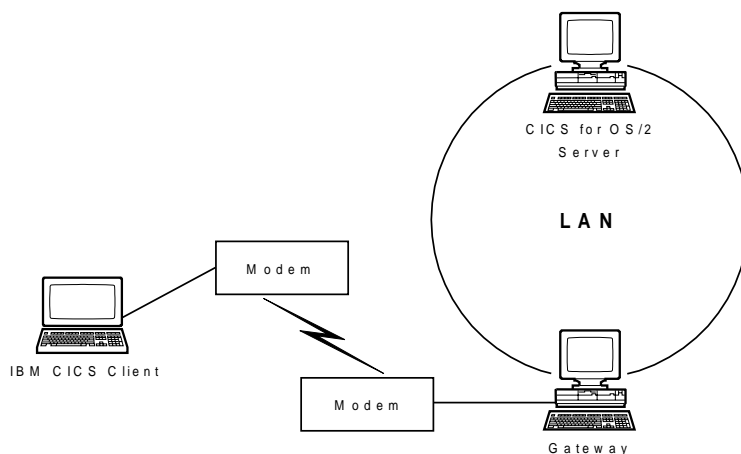


Figure 2. Mobile communication to a CICS server

### Communication between CICS Universal Clients and CICS servers

You should read the following table in conjunction with the notes that follow it:

Table 2. Communication between CICS Universal Client for OS/2 and CICS servers and functions

SERVER	NetBIOS	TCP/IP	APPC	TCP62	ECI	EPI Emulation note 1	AUTO- INSTALL note 2
CICS Transaction Server for OS/2 Version 4.1	●	●	●	○	●	●	●
Transaction Server for Windows NT Version 4	●	●	●	○	●	●	●
TXSeries Version 4.2 for Windows NT	○	●	●	○	●	●	●
'CICS on open systems'	○	●	●	○	●	●	●
CICS for OS/400 Version 3.1	○	○	●	○	●	●	●
CICS for MVS/ESA Version 4.1	○	● note 3	●	●	●	● note 4	●



Table 2. Communication between CICS Universal Client for OS/2 and CICS servers and functions (continued)

SERVER	NetBIOS	TCP/IP	APPC	TCP62	ECI	EPI Emulation note 1	AUTO- INSTALL note 2
Transaction Server for OS/390 Version 1.x	○	● note 3	●	●	●	●	●
CICS for VSE/ESA Version 2.3	○	○	●	○	●	● note 6	●
CICS Transaction Server for VSE/ESA Version 1.1	○	○	●	○	●	●	●

**Note:** ● = supported, ○ = not supported.

## Notes on table:

1. EPI always incorporates CICS 3270 terminal emulation and CICS 3270 client printer support.
2. Autoinstall means that the user does not need to predefine the client to the CICS server, that is, control table definitions are automatically created for the client at the CICS server. For CICS for VSE/ESA autoinstall is possible via LU 6.2 single session only, although this restriction does not apply to CICS Transaction Server for VSE/ESA.
3. TCP/IP communication between CICS Universal Client for OS/2, CICS Universal Client for Windows NT, CICS Universal Client for Windows 98 and CICS for MVS/ESA Version 4.1 and later is achieved using TCP62 support.
4. EPI Emulation is supported on CICS for MVS/ESA Version 4.1 with APAR PN68409.
5. APAR PN79262 is required for DBCS support in CICS for MVS/ESA.
6. EPI emulation is supported on CICS for VSE/ESA Version 2.3 with APAR PN87445 (PTF numbers UN90168 and UN90169).

## Restrictions on the use of CICS servers

The following restrictions apply:

- For CICS for OS/400, DBCS languages are not supported.
- For CICS for OS/400, the CEDA transaction cannot be started from a client terminal.
- For CICS for OS/400, you cannot use PF1 to get CICS online help from a client terminal.

## **CICS Universal Clients and your network**

- CICS for OS/400 does not support signon capable terminals.

## Communication protocols

This section discusses the protocols that can be used for client/server communication, and which protocols can be used for a particular client/server connection.

### Which protocols can you use?

The following protocols can be used for communication between CICS Universal Clients and CICS servers:

#### NetBIOS

Network Basic Input/Output System (NetBIOS) is a communication interface for the LAN environment. NetBIOS is used for connections between CICS Universal Clients and CICS for OS/2 and Transaction Server for Windows NT servers in IBM LAN Server, Novell NetWare, and other network environments.

#### TCP/IP

Transmission Control Protocol/Internet Protocol (TCP/IP) is a widely used, robust, suite of protocols, particularly important in connecting heterogeneous networks. TCP/IP originated from the UNIX<sup>®</sup> operating system, but is now supported on virtually every platform.

Also, with the **TCP62** support provided with CICS Universal Client for OS/2, CICS Universal Client for Windows 98, and CICS Universal Client for Windows NT, access is provided to CICS for MVS/ESA Version 4.1 and Transaction Server for OS/390.

**APPC** Advanced program-to-program communication (APPC) is a commonly used term for the verbs and services included in the SNA Logical Unit type 6.2 (LU 6.2). An LU 6.2 communication session provides device-independent application-to-application communication.

Also, if you install CICS Universal Client for OS/2 on the same machine as CICS for OS/2 Version 3.1, the client can communicate with one or more CICS servers using the CICS internal Multi-region Operation (MRO) protocol. For more information, refer to *CICS for OS/2 Operation*, SC33-1582..

The protocols that can be used for the various client/server connections are shown in Table 2 on page 10.

### Why use a particular protocol?

As shown in table 2 some protocols can be used only for certain types of client/server connection.

## Communication protocols

If you need to connect different types of networks, for example, Token Ring and Ethernet networks, you will probably want to use TCP/IP. Indeed, TCP/IP is assuming ever greater importance and use in Internet communication.

With the TCP62 support provided for CICS Universal Client for OS/2, CICS Universal Client for Windows 98, and CICS Universal Client for Windows NT, the CICS Universal Clients can access CICS for MVS/ESA and Transaction Server for OS/390 using the TCP/IP network. This allows partner LU62 applications to communicate without complex SNA configuration definitions on the client, and without any changes to the LU62 applications on the client or server. Therefore, if you require easy client access to CICS for MVS/ESA over your TCP/IP network you should use the TCP62 support. For information on configuring TCP62, see “TCP62 configuration” on page 35.

If you are already using a particular protocol, you may wish to continue with that protocol alone. Note however, that you can only have one SNA stack on a system. The CICS Host TCP/IP Access Feature contains a SNA stack, so this cannot be used simultaneously with an existing SNA stack on the same system.

---

## Hardware requirements

The following table lists the hardware requirements of CICS Universal Client for OS/2

*Table 3. Hardware requirements for CICS Universal Client for OS/2*

<b>Processor type</b>
Any machine capable of running the appropriate operating system and other prerequisite software.
<b>Network adapters</b>
<p>To connect workstations to the LAN, a suitable adapter such as one of the following is required:</p> <ul style="list-style-type: none"><li>• The IBM Token-Ring Network Adapter/A</li><li>• The IBM Token-Ring Network 16/4 Adapter</li></ul> <p>To connect a workstation to a Novell NetWare LAN, a suitable Token-Ring or Ethernet network adapter is required.</p>
<b>Memory requirements</b>
A minimum of 16MB
<b>Disk space requirements</b>
18MB

Software requirements

This section lists the operating systems under which your CICS Universal Clients can run and the communication software necessary to connect to CICS servers.

**Note: It is important that you consult the README file supplied with CICS Universal Clients for further information on software levels required, and possible restrictions on the use of products.**

Clients may be double-byte character set (DBCS) enabled, in which case language-specific versions of the operating system are used.

The following table lists the software requirements of CICS Universal Client for OS/2.

Table 4. Software requirements for CICS Universal Client for OS/2

<b>Operating system version</b>
CICS Universal Client for OS/2 can run under any of the following operating systems: <ul style="list-style-type: none"><li>• IBM OS/2 Version 4.0 (Warp) or later (Fixpack 6 is required for Year 2000 and Euro support)</li><li>• IBM OS/2 Warp Server Version 4.0 or later (Fixpack 35 is required for Year 2000 and Euro support).</li></ul>
<b>NetBIOS support</b>
NetBIOS support is provided by the operating system.  To use CICS Universal Client for OS/2 in a Novell NetWare LAN to communicate with CICS for OS/2, using NetBIOS emulation over IPX, the following is required: <ul style="list-style-type: none"><li>• Novell NetWare Requester for OS/2 Version 2.0.1, or later.</li></ul>
<b>TCP/IP support</b>
TCP/IP support is included in the operating system.
<b>SNA support</b>
The following is required: <ul style="list-style-type: none"><li>• IBM eNetwork Communications Server Version 4.0 or later</li></ul> For TCP62 support, IBM eNetwork Communications Server for OS/2 Warp Version 5.0 is required.
<b>SystemView for OS/2 support</b>
If you are going to install CICS Universal Client for OS/2 using SystemView for OS/2, you must have the following installed on each workstation on which you will install CICS Universal Client for OS/2: <ul style="list-style-type: none"><li>• OS/2 Warp Server</li><li>• SystemView for OS/2 client</li></ul>
<b>Application development language support</b>

Table 4. Software requirements for CICS Universal Client for OS/2 (continued)

The following language compilers are supported:

- IBM VisualAge for COBOL for OS/2 Version 2.1, or later
- IBM VisualAge for C++ for OS/2 Version 3.0, or later
- IBM VisualAge for PL/I for OS/2 Version 2.1 or later
- IBM Object REXX for OS/2
- Any other compiler compatible with the above, though only these compilers have been tested by IBM

## CICS server PTF requirements

For signon capable terminals, you must ensure that you have the appropriate PTF level applied to any CICS servers being used. You should refer to the CICS Transaction Gateway/CICS Universal Client README file for the latest details and check the PTFs for the CICS servers.

To provide complete support for timeouts, if you are using any TXSeries or Transaction Server on UNIX and Windows NT platforms, it must include the appropriate PTF level. You should refer to the CICS Transaction Gateway/CICS Universal Client README file for the latest details and check the PTFs for the CICS server.

## Code page support

Some CICS servers do not support all of the code pages that are supported by CICS Universal Clients.

If the code page of an ECI application is different from the code page of the server, data conversion must be performed at the CICS server. This would be the case with an EBCDIC CICS server such as CICS for MVS/ESA. For more information, see the CICS server documentation and *CICS Clients Unmasked*.

## Software requirements



---

## Chapter 3. Installing CICS Universal Clients

This chapter describes how to install CICS Universal Clients. The components of CICS Universal Clients that you can choose to install are:

- Program Files
- Programming Samples
- Toolkit
- PDF Documentation
- IBM Java Runtime Environment (JRE) Version 1.1.8

You can also install the Adobe Acrobat Reader if required.

You can install the CICS Universal Clients program files as a minimum, and install the other components later.

The installation processes allow you to select the language in which end-user messages are displayed.

---

### Installing CICS Universal Clients

You can install CICS Universal Clients from the supplied CD-ROM, or download and install them from the World Wide Web (WWW).

**If installing from CD-ROM**, you can install directly from the CD-ROM, from the CD-ROM drive of a remote machine, or from a network drive to which the appropriate client directory has been copied. To connect to the appropriate drive, issue a NET USE command, for example:

```
NET USE F: \\CICSCLI\INSTALL
```

where F: is the drive on the remote machine, and CICSCLI\INSTALL are the servername and netname assigned by your network administrator.

**If installing from the WWW**, you download CICS Universal Clients from the following Web site: <http://service.software.ibm.com/dl/cclient/ccli3-w>

Follow the instructions for downloading the client software. The clients are downloaded as self-extracting .EXE files.

When you have extracted the CICS Universal Clients, you will have an INSTALL.EXE program on your hard disk.

You can then proceed with installation as described in the following section.

### Installing CICS Universal Client on OS/2

This section describes the process of installing CICS Universal Client interactively. It is also possible to install CICS Universal Client using a response file, see “Automatic installation of CICS Universal Client” on page 24.

Online help is available throughout the installation process, by selecting the **Help** push button or by pressing PF1.

#### **CAUTION:**

**Be very careful if you are installing CICS Universal Client onto a workstation that already has a CICS for OS/2 server installed. Both products have some files of the same name, therefore you must ensure that the installation of CICS Universal Client does not affect the server. This is particularly important with respect to the LIBPATH setting. It is not recommended that you install both CICS Universal Client and CICS for OS/2 server on the same workstation, unless you are a software developer familiar with problems of this sort.**

**Note:** If you have a previous version of CICS Universal Client for OS/2 installed, you should either install CICS Universal Client for OS/2 Version 3.1 to the same directory as the older version, or uninstall the older version first. If you do not want to remove or replace the older version of CICS Universal Client for OS/2 you must ensure that the correct dynamic link libraries (DLL) are used for each version. If CICS Universal Client for OS/2 loads a DLL from an older version, traps will occur.

Before you start, make sure that you have at least 150 KB of free space on the drive containing the operating system. This is required by the installation program.

#### **The installation procedure**

1. Change to the drive containing the INSTALL.EXE program.
2. At the command prompt, type INSTALL, then press the Enter key. After the files are loaded, an introductory CICS Universal Client panel is displayed.
3. To proceed, click on the **Continue** push button, or press the Enter key. The Install panel is displayed.
4. If you select the check box to update the CONFIG.SYS file, the file is updated automatically as part of the installation process. Your original CONFIG.SYS file is renamed to CONFIG.BAK and is stored in the same

directory. If you do not select this check box, a CONFIG.ADD file is generated, containing the necessary updates to the LIBPATH or PATH statements in CONFIG.SYS.

5. Select the **OK** push button to continue. The Install—directories panel is displayed.
6. The list box shows the installation options that you can select. When you select one or more of these options, the **Bytes needed** field shows the amount of disk space required for installation. If there is too little space on your hard disk for the CICS Universal Client installation, an action message appears before the Install—directories panel. In this case, cancel the installation by selecting the **OK** push button. Find out which of your existing files you can archive or delete to make more space before proceeding further.

Use the push buttons as necessary:

- To display descriptions of the selected options, select **Descriptions**.
- To select all of the options, select **Select all**.
- To deselect all of the options, select **Deselect all**.

The **High-Level Directory** entry field allows you to specify a drive and directory other than the default for installation.

7. To show how much disk space is available on each drive, select the **Disk space** push button. The Disk space panel is displayed.

The list box shows the amount of available disk space for each drive, and whether installation is possible. Select a drive from the list box if required. When you return to the Install—directories panel, your selected drive is shown in the **High-Level Directory** field.

8. Select the **Install** push button to continue. An Installation progress panel is displayed.

If you select the **Stop** push button, you are asked whether you want to delete the partial system you have installed. Select **Yes** to delete the files already installed and return to the introductory CICS Universal Client panel. Then, select **Start install** from the **File** menu to start the installation again.

9. A panel allowing you to select the language in which CICS Universal Client messages are displayed appears. Choose the required language and select **OK**.
10. When installation is complete, the Installation and Maintenance panel is displayed. Select **OK**.
11. When the installation process is complete, an OS/2 desktop folder is created.

## CICS Universal Client installation on OS/2

12. If your CONFIG.SYS file has been updated, shut down the system and restart. If the file was not updated rename the CONFIG.ADD file to CONFIG.SYS before shutting down the system. (CONFIG.ADD will be in the same directory as CONFIG.SYS.)

### Installing a Java Virtual Machine (JVM)

You need a Java™ Virtual Machine (JVM) to run the configuration tool supplied with your CICS Universal Client system.

You can find an IBM Java Runtime Environment (JRE) either:

- In the JRE\OS2 directory on the CD-ROM
- At the Web site:

<http://service.software.ibm.com/asd-bin/doc/index.htm>

### Updating your CICS Universal Client system

This section describes how you update your CICS Universal Client system with new service levels. This is done using corrective service software, which allows you to service an installed CICS Universal Client system with the latest level of software.

Corrective service software is provided on CD-ROM, and is also available for downloading from the following World Wide Web site:

<http://ps.software.ibm.com/>

You can install the corrective service software:

- Interactively, by running the installation and maintenance program from the icon in the CICS Universal Client folder
- Automatically, using a response file (see “Automatic installation of CICS Universal Client” on page 24)

### The installation and maintenance program

When a CICS Universal Client is installed, the installation and maintenance program is installed along with the product software.

The icon for the installation and maintenance program is displayed in the CICS Universal Client folder. From this icon you can run the program to maintain all installed CICS Universal Client systems. The installation and maintenance program allows you to:

- Install new CICS Universal Client systems, and additional CICS Universal Client components
- Update installed CICS Universal Client systems
- Restore backed up CICS Universal Client systems

- Delete installed CICS Universal Client systems

You can update from CD-ROM, or from a local or redirected LAN drive.

### The update process

The installation and maintenance program works with catalogs of products. A catalog contains a list of products; when you open a catalog, the installation and maintenance program lists the products that it contains. Before you update CICS Universal Client, you must open the catalog for CICS Universal Client and select the required product.

The process is as follows:

1. Click on the **Installation Utility** icon in the CICS Universal Client folder. The installation and maintenance program starts, and displays the main panel.
2. Select the **File** action, followed by the **Open catalog** option from the pulldown menu.
3. Now, select the **Drive** option. The Open drive catalog panel is displayed and this allows you to select the drive containing the corrective service software. When you select the drive, the catalog file (for example, CCLCLNT.ICF) is listed under **Filename**. Select the appropriate drive and open the catalog. The products in the catalog are listed.
4. Make sure that the correct service level of CICS Universal Client is selected in the list of products.  
Make sure that you will have enough disk space for the product components you want to update, and any that you want to back up.
5. Select the **Update** action from the **Action** menu. The Update panel is displayed.
6. If you select the check box to update the CONFIG.SYS file, the file is updated automatically as part of the installation process. Your original CONFIG.SYS file is renamed to CONFIG.BAK and is stored in the same directory. If you do not select this check box, a CONFIG.ADD file is generated containing the necessary updates to the LIBPATH statement in the CONFIG.SYS file.
7. If you select the **Save a backup version** check box, the system being replaced by the application of the corrective service software is backed up. The backed up system can later be restored to the previous service level by selecting the **Restore** option from the **Action** menu. This gives you the chance to check that the updated system works correctly before you delete the old system.
8. Select the **Update** push button. The installation and maintenance program unpacks and copies the modules from the corrective service software to the installed system.

## CICS Universal Client installation on OS/2

Updating an installed CICS Universal Client system causes the configuration file, keyboard mapping file, and color mapping file, to be overwritten. Although these files are backed up, it is recommended that you keep your own customized versions of these files with different names to those supplied. You should then reference these customized files by environment variables see “Renaming the client configuration and mapping files” on page 51.

### Deleting a CICS Universal Client system

To delete a CICS Universal Client system, first make sure it is not running then perform the following:

1. Use the Installation Utility to open the catalog used to install the system. If you use the catalog you used for updating, only the files transferred from the corrective service software are deleted.
2. Select the **Delete** option from the **Actions** menu. The Delete panel is displayed.
3. Select the components you want to delete. To select or deselect all of the components, click on **Select** or **Deselect** respectively. If you want to delete only the backup versions, select the **Delete only the backup versions** check box.
4. Select the **Delete** push button. A progress panel is displayed, showing the proportion of files already deleted. You can select the **Stop** push button to stop the deletion.

**Attention:** When you mark components for deletion it is not possible to delete some files immediately, because they are in use. You **must** therefore shut down and restart the system. The files are then deleted when the system is restarted. Also, any files that you added after installation are not deleted.

### Automatic installation of CICS Universal Client

You can perform automatic installation and maintenance of CICS Universal Client by supplying the required options in a **response file**. It is therefore possible to install CICS Universal Client on workstations without user interaction being required at the workstation.

This kind of automatic installation is particularly useful for installing CICS Universal Client over a network because installation can be performed from a redirected drive on a LAN server. (Refer to the documentation for your network operating system if you need to know how to connect to network drives.)

To start the installation, call the installation program by entering **INSTALL** together with the required parameters. For example:

```
INSTALL /A:I /R:F:\INSTCICS\CICSCLI.RSP  
/S:F:\INSTCICS  
/L1:F:\INSTCICS\CICSCLI.OUT  
/L2:F:\INSTCICS\CICSCLI.HIS /X
```

In this example, CICS Universal Client is installed from the redirected drive F: on the server, according to the options supplied in the response file (.RSP). The response file specifies the drive and directory in which to install CICS Universal Client. Errors are logged in the .OUT file and the history log is contained in the .HIS file. The /X parameter specifies that the installation is noninteractive.

**Note:** The base level of CICS Universal Client is installed using the INSTALL command, while corrective service software is installed using the EPFINSTS command.

The installation and maintenance parameters are described in “The installation and maintenance parameters” on page 128.

---

### Other actions required in setting up a CICS Universal Client

This section describes what you can do after the client software has been installed, and before you configure and start the client.

If you want the CICS terminal emulation to be started automatically when OS/2 starts up, create a shadow of the CICS Terminal icon and place it in the OS/2 startup folder.





---

## Chapter 4. Setting up client/server communication

**This chapter describes the following:**

- “NetBIOS configuration” on page 29
  - “TCP/IP configuration” on page 30
  - “APPC configuration” on page 37
  - “CICS Universal Clients in Novell NetWare LANs” on page 39
- This describes how CICS Universal Clients can communicate over NetBIOS in a Novell NetWare LAN.

This chapter discusses how to set up the communication links between CICS Universal Clients and the CICS servers.

The NetBIOS, TCP/IP, or APPC protocols can be used to enable clients and servers to communicate. See “Chapter 2. Planning before installation” on page 7 for details of what levels of the various protocols are required.

This chapter summarizes the products and definitions that are required on CICS servers. Full details of setting up a CICS server for communication are given in the individual CICS server books, see “CICS publications” on page xvi.

After the client has been installed, and the server has been set up for communication, the following steps are necessary:

- Enable and set up NetBIOS, TCP/IP, or APPC communication
- Set up the required settings in the configuration file, (see “Using the configuration tool” on page 52)

---

### Client/server definitions

This chapter includes definition checklists for each CICS Universal Client. Definition checklists show the definitions required to allow client/server communication using a particular product and protocol:

- Definitions required for communication products at the CICS server.
- Definitions required on the CICS server.
- Definitions required for the communication product at the CICS client.
- Definitions required in the client configuration file.

## Setting up client/server communication

- Example values for the definitions.

As far as possible, the example values are based on the values in the supplied configuration file, ctg.ini. See “Using the configuration tool” on page 52 for more information.

- The definitions that must match.

The following is a simple example of a set of definition checklists. It shows the definitions required to connect a CICS Universal Client to a CICS for OS/2 server. ***The example values are samples only—you must enter your own values.*** However, in the checklists in this chapter, the values shown for DriverName in the configuration file are the actual driver names that you must use.

### Definition checklists

#### CICS server definitions

CICS for OS/2 System Initialization Table Definitions	Ref. Key	Example
Local System Appl ID	<b>1</b>	CICSOS2
NetBIOS Listener Adapter	<b>2</b>	0
Maximum NetBIOS Systems		15

#### CICS Universal Client definitions

Client Configuration settings	Ref. Key	Example
Server name		CICSNETB
Network Protocol		NETBIOS
Netname	<b>1</b>	CICSOS2
Adapter	<b>2</b>	0

Table 5. Matching definitions: CICS Universal Clients to CICS for OS/2 using NetBIOS

Ref. Key	CICS for OS/2	Client Configuration Settings	Example
<b>1</b>	Local System Appl ID	Netname	CICSOS2
<b>2</b>	NETBIOS Listener Adapter	Adapter	0

In the above tables the definitions that must match are indicated by reference keys, **1**, **2**, and so on. ***Definitions with the same reference key must match.***

This book does not give details for every possible client/server configuration. Typical examples are given representing the most likely configurations. More example configurations are described in the *Revealed! CICS Transaction Gateway*

with more *CICS Clients Unmasked* book, and also in a set of special configuration documents, see “Sample configuration documents” on page 145.

### NetBIOS configuration

This section describes how to set up CICS Universal Clients for communication via the NetBIOS protocol.

All communication between a client and server is multiplexed on one NetBIOS session.

NetBIOS support is provided in the OS/2 operating system and no further configuration is required after installation.

### Definition checklists

The following shows example definitions for connecting CICS Universal Client for OS/2 to a CICS for OS/2 server:

#### CICS server definitions

CICS for OS/2 System Initialization Table Definitions	Ref. Key	Example
Local System Appl ID	<b>1</b>	CICSOS2
NetBIOS Listener Adapter	<b>2</b>	0
Maximum NetBIOS Systems		15

#### CICS Universal Client definitions

Client Configuration settings	Ref. Key	Example
Server name		CICSNETB
Network Protocol		NETBIOS
Netname	<b>1</b>	CICSOS2
Adapter	<b>2</b>	0

Table 6. Matching definitions: CICS Universal Clients to CICS for OS/2 using NetBIOS

Ref. Key	CICS for OS/2	Client Configuration Settings	Example
<b>1</b>	Local System Appl ID	Netname	CICSOS2
<b>2</b>	NETBIOS Listener Adapter	Adapter	0

## NetBIOS configuration

### NetBIOS resources

The CICS Universal Client requirement for NetBIOS resources is affected by the **Maximum servers** setting in the configuration file. The resource requirements must therefore be increased as follows:

<b>NetBIOS Names</b>	1
<b>NetBIOS Commands</b>	Value of Maximum servers + 1
<b>NetBIOS Sessions</b>	Value of Maximum servers.

For CICS Universal Client for OS/2, the NetBIOS resource requirements are coded in the \IBMCOM\PROTOCOL.INI file. PROTOCOL.INI is the LAN adapter and protocol support (LAPS) configuration file.

“CICS Universal Clients in Novell NetWare LANs” on page 39 describes how CICS Universal Clients can communicate over NetBIOS in a Novell NetWare LAN.

---

## TCP/IP configuration

This section describes how to set up CICS Universal Clients for communication via the TCP/IP protocol. To configure TCP/IP for communication with CICS servers, you need to know the following:

- The IP address/host name of the CICS Universal Client
- The IP address/host name of the CICS server
- The IP address/name of any domain nameserver
- The subnet mask (if required)
- The port number of the CICS server

If you are using a domain nameserver you can specify symbolic hostnames rather than IP addresses. Hostnames are easier to remember, and can be made meaningful.

Your system administrator can supply this information, which is also used when you use the configuration tool.

**Note:** In the \ETC\SERVICES file, you can enter CICS in either uppercase or lowercase.

The following sections summarize TCP/IP configuration for CICS Universal Client for OS/2.

## IBM TCP/IP for OS/2

For information on installing and configuring IBM TCP/IP for OS/2, see, for example, the *IBM TCP/IP Version 2.0 for OS/2 Installation and Maintenance* book.

### Definition checklists

The following shows example definitions for connecting CICS Universal Client for OS/2 to a CICS for OS/2 server:

#### CICS server definitions

TCP/IP Definitions (Server)	Ref. Key	Example
IP address of workstation	<b>1</b>	9.113.36.31
IP address of Domain Name Server	<b>4</b>	9.113.42.240
IP address of router		9.113.36.254
Subnet Mask		255.255.254.0
Hostname	<b>2</b>	cicstcp.ibm.com
Port number	<b>3</b>	0

CICS for OS/2 System Initialization Table Definitions	Ref. Key	Example
TCP/IP Local Host Name	<b>1</b> or <b>2</b>	*
TCP/IP Local Host Port	<b>3</b>	*
Maximum TCP/IP Systems		15
<b>Note:</b> You can enter values for the TCP/IP Local Host Name and TCP/IP Local Host Port, or use the default values of *. See <i>CICS for OS/2 Customization</i> for more information.		

#### CICS Universal Client definitions

TCP/IP Definitions (Client)	Ref. Key	Example
IP address of workstation		9.113.36.25
IP address of Domain Name Server	<b>4</b>	9.113.42.240
Subnet Mask		255.255.254.0
IP address of router		9.113.36.254

## TCP/IP configuration

Client Configuration Settings	Ref. Key	Example
Server name		CICSTCP
Network Protocol		TCP/IP
Host name or IP address	<b>1</b> or <b>2</b>	cicstcp.ibm.com
Port	<b>3</b>	0

Table 7. Matching definitions: CICS Universal Client for OS/2 to CICS for OS/2 using TCP/IP for OS/2

Ref. Key	TCP/IP on Server	CICS for OS/2 on Server	TCP/IP on Client	Client Configuration Settings	Example
<b>1</b>	IP address	TCP/IP Local Host Name		IP address	9.113.36.31
<b>2</b>	Hostname	TCP/IP Local Host Name		Host name	cicstcp.ibm.com
<b>3</b>	Port number	TCP/IP Local Host Port		Port	0
<b>4</b>	IP address of Domain Name Server		IP address of Domain Name Server		9.113.42.240

The following shows example definitions for connecting CICS Universal Client for OS/2 to a TXSeries for AIX server:

### CICS server definitions

TCP/IP for AIX/6000 Definitions	Ref. Key	Example
HOSTNAME	<b>1</b>	cicstcp.ibm.com
Internet ADDRESS	<b>2</b>	192.113.36.200
Network MASK	<b>3</b>	255.255.255.0
Client Hostname	<b>4</b>	myhost
Client Address	<b>5</b>	192.113.36.78
TCP/IP service name	<b>6</b>	cicstcp
Port number	<b>7</b>	0
Protocol type (TCP only)		TCP

TXSeries for AIX Server Definitions	Ref. Key	Example
Listener Identifier		anyname
Protocol type (TCP only)		TCP
TCP adapter address	<b>2</b>	192.113.36.200
TCP service name	<b>6</b>	cicstcp

### CICS Universal Client definitions

TCP/IP Definitions	Ref. Key	Example
Host name	<b>4</b>	myhost
IP Address	<b>5</b>	192.113.36.78
Subnet Mask	<b>3</b>	255.255.255.0
Server IP Address	<b>2</b>	192.113.36.200
Server HostName	<b>1</b>	cicstcp.ibm.com

Client Configuration Settings	Ref. Key	Example
Server name		CICSTCP
Network Protocol		TCPIP
Host name	<b>1</b>	cicstcp.ibm.com
IP address	<b>2</b>	192.113.36.200
Port	<b>7</b>	0

Table 8. Matching definitions: CICS Universal Client for OS/2 to TXSeries for AIX using TCP/IP

Ref. Key	TCP/IP for AIX/6000	TXSeries for AIX	TCP/IP on Client	Client Configuration Settings	Example
<b>1</b>	HOSTNAME		Server Hostname	Host name	cicstcp.ibm.com
<b>2</b>	Internet ADDRESS	TCP adapter address	Server IP Address	IP address	192.113.36.200
<b>3</b>	Network MASK		Subnet Mask		255.255.255.0
<b>4</b>	Client Hostname		Hostname		myhost
<b>5</b>	Client Address		IP Address		192.113.36.78
<b>6</b>	TCP/IP service name	TCP service name			cicstcp
<b>7</b>	Port number			Port	0

## TCP/IP configuration

### Verifying the TCP/IP installation

To verify that CICS Universal Clients can communicate with CICS servers, you can use the TCP/IP PING command to check the route to the CICS server:

```
ping [machine address | name]
```

Therefore, to start PING, enter a command like the following:

```
ping 192.113.36.200
```

where 192.113.36.200 is an example of an IP address of the CICS server. If you are using a Domain Name Server, you can specify the symbolic hostname rather than the IP address of the server.

To stop the PING command, press the Ctrl+C keys simultaneously.

If you have installed TCP/IP correctly, you see messages as shown in the following figure:

```
ping 192.113.36.200 56 3
PING 192.113.36.200: 56 data bytes
64 bytes from 192.113.36.200: icmp_seq=0 ttl=255 time=3 ms
64 bytes from 192.113.36.200: icmp_seq=1 ttl=255 time=3 ms
64 bytes from 192.113.36.200: icmp_seq=2 ttl=255 time=3 ms

----192.113.36.200 PING Statistics----
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 3/3/3 ms
#
```

*Figure 3. Example messages returned by ping command*

If the statistics message shows a value other than 0% packet loss, it is likely that you did not install or customize TCP/IP correctly. Therefore:

- Check for TCP/IP definition errors
- Check for TCP/IP definition mismatches
- Check the network physical connection

Depending on the platform, the implementation of the PING command might be slightly different, regarding the operation of the utility (icon or command) and the parameters that can be used. For more information, refer to the documentation associated with your TCP/IP product.



---

## TCP62 configuration

The TCP62 support for CICS Universal Clients allows communication with CICS for MVS/ESA Version 4.1 and later over a TCP/IP network.

On CICS for MVS/ESA you can use autoinstall to define APPC connections to clients dynamically, on their first use. On CICS for MVS/ESA Version 4.1 and CICS Transaction Server for OS/390, you can autoinstall parallel-session APPC connections. The advantage of autoinstall is that it allows you to use the same client configuration settings for all workstations without having to define multiple entries in VTAM<sup>®</sup> and CICS.

CICS Universal Client TCP62 communication only supports parallel-session APPC connections, not single-session connections.

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

Enabling CICS for MVS/ESA to communicate with a CICS Universal Client using TCP62 requires actions on OS/390, CICS, VTAM, and the client workstation as summarized in the following sections.

For detailed information on configuring TCP62, refer to the *Revealed! CICS Transaction Gateway with more CICS Clients Unmasked* book and to any configuration document that is available, see “Sample configuration documents” on page 145.

### On OS/390

Any of the VTAM AnyNet<sup>®</sup> releases must be installed. AnyNet originated as the Multi-Protocol Transport Feature (MPTF) with VTAM Version 3 Release 4.2 and this feature included support for APPC over TCP/IP.

With VTAM Version 4 Release 2, it was retitled “VTAM V4 R2 AnyNet Feature”. Similarly with VTAM Version 4 Release 3, AnyNet was available as the “VTAM V4 R3 AnyNet Feature”. These features all require MVS/ESA SP Version 3 Release 1.3 or later and TCP/IP Version 2 Release 2.1 or later. MPTF V3 R4.2 and “AnyNet V4 R2” also require C/370 Version 2, but AnyNet V4 R3 requires Language Environment for MVS.

With VTAM Version 4 Release 4, which is also shipped as part of OS/390 Release 3, AnyNet has been integrated with VTAM and is no longer a separate feature. VTAM Version 4 Release 4 requires MVS/ESA SP Version 4 Release 3, LE Version 1 Release 3 and TCP/IP Version 3 Release 1.

## TCP62 configuration

On OS/390, you must:

1. Install a TCP major node, which 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* book, SC31-6527.
2. Install a CDRSC major node, which 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.

### 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 configuration file. (The default modename in the configuration file is TCP62.)
  - On the MAXIMUM option of the SESSIONS definition, specify the second value as one, that is, that CICS for MVS/ESA is to have one contention winner. For example, MAXIMUM(8,1) means that the modeset is to support eight sessions, and that CICS for MVS/ESA has one contention winner.
2. Add an entry to the VTAM logon mode (LOGMODE) table for the modename specified on the SESSIONS definition. This entry specifies the class of service required for the group of sessions.

### On the client workstation

On the client workstation, you must install and configure TCP/IP and ensure that the MVS system name is added to the TCP/IP HOSTS file.

For CICS Universal Client for OS/2, you must install TCP62 support as provided by eNetwork Communications Server for OS/2 Warp Version 5.0 or later. Use a default AnyNet response file so that the AnyNet and TCP62 support is installed. (Sample response files are supplied on the Communications Server CD-ROM.)

If eNetwork Communications Server (Kernel and SNA subsystem components) is stopped and you start a TCP62 client connection, eNetwork Communications Server is then automatically started with a new response file, together with the TCP62 definitions in the configuration file. Any TCP62 definitions in the default response file are overwritten by those supplied in the configuration file in this case.

If eNetwork Communications Server is already running and you start a TCP62 client connection, the required TCP62 definitions must be in the default eNetwork Communications Server configuration file, as those supplied in the configuration file are ignored.

To configure eNetwork Communications Server for non-automatic start:

1. Start Communications Manager Setup by selecting the **Setup** icon.
2. Open a configuration file.
3. On the Communications Manager Configuration Definitions window, select **Options**
4. Select **Configure any profile or feature**
5. Configure SNA local node characteristics—only Network ID and Local node name are required for TCP62.
6. Configure AnyNet Base Parameters—specify a SNA domain name suffix and set the default routing preference to:  

<b>Non-native only</b>	If using TCP62 only.
<b>Non-native first</b>	If using TCP62 and SNA.
7. Add SNA definitions if required.

For more information, refer to the *Revealed! CICS Transaction Gateway with more CICS Clients Unmasked* book.

### Firewall implications

You may experience some problems when configuring a TCP62 connection through a firewall. You may find that the firewall discards TCP/IP KeepAlive packets that the CICS client sends to the CICS server, which causes the session to be dropped.

You can enable or disable the sending of KeepAlive packets using the Send TCP/IP Keepalive packets setting in the configuration tool.

---

## APPC configuration

To set up CICS Universal Clients for communication via the APPC protocol, the following are generally defined:

- The **local node characteristics** that are common to all APPC users at the workstation
- A **local logical unit (LU)** definition for the client
- A **partner logical unit (PLU)** definition for each CICS server with which the client will communicate

## APPC configuration

- One or more **mode** definitions to specify sets of session properties that are used in binding APPC sessions

**Note:** The terms used to describe these definitions vary with the product used to provide APPC support. The terms used above are the ones used by IBM eNetwork Communications Server.

### Configuring APPC for CICS Universal Client for OS/2

The following sections summarize APPC configuration for CICS Universal Client for OS/2.

#### eNetwork Communications Server

A eNetwork Communications Server configuration is defined in a set of files as shown in Table 9.

*Table 9. eNetwork Communications Server configuration files*

File	Contents
<i>configname.CFG</i>	eNetwork Communications Server configuration (.CFG) file
<i>configname.NDF</i>	Node definitions (.NDF) file
<i>configname.CF2</i>	SNA configuration (.CF2) file (copy of .NDF file)
<i>configname.SEC</i>	SNA security (.SEC) file
PROTOCOL.INI	LAN Adapter and Protocol Support (LAPS) configuration file

**eNetwork Communications Server configuration (.CFG) file:** The eNetwork Communications Server configuration (.CFG) file contains definitions that specify the environment within which the workstation operates.

The .CFG file is the central configuration file and is the minimum requirement for the installation or use of eNetwork Communications Server. When the workstation is connected to a LAN or an SNA network, you need additional files as indicated in Table 9.

Configuration files are not supplied for SNA network definitions or for LAN Adapter and Protocol Support.

**Node definitions (.NDF) file:** The node definitions (.NDF) file contains the node SNA network definitions in the form of system management configuration verbs.

The following definitions are included in the .NDF file:

- Local node characteristics
- Connection to peer node or primary host
- Connection to network node

- Local logical unit
- Partner logical unit
- Mode

**SNA configuration (.CF2) file:** The SNA configuration (.CF2) file contains a binary translation of the .NDF file and is used to initialize the node SNA definitions when APPC is started.

**SNA security (.SEC) file:** The SNA security (.SEC) file contains SNA definitions for session-level and conversation-level security.

**APPC definitions:** For APPC communication, the following are defined:

- One **local node Characteristics** definition
- At least one **connection** definition
- A **local logical unit (LU)** definition
- A **partner logical unit (PLU)** definition for each CICS server with which the client will communicate
- One or more **mode** definitions to specify sets of session properties that are used in binding APPC sessions

### Detailed configuration

Detailed information on connecting CICS Universal Client for OS/2 to CICS Transaction Server for OS/390 is given in the *Revealed! CICS Transaction Gateway with more CICS Clients Unmasked* book and any configuration document that is available, see “Sample configuration documents” on page 145 .

---

## CICS Universal Clients in Novell NetWare LANs

### Note

The procedures described in this section may be not be necessary, depending on the Novell NetWare LAN support provided by the operating system you are using. If you are in doubt about this, refer to the documentation supplied with your operating system.

You can install CICS Universal Clients on workstations that are part of a Novell NetWare LAN. Such CICS Universal Clients can run transactions on a CICS for OS/2 server just as they can in an IBM LAN Server network.

## CICS Universal Clients in Novell NetWare LANs

Novell LANs normally use the Internet Packet Exchange (IPX) protocol, rather than IBM's NetBIOS for their LAN transport, but you can overcome this in two ways:

### 1. *Coexistence with Novell's IPX*

CICS Universal Clients can run on Novell NetWare workstations by making use of Novell's Open Data-Link Interface (ODI). The ODI enables a workstation to support multiple communication protocols over a *single* network adapter. Therefore by using the ODI, a workstation can run a CICS Universal Client over either NetBIOS or TCP/IP, *and* access a NetWare file server over IPX. This requires no change to the setup of the CICS for OS/2 server because it receives normal NetBIOS or TCP/IP flows.

### 2. *Novell's emulation of NetBIOS over IPX*

Novell's NetBIOS emulator enables distributed client/server NetBIOS applications to run over IPX on a Novell LAN. NetBIOS emulator flows are transmitted with IPX headers. NetBIOS emulation is available on OS/2 workstations.

CICS Universal Clients and CICS for OS/2 can use this emulator program. It means that a workstation can use IPX to access both a Novell file server and a CICS for OS/2 application server.

When a workstation is configured to use Novell's NetBIOS emulator program it will only accept IPX flows, and will not accept native NetBIOS flows. Therefore, the CICS for OS/2 server will *only* accept input from CICS Universal Client workstations that have also been configured to use Novell's NetBIOS emulator over IPX.

Both of these techniques result in the same operation for end users.

**Note:** There are different ways in which coexistence and NetBIOS emulation can be achieved. This book describes techniques for token ring networks. You may have to adapt the described techniques to your own environment.

## An overview of the Open Data-Link Interface

Novell's ODI supports multiple protocols and multiple LAN adapters in a single workstation. The implementation of ODI is shown in Figure 4 on page 41.

You can use more than one protocol on the same LAN cabling because the link support layer (LSL) part of the ODI allows the LAN driver for an adapter to service more than one protocol.

Open Data-Link Interface Network driver interface specification Support (ODINSUP) is an interface that allows the coexistence of ODI and Network

Driver Interface Specification (NDIS). ODINSUP therefore allows LAN Server or other NDIS protocols to coexist with NetWare's ODI on a network.

When OS/2 LAN Requester or Extended Services is installed with NetBIOS support, a *LAN adapter-specific* NDIS LAN driver is installed and loaded in CONFIG.SYS. For example, in an IBM token ring network, the IBMTOK.OS2 NDIS MAC (Medium Access Control) driver is installed. To allow coexistence with NetWare Requester for OS/2, the NDIS LAN driver must be replaced with TOKEN.SYS (the ODI LAN driver for IBM token ring).

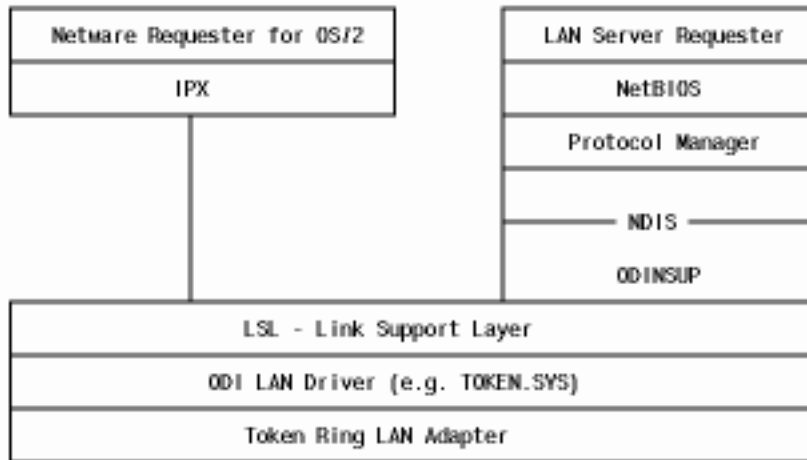


Figure 4. Implementation of Novell's ODI layer and driver

For further information on ODI, refer to the publications:

*NOVELL: NetWare Requester for OS/2.*

### Coexistence with Novell's IPX

This section describes how to set up a workstation both as a NetWare Requester and as an IBM LAN Server Requester. It assumes that the LAN Server software has already been installed.

#### NetWare OS/2 Requester running CICS Universal Client for OS/2

The following describes the procedure for a token ring network. For more information refer to *Coexistence Guide for OS/2 LAN Server and NetWare from IBM*.

1. Install NetWare Requester for OS/2 Version 2.0.1.
2. You must now configure the ODINSUP interface for token ring operation:

## CICS Universal Clients in Novell NetWare LANs

- a. First you must replace each NDIS MAC driver name with the ODI driver name. Edit the PROTOCOL.INI file to replace all occurrences of:

```
bindings = IBMTOK_nif
```

with:

```
bindings = TOKEN
```

- b. Next, you must prevent NDIS MAC drivers from loading. In CONFIG.SYS, comment out the line:

```
DEVICE=C:\IBMLAN\MACS\IBMTOK.OS2
```

- c. In CONFIG.SYS, add the line:

```
DEVICE=C:\NETWARE\ODINSUP.SYS
```

Make sure that the new ODINSUP driver comes after the line:

```
DEVICE=C:\NETWARE\ROUTE.SYS
```

- d. Now, you must enable the frame types and bind ODINSUP to your LAN driver. In the NET.CFG file, add the following lines:

```
link driver TOKEN
  frame token-ring
  frame token-ring_snap
```

```
protocol odinsup
  bind TOKEN
```

3. Restart the workstation.

This procedure results in a CONFIG.SYS (see Figure 5 on page 43) and PROTOCOL.INI file (see Figure 6 on page 44) like the following:



```
REM --- NetWare Requester statements BEGIN ---  
SET NWLANGUAGE=ENGLISH  
DEVICE=C:\NETWARE\LSL.SYS  
RUN=C:\NETWARE\DDAEMON.EXE
```

```
REM -- ODI-Driver Files BEGIN --  
DEVICE=C:\NETWARE\TOKEN.SYS  
DEVICE=C:\NETWARE\ROUTE.SYS
```

```
REM ***** Next line added manually *****  
DEVICE=C:\NETWARE\ODINSUP.SYS  
REM -- ODI-Driver Files END --
```

```
DEVICE=C:\NETWARE\IPX.SYS  
rem DEVICE=C:\NETWARE\SPX.SYS  
rem RUN=C:\NETWARE\SPDAEMON.EXE  
rem DEVICE=C:\NETWARE\NMPIPE.SYS  
rem DEVICE=C:\NETWARE\NPSEVER.SYS  
rem RUN=C:\NETWARE\NPDAEMON.EXE  
DEVICE=C:\NETWARE\NWREQ.SYS  
IFS=C:\NETWARE\NWIFS.IFS  
RUN=C:\NETWARE\NWDAEMON.EXE  
rem DEVICE=C:\NETWARE\NETBIOS.SYS  
rem RUN=C:\NETWARE\NBDAEMON.EXE  
rem DEVICE=C:\NETWARE\VIPX.SYS  
rem DEVICE=C:\NETWARE\VSHELL.SYS PRIVATE  
REM --- NetWare Requester statements END ---
```

*Figure 5. Statements added to CONFIG.SYS for coexistence*

## CICS Universal Clients in Novell NetWare LANs

```
[PROT_MAN]

    DRIVERNAME = PROTMAN$

[IBMLXCFG]

    IBMTOK_nif = IBMTOK.nif
    LANDD_nif = LANDD.NIF
    NETBEUI_nif = NETBEUI.NIF

[LANDD_nif]

    DriverName = LANDD$
; Bindings = IBMTOK_nif - This line manually replaced by next
  Bindings = TOKEN
    NETADDRESS = "08005a0347bf"
    ETHERAND_TYPE = "I"
    SYTEM_KEY = 0x0
    OPEN_OPTIONS = 0x2000
    TRACE = 0x0
    LINKS = 8
    MAX_SAPS = 3
    MAX_G_SAPS = 0
    USERS = 3
    T1_TICK_G1 = 255
    T1_TICK_G1 = 15
    T2_TICK_G1 = 3
    T1_TICK_G2 = 255
    T1_TICK_G2 = 25
    T2_TICK_G2 = 10
    IPACKETS = 250
    UIPACKETS = 100
    MAXTRANSMITS = 6
    MINTRANSMITS = 2
    TCBS = 64
    GDTS = 30
    ELEMENTS = 800
```

Figure 6. Example of *PROTOCOL.INI* after setting up coexistence (Part 1 of 2)

```

[NETBEUI_nif]

    DriverName = netbeui$
; Bindings = IBMTOK_nif - This line manually replaced by next
  Bindings = TOKEN
  NETADDRESS = "08005a0347bf"
  ETHERAND_TYPE = "I"
  USEADDRREV = "YES"
  OS2TRACEMASK = 0x0
  SESSIONS = 210
  NCBS = 250
  NAMES = 40
  SELECTORS = 5
  USEMAXDATAGRAM = "NO"
  ADAPTRATE = 1000
  WINDOWERRORS = 0
  MAXDATARCV = 4168
  TI = 30600
  T1 = 500
  T2 = 200
  MAXIN = 1
  MAXOUT = 1
  NETBIOS_TIMEOUT = 500
  NETBIOS_RETRIES = 8
  NAMECACHE = 0
  PIGGYBACKACKS = 1
  DATAGRAMPACKETS = 2
  PACKETS = 350
  LOOPACKETS = 1
  PIPELINE = 5
  MAXTRANSMITS = 6
  MINTRANSMITS = 2
  DLCRETRIES = 5
  NETFLAGS = 0x0

[IBMTOK_nif]

    DriverName = IBMTOK$
    ADAPTER = "PRIMARY"
    MAXTRANSMITS = 6
    RECVBUFS = 2
    RECVBUFSIZE = 256
    XMITBUFS = 1

```

Figure 6. Example of *PROTOCOL.INI* after setting up coexistence (Part 2 of 2)

### Novell's emulation of NetBIOS over IPX

Novell's NetBIOS emulator supports two kinds of NetBIOS: NetBIOS NB30 and NetBIOS Submit. For communication with CICS for OS/2, NetBIOS NB30 is required. For more information refer to the *NOVELL: NetWare Requester for OS/2* manual.

## CICS Universal Clients in Novell NetWare LANs

**Setting up the CICS server:** For the CICS for OS/2 system to act as the CICS server on a Novell LAN and use Novell's NetBIOS emulator, the workstation must be configured in a similar manner to an OS/2 NetWare Requester workstation, but with extra steps to allow the use of NetBIOS. The stages in the configuration process are:

1. Configure the workstation as a NetWare Requester
2. Install IBM NetWare Requester Support from LAPS
3. Bind the NetBIOS protocol to the adapter.

The process is described in the *CICS for OS/2 Intercommunication* book, and is essentially the same as for setting up CICS Universal Client for OS/2 on a Novell LAN.

**Setting up CICS Universal Client for OS/2:** You set up the client in three steps:

### Step 1—Configure the workstation as a NetWare OS/2 Requester

You must first configure the workstation as a NetWare OS/2 Requester with NetBIOS emulator support:

1. Insert the Novell *Workstation for OS/2* diskette and enter A:\install at the command prompt.
2. Select the **Requester on Workstation** option from the **Installation** pull down menu.
3. Enter a directory in the Set Target Directory panel, if you do not want the default.
4. On the Requester Installation panel, select **Edit CONFIG.SYS and Copy All Files**.
5. Enter the name of the LAN driver required, for example, TOKEN.SYS, or select a driver from the list.
6. In the next panel select **On** or **Off** for **IPX Support for DOS and Windows**. (This is not critical for NetBIOS emulation.)
7. On the Choose Optional Protocols panel, select **NetBIOS Emulation for OS/2 Sessions**. Select **On** or **Off** for **Remote Named Pipes Support**. (This is not critical for NetBIOS emulation.)
8. Select **Save** and allow the installation to complete in the usual way.
9. Ignore the instruction to restart at this stage.

The statements added to CONFIG.SYS are shown in Figure 7 on page 48.

### Step 2—Install IBM NetWare Requester Support from LAPS

The LAN adapter and protocol support (LAPS) program is used to configure the protocol to the adapter. LAPS is discussed in the *CICS for OS/2 Intercommunication* book. You can obtain LAPS from different

sources; in this technique it is obtained from IBM Network Transport Services/2 (NTS/2). (See the *NTS/2 Lan Adapter and Protocol Support Configuration Guide*.)

1. From the configure function of LAPS, display the Configure Workstation panel.
2. In the **Protocols** listbox in the Configure Workstation panel, select the **IBM OS/2 NetBIOS** and **IBM NetWare Requester Support** protocols and add them using the **Add** button. This adds the protocols to the **Current Configuration** listbox.
3. Select the **IBM NetWare Requester Support** protocol from the **Current Configuration** listbox and edit its parameters by selecting the **Edit** button. The Parameters for IBM NetWare Requester Support panel is then displayed.
4. Enter the hexadecimal address of the workstation's adapter card into the **Network Adapter Address** field. You can find this address in the LANTRAN.LOG file.
5. Complete the configuration of LAPS in the usual way.
6. Do not restart the system at this stage.

This step adds ODI2NDI.OS2, which maps the lower-level NDIS protocol APIs to the upper layer ODI APIs.

### Step 3—Bind the NetBIOS protocol to the adapter

The NetBIOS protocol added in the above steps must now be bound to the adapter as follows:

1. Edit the file C:\IBMCOM\PROTOCOL.INI by adding the following:

```
[NETBIOS]
  DriverName = NetBios$
  Bindings = IBMTOK_nif
  ADAPTER0 = IPXNB$,0
```

Remember, this example is for a token ring adapter.

An example of a PROTOCOL.INI file resulting from setting up NetBIOS emulation over IPX is shown in Figure 8 on page 49.

At this point you can restart the workstation.

## CICS Universal Clients in Novell NetWare LANs

```
REM --- NetWare Requester statements BEGIN ---
SET NWLANGUAGE=ENGLISH
DEVICE=C:\NETWARE\LSL.SYS
RUN=C:\NETWARE\DDAEMON.EXE
REM -- Following statement added by LAPS
DEVICE=C:\IBMCOM\PROTOCOL\ODI2NDI.OS2

REM -- ODI-Driver Files BEGIN --
REM DEVICE=C:\NETWARE\TOKEN.SYS
DEVICE=C:\NETWARE\ROUTE.SYS
REM -- ODI-Driver Files END --

DEVICE=C:\NETWARE\IPX.SYS
rem DEVICE=C:\NETWARE\SPX.SYS
rem RUN=C:\NETWARE\SPDAEMON.EXE
rem DEVICE=C:\NETWARE\NMPIPE.SYS
rem DEVICE=C:\NETWARE\NPSEVER.SYS
rem RUN=C:\NETWARE\NPDAEMON.EXE
DEVICE=C:\NETWARE\NWREQ.SYS
IFS=C:\NETWARE\NWIFS.IFS
RUN=C:\NETWARE\NWDAEMON.EXE
DEVICE=C:\NETWARE\NETBIOS.SYS
RUN=C:\NETWARE\NBDAEMON.EXE
rem DEVICE=C:\NETWARE\VIPX.SYS
rem DEVICE=C:\NETWARE\VSHELL.SYS PRIVATE
REM --- NetWare Requester statements END ---
```

*Figure 7. Statements added to CONFIG.SYS for NetWare Requester for OS/2 and LAPS*

```

[PROT_MAN]

    DRIVERNAME = PROTMAN$

[IBMLXCFG]

    NETBEUI_nif = NETBEUI.NIF
    ODI2NDI_nif = ODI2NDI.NIF
    IBMTOK_nif = IBMTOK.nif

[NETBEUI_nif]

    DriverName = netbeui$
    Bindings = IBMTOK_nif
    ETHERAND_TYPE = "I"
    USEADDRREV = "YES"
    OS2TRACEMASK = 0x0
    SESSIONS = 210
    NCBS = 250
    NAMES = 40
    SELECTORS = 5
    USEMAXDATAGRAM = "NO"
    ADAPTRATE = 1000
    WINDOWERRORS = 0
    MAXDATARCV = 4168
    TI = 30600
    T1 = 500
    T2 = 200
    MAXIN = 1
    MAXOUT = 1
    NETBIOS_TIMEOUT = 500
    NETBIOS_RETRIES = 8
    NAMECACHE = 0
    PIGGYBACKACKS = 1
    DATAGRAMPACKETS = 2
    PACKETS = 350
    LOOPPACKETS = 1
    PIPELINE = 5
    MAXTRANSMITS = 6
    MINTRANSMITS = 2
    DLCRETRIES = 5
    NETFLAGS = 0x0

[NETBIOS]
    DriverName = NetBios$
    Bindings = IBMTOK_nif
    ADAPTER0 = IPXNB$,0

```

Figure 8. Example of *PROTOCOL.INI* after setting up NetBIOS emulation (Part 1 of 2)

```
[ODI2NDI_nif]
  DriverName = odi2ndi$
  Bindings = IBMTOK_nif
  NETADDRESS = "10005a6ecddb"
  TOKEN-RING = "yes"
  TOKEN-RING_SNAP = "no"
  ETHERNET_802.3 = "no"
  ETHERNET_802.2 = "no"
  ETHERNET_II = "no"
  ETHERNET_SNAP = "no"
  TRACE = 0x0
```

```
[IBMTOK_nif]

  DriverName = IBMTOK$
  ADAPTER = "PRIMARY"
  MAXTRANSMITS = 6
  RECVBUFS = 2
  RECVBUFSIZE = 256
  XMITBUFS = 1
```

*Figure 8. Example of PROTOCOL.INI after setting up NetBIOS emulation (Part 2 of 2)*



---

## Chapter 5. Customization

### This chapter describes

- “Renaming the client configuration and mapping files”
- “Using the configuration tool” on page 52
- “Customizing the keyboard” on page 69
- “Customizing the screen colors” on page 74
- “Testing the client/server link” on page 78

This chapter describes how to use the configuration tool to configure your CICS Universal Clients, and also how you can customize the client keyboard mapping, screen colors, and attributes.

---

### Renaming the client configuration and mapping files

The following files are supplied with CICS Universal Clients in the Client \BIN directory by default:

<b>ctgsamp.ini</b>	A sample client configuration file. (The default configuration file name is ctg.ini.)
<b>cicskey.ini</b>	The keyboard mapping file
<b>cicscol.ini</b>	The color mapping file

It is recommended that you create your own customized versions of these files with different names, because installation of service updates may overwrite the files and cause any customization to be lost.

You should reference your customized files through the following environment variables:

File	Environment variable
<b>client configuration file</b>	CICSCLI
<b>keyboard mapping file</b>	CICSKEY
<b>color mapping file</b>	CICSCOL

You can set these environment variables in the CONFIG.SYS file for example:

```
SET CICSCLI=C:\CUSTOM\MYCLI.INI
```

## Using the configuration tool

---

### Using the configuration tool

You use the configuration tool to set configuration parameters for the CICS Universal Clients (and CICS Transaction Gateway).

The configuration tool is a Java program, therefore you must have a suitable Java Virtual Machine (JVM) installed. It is recommended that you use the IBM JRE Version 1.1.8 that is installed with your product.

To start the configuration tool, click on its icon, or enter the **ctgcfg** command.

When you start the configuration tool for the first time, a number of TaskGuides help you in setting up a new configuration. The TaskGuides let you:

1. Create CICS server definitions, and define the protocols to be used by the CICS Universal Clients
2. Configure the Workload Manager (relevant to CICS Universal Client for Windows NT and CICS Universal Client for Windows 98 only).

Default settings for the CICS Universal Clients are created.

The configuration is stored by default in the **ctg.ini** file in the **bin** subdirectory where you installed the CICS Universal Clients. You can edit this file directly, but it is recommended that you use the configuration tool to perform configuration.

The configuration file contains equivalent entries to the client initialization file (**CICSCLI.INI**) of CICS Universal Clients Version 3.0.

If a configuration file already exists when you start the configuration tool, the settings in the file are loaded into the configuration tool.

### The configuration tool interface

The user interface of the configuration tool consists of a menu bar, toolbar, tree structure, and Settings panel.

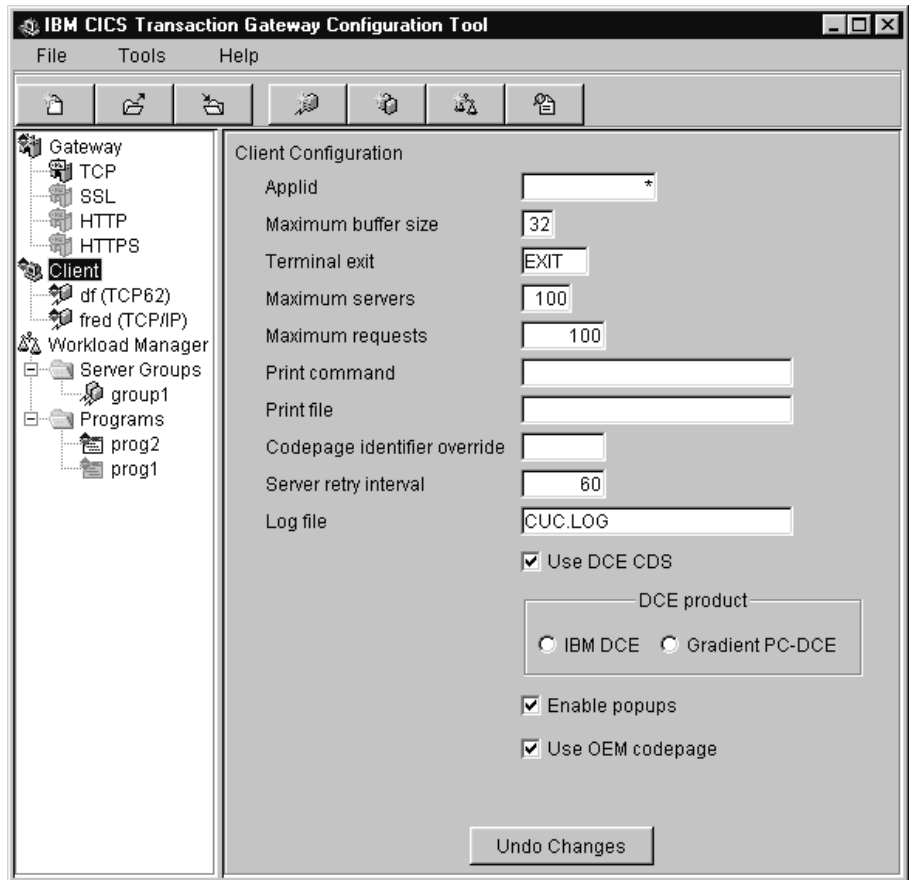


Figure 9. The configuration tool

**Note:** On your platform, the configuration tool may not appear exactly as shown in Figure 9.

### Tree structure

The tree structure (see Figure 9) allows you to navigate through all of the settings in your configuration. The types of root node are as follows:

**Gateway** Contains up to four subnodes, that is, one for each protocol the CICS Transaction Gateway can use (TCP, SSL, HTTP, and HTTPS).

If you have only a CICS Universal Client, you will not see this node.

**Client** Contains a subnode for each of your server definitions.

## Using the configuration tool

### Menu bar

The menu bar contains the **File**, **Tools**, and **Help** pulldowns.

The **File** pulldown has the following options:

<b>New</b>	Creates a new configuration.
<b>Open</b>	Opens an existing configuration.
<b>Save</b>	Saves the current configuration. If a new configuration has been created, the file name is ctg.ini (in the bin subdirectory).
<b>Save As</b>	Allows you to override the default path and name of the configuration file.
<b>Exit</b>	Exits the configuration tool. You are asked whether you want to save the configuration.

The **Tools** pulldown has the following options:

<b>Trace Settings</b>	Displays the Trace Settings dialog.
<b>New Server</b>	Starts the TaskGuide for creating new server definitions.
<b>Delete Server</b>	Deletes a server node from the tree structure.

The **Help** pulldown has the following options:

<b>Context Help</b>	Displays online help information according to the current screen context, for example, for a particular configuration setting.
<b>Contents</b>	Displays the contents list for the online help information.
<b>Index</b>	Displays a subject index for the online help information.

### Toolbar

The toolbar has the same functionality as that of the menu bar, except that it does not include the **Save As** or **Exit** options. The icons have hover helps, so that when you move the cursor over them, a text box describing the option appears.

### Settings panels

When you select nodes in the tree structure, the relevant settings panel is displayed. The settings in these panels correspond to parameters in the ctg.ini file.

On each settings panel there is an **Undo Changes** button that allows you to undo changes you have made.

## Configuring Client settings

To display the Client Settings panel, select the Client node in the tree structure. The settings map to the parameters in the Client section of the ctg.ini file.

### Applid

Enter up to 8 characters, or leave this field with the default value of \*.

This value specifies the applid of the CICS Universal Client workstation in the form in which it will be autoinstalled as a system at the CICS server. The name must be unique within the CICS server system. The value of \* automatically generates a name that is guaranteed to be unique.

**Note:** If the client is to be autoinstalled to more than one CICS server, and if you enter a specific name for the applid, that name must be unique with respect to all servers it is connected to. If the name is not unique, then attempts to connect to a server may be rejected because another client has already been installed using the same name. If a name of \* is used, the client may be known by a different unique name at each server.

If the client is to communicate with a given server via APPC, then this applid may be overridden at the time the client is installed at the server by the Local LU name for the client.

### Maximum buffer size

Enter a number of kilobytes, in the range 4 through 32. The default is 32 KB.

This value specifies the size of the transmission buffers in which application or terminal data will flow. The value should be large enough to cater for the largest possible COMMAREA or terminal input/output area (TIOA) to be used. The value does *not* include an overhead of 512 bytes needed by the clients for some protocols.

This setting need normally be specified only for clients running in a memory-constrained environment.

### Terminal exit

Enter a character string of between 1 and 4 characters. The default is EXIT.

## Using the configuration tool

The string, when entered at a terminal emulator at any time and place where a transaction name can be entered, causes the terminal emulator to terminate. The string must not contain any blank characters.

The string is case-sensitive. If a terminal emulator has uppercase translation in its CICS terminal definition, you should enter this string in uppercase.

### Maximum servers

Enter a value in the range 1 through 256. The default is 10.

This value specifies the maximum number of servers that can be accessed concurrently from the client.

For more information on the use of Maximum servers, see “NetBIOS resources” on page 30.

### Maximum requests

Enter a value in the range 1 through 10 000. The default is 90.

This value specifies the maximum number of concurrent items that may be executing on the client, when an item is defined as a terminal emulator, an EPI terminal or an ECI unit of work.

This value is used to detect runaway situations where an application could, in error, submit an excessive number of requests to a server. The actual limit may be less than this setting if other operating system limits (for example, memory constraint or communication sessions), come into effect.

### Print command

Enter a character string, from 1 to 256 characters long.

The specified string is a command specific to the operating system under which the client is running. When a request to print is received at the client, the client generates a temporary print file with a unique name per print request.

The parameter string is appended with the temporary file name, and the resultant command executed. This allows, for example, print requests to be copied to a file, directed to a local printer, formatted for inclusion into documentation, and so on.

A command file may be necessary to act as an interface between the syntax of the invocation command defined here, and more general operating system

syntax. For example, if the desired final command was COPY printfile LPT2, a simple command file would be required to reorder the parameters. It is the responsibility of the Print Command to delete the temporary print file after it has finished processing it.

See also the **Print file** description for more information.

### **Print file**

Enter a character string, 1 to 256 characters long.

This option is applicable only if the Print Command setting is omitted.

The specified string identifies a file to which output from print requests received at the client is directed. Each print request is appended to the end of the current file.

If both this setting and **Print command** are omitted, the default action is to direct the print data to LPT1.

**Note:** This setting acts only as a default. The terminal and print emulators provide options to override this value. (See “CICSTERM command reference” on page 94 and “CICSPRNT command reference” on page 99).

### **Codepage identifier override**

Enter a value indicating a Coded Character Set Identifier (CCSID) to override your local codepage identifier.

You should use this setting if your platform has been updated for Euro support, and the CICS Server has Euro support. For example, for Latin-1 countries, use a CCSID value of 858 to indicate that the codepage 850 includes Euro support. For codepage 1252, specify a CCSID value of 5348.

Note that CICSTERM will always display characters based on the local codepage of the workstation, regardless of the value specified by the **Codepage identifier override** setting.

Also note that if you use the CCSID to change the codepage identifier used, data that was stored previously in the server may be modified when retrieved to the Client, if it includes characters for which the code points produce different characters.

For more information on CCSIDs and data conversion, refer to “Appendix A. CICS Universal Clients data conversion” on page 139.

## Using the configuration tool

### Server retry interval

Enter a number of seconds. This is the time in seconds between attempts by the client to reconnect to a server to which it was connected. The default interval is 60 seconds.

When the client becomes aware that a server it was connected to is no longer active, it attempts to reconnect to the server 1 second after it becomes inactive. Subsequent retry attempts then occur at the interval defined by **Server retry interval**.

### Log file

Enter the name of the log file to be used for problem diagnosis.

If not specified, the log filename defaults to cicscli.log in the Client \BIN subdirectory.

### Enable popups

Select this check box to enable the display of popup messages at startup.

You can also change this setting dynamically by using the CICSCLI /n and CICSCLI /e commands.

## Configuring Server settings

To display the Server Settings panel, select a Server node in the tree structure. The settings map to the parameters in a Server section of the ctg.ini file.

### Server name

Enter a name of between 1 and 8 characters. This provides a communications-protocol-independent name for the server, local to the client.

Requests to access the server from ECI, EPI, ESI, or terminal emulators reference the server through this name.

### Description

Enter a description for the server of between 1 and 60 characters. This description is optional.

The description is returned to applications running on the client via the CICS\_EpiListSystems and CICS\_EciListSystems functions. (See the *CICS Family: Client/Server Programming* book).



### Initial transaction

Enter a transaction identifier of between 1 and 128 characters.

This string is case-sensitive and identifies the initial transaction (and any parameters) to be run when the terminal emulator connects to the server. If you do not enter anything, no initial transaction is run. The first four characters, or the characters up to the first blank in the string are taken as the transaction. The remaining data is passed to the transaction on its invocation.

### Model terminal definition

Enter a string of between 1 and 16 characters.

The string is case-sensitive and specifies the name of a model terminal definition at the server, identifying the characteristics of terminals to be autoinstalled from the client. If the model cannot be located at the server, or you do not enter anything, a default terminal definition is used. This default is server-specific.

The interpretation of the Model terminal definition setting is server-specific. For example, for a CICS for OS/2 server, the value is validated (by the server) as a 1- to 4-character string identifying a model entry in the CICS terminal definition control tables with a terminal name set to the value specified on this parameter. For a TXSeries for AIX server, the value is 1 to 16 characters, and is the DevType for a CICS terminal definition entry to be used as the model.

### Use upper case security

Select this check box to specify that any userid or password from an ECI application or resulting from a user prompt is converted to uppercase.

This setting is disabled by default.

### TCP/IP

Select this option to specify TCP/IP as the protocol for the server connection.

### SNA

Select this option to specify SNA as the protocol for the server connection.

### TCP62

Select this option to specify TCP62 as the protocol for the server connection.

## Using the configuration tool

### NetBIOS

Select this option to specify NetBIOS as the protocol for the server connection.

Protocol settings displayed on the panel change according to the protocol button you select.

### TCP/IP settings

To display these settings, select the **TCP/IP** option.

**Hostname or IP address:** Enter the character or numeric TCP/IP identifier for the host on which the CICS server is running. For example, cicsrv2.company.com (HostName) or 9.20.4.1 (IPAddress).

Hostnames are mapped to IP addresses either by the name server or in the hosts file in the etc subdirectory. It is, however, better to use a Hostname in case the IP address changes.

The \etc subdirectory is located as follows:

**OS/2** c:\mptn\etc

#### Windows NT

c:\winnt\system32\drivers\etc

#### Windows 95/98

c:\windows\etc

**Port:** Enter a numeric value in the range 0 through 65 535 defining the port number at the server to which the client should connect. The default value is 0.

A value of 0 indicates that the SERVICES file in the TCP/IP \etc subdirectory should be used to locate the port number for the service CICS using a protocol of TCP.

The \etc subdirectory is located as follows:

**OS/2** c:\mptn\etc

#### Windows NT

c:\winnt\system32\drivers\etc

#### Windows 95/98

c:\windows\etc

If no entry can be located in the SERVICES file, a value of 1435 is assumed, this being the TCP/IP architecture assigned port for CICS Universal Clients.

**Connection timeout:** Enter a value in the range 0 through 3600, specifying the maximum time in seconds that establishing a connection is allowed to take; the default value of 0 means that no limit is set by the client.

A timeout occurs if connection establishment takes longer than the specified time. The TCP/IP socket is closed and the return code passed back to the client application is either ECI\_ERR\_NO\_CICS or CICS\_EPI\_ERR\_FAILED.

**Send TCP/IP Keepalive packets:** Select this check box if you want TCP/IP to periodically send KeepAlive packets to the server to check the connection.

### SNA settings

To display these settings, select the **SNA** option.

**Use LU alias names:** Select this check box to use LU alias names.

Selecting this setting enables the **Partner LU name** and **Local LU name** to be specified as alias names instead of real LU names. This means, for example, that it is possible to switch between servers without stopping the CICS Universal Client. The default is that LU alias names are not used.

**Partner LU name:** Enter the LU Name of the server as it is known to the APPC configuration at the CICS Universal Client.

This can be a qualified 17-character name, for example, ABC3XYZ4.PQRS1234. Alternatively, you can enter an alias name, as long as **Use LU alias names** is selected.

**Local LU name:** Enter the name of a local LU to be used when connecting to the server. The same LU can be used for all server connections.

You can enter an alias name, as long as **Use LU alias names** is selected.

**Mode name:** Enter between 1 and 8 characters specifying the mode name to be used when connecting to the server. Omitting this setting results in a blank mode name.

(The eNetwork Communications Server API supports a mode name of all EBCDIC blanks.)

A value of \* can be specified if a blank mode name is required.

### TCP62 settings

To display these settings, select the **TCP62** option.

## Using the configuration tool

**Partner LU name:** Enter the LU Name of the server as it is known to the SNA configuration at the client. This must be a qualified 17-character name, for example, ABC3XYZ4.PQRS1234.

**Local LU name or template:** Enter a real LU name or a template for the name of a local LU to be used when connecting to the CICS server.

If you enter a template, you must fill in the **IP address mask for LU name template (optional)** field.

A template contains template replacement characters, that is, asterisks (\*), and is used to dynamically generate an LU name. For example, a template of CTS8BC\*\* indicates that a name must be dynamically generated to replace the two asterisks. An algorithm based on the template, IP address mask, and the workstation IP address, is used to create a unique Local LU name of, for example, CTS8BC38.

With dynamic name generation for the Local LU name, you can use the same configuration on all CICS Client machines because different Local LU names are generated on the basis of the IP address of the client machine. Therefore, you do not have to define multiple LUs within VTAM.

If several hundred clients are to use dynamic LU name generation to connect to the CICS server, you should have more template replacement characters in your Local LU name, for example, CTS8\*\*\*\*. In fact, CICS Transaction Server for OS/390 uses the last four characters of the LU name as the connection name, so these characters must be unique.

You can use the eNetwork Personal Communications **sxmap convert** utility to display the LU name that is generated from a specified internet IP address, IP address mask, and LU name template.

**IP address mask for LU name template (optional):** Enter an IP address mask in big-endian, that is, most significant byte first order. This must be a hexadecimal number, for example, FFFFFFF80.

This address mask is used in dynamic LU name generation. See the description of **Local LU name or template** for more information.

This parameter is ignored if **Local LU name** is not a template. The default is 00000000.

**Mode name:** Enter between 1 and 8 characters specifying the mode name to be used when connecting to the server. Omitting this setting results in a blank mode name.

(The eNetwork Communications Server API supports a mode name of all EBCDIC blanks.)

A value of \* can be specified if a blank mode name is required.

If ModeName is not specified, a default mode name of TCP62 is used.

You can use the **Maximum logical SNA sessions**, **Maximum SNA RU size**, and **SNA pacing size** settings to define the mode specified by **Mode name**.

If you specify a mode that is already specified in the eNetwork Personal Communications configuration file, the mode is not redefined. If the same mode is specified on two different TCP62 server connections, the mode definition is taken from the first server to be connected.

### Common TCP62 settings

To display these settings, select **Common TCP62 Settings**

**Fully qualified CP name or template:** Enter a fully-qualified CP name, or a template for the fully-qualified CP name, of the node to be started.

If you enter a template, you must fill in the **IP address mask for CP name (optional)** field.

A template contains template replacement characters, that is, asterisks (\*), and is used to dynamically generate a CP name. For example, a template of USIBMJKA.CP62\*\*\*\* indicates that a name must be dynamically generated to replace the four asterisks. An algorithm based on the template, IP address mask and the workstation IP address, is used to create a unique CP name of, for example, USIBMJKA.CP62AH38.

With dynamic name generation for the CP name, you can use the same configuration on all CICS Client machines because different CP names are generated on the basis of the IP address of the client machine. Therefore, you do not have to define multiple CPs to VTAM.

If several hundred clients are to use dynamic CP name generation, you should have more template replacement characters in your template, for example, USIBMJKA.CP\*\*\*\*\*.

The net ID part (the first part) of the template must not contain any template replacement characters.

## Using the configuration tool

**IP address mask for CP name (optional):** Enter an IP address mask in big-endian, that is, most significant byte first order. This must be a hexadecimal number, for example, FFFF8080.

This address mask is used in dynamic CP name generation. See the description of **Fully qualified CP name or template** for more information.

This parameter is ignored if **Fully qualified CP name or template** is not a template. The default is 00000000.

**Anynet domain name suffix:** Enter the AnyNet<sup>®</sup> domain name suffix. This parameter is used with the partner (**Partner LU name**) to determine the IP address of the host associated with that LU.

For example, if the domain name suffix is sna.ibm.com and the partner LU name is NETID.LUA a gethostbyname call is issued for hostname LUA.NETID.sna.ibm.com to obtain the IP address of the node that has the LU name NETID.LUA.

The default is sna.ibm.com.

**Remote node inactivity timeout:** Enter a number of seconds between 1 and 65535. The default value is 30 seconds.

This configuration setting controls the time that eNetwork Personal Communications waits before polling inactive connections. Refer to the eNetwork Personal Communications documentation for details on how this parameter behaves.

### Advanced TCP62 settings

To display these settings, select **Advanced TCP62 Settings**.

**Maximum logical SNA sessions:** Enter a value in the range 1 through 255. The default value is 8.

This parameter is used for TCP62 to define the number of logical SNA sessions.

**Maximum SNA RU size:** Enter a value in the range 256 through 4096. The default value is 1024.

This parameter is used for TCP62 in defining the mode specified by the **Mode name** setting.

**SNA pacing size:** Enter a value in the range 0 through 63. The default value is 8.

This parameter is used for TCP62 in defining the mode specified by the **Mode name** setting.

### NetBIOS settings

To display these settings, select the **NetBIOS** option.

**Applid:** Enter the NetBIOS name of the server. For CICS for OS/2 this is specified in the Local System Appl ID field in the system initialization table (SIT).

The name is case-sensitive.

**Adapter:** Enter a numeric value in the range 0 through 3 specifying which LAN adapter the client should use. The default is 0.

### Trace settings

To configure the trace settings, select the **Trace** option from the **Tools** menu.

#### Trace Settings

Select check boxes to specify the CICS Transaction Gateway and CICS Universal Client components that will be traced when tracing is turned on.

<b>Trace everything</b>	All components.
<b>Client API level 1</b>	The client API layer (level 1).
<b>Client API level 2</b>	The client API layer (level 1 and 2).
<b>CICSCLI command line</b>	The CICSCLI command interface.
<b>CICSTELD</b>	CICSTELD process.
<b>CICSTERM and CICSPRINT</b>	CICSTERM and CICSPRINT emulators.
<b>CPP classes</b>	The C++ class libraries.
<b>Client daemon</b>	CICS Universal Client daemon.
<b>Transport layer</b>	Interprocess communication.
<b>Protocol drivers</b>	Protocol drivers (TCP, SNA, and so on).
<b>REXX API</b>	The REXX API.
<b>Java Gateway</b>	The CICS Transaction Gateway. If you only have a CICS Universal Client, you will not see this setting.

You can also specify trace components by using the `/m` parameter on the CICSCLI command (except the Java Gateway component). Any component

## Using the configuration tool

tracing specified using CICSCLI overrides that specified with the configuration tool. If component tracing is not specified either by the CICSCLI command or the configuration tool, a default set of components is traced, namely: **Protocol drivers**, **Client daemon**, and **Client API level 1**. If you select any of the check boxes, this overrides the default set of components.

For the API component, you can specify the level of information to trace. The **Client API level 1** check box specifies that basic API-related information is traced, for example, before and after ECI, EPI, and ESI calls. The **Client API level 2** check box specifies that additional API trace entries are produced in addition to those of level 1.

Note that the CICSCLI /d=*nnn* command is used to set the maximum size of the data areas to be traced. The trace data may be truncated if you set *nnn* lower than the size of data expected.

### Client trace file

Enter the pathname of a trace file to which trace messages will be written, if tracing is enabled.

You do not have to enter an extension for the filename, as a file of type .BIN is always generated (or .WRP if the trace file wraps).

If no path is specified, the trace is written to the cicscli.bin file. This is in the Client \BIN subdirectory.

You can convert the binary trace file to an ASCII file using the CICSFTRC command.

For more information about tracing, refer to “CICS Universal Clients trace” on page 112.

### Maximum Client wrap size

Enter a value in the range 0 through 999 kilobytes. The default is 0.

This value specifies how large the trace file will grow on disk before wrapping. If the default value of 0 is specified, wrapping of trace is disabled.



## The configuration conversion tool

You use the configuration conversion tool (CTGCONV) to convert the configuration files of previous versions of IBM CICS Universal Clients to the new format of the CICS Universal Client Version 3.1 configuration file.

The conversion tool converts the following:

<b>CICSCLI.INI</b>	Client initialization file of CICS Clients Version 2 and Version 3.0.
--------------------	--

The conversion tool produces one output file called `ctg.ini` by default. Samples from this file are shown in “Editing the configuration file” on page 68.

The old files are renamed with the `.BAK` extension, but a banner is inserted into them stating that they are obsolete.

## Using the conversion tool

The parameters of CTGCONV are:

```
CTGCONV /c=file [/o=file]
```

for each parameter, *file* can be:

- a filename with extension, in which case the Client \BIN directory is assumed. The extension must be `.INI`.
- a full pathname.
- a directory name (without \ at the end), in which case the default filename is assumed.

**/c=*file***

Specifies, the client initialization file to be converted. This parameter is mandatory.

If *file* specifies a directory, a filename of `CICSCLI.INI` is assumed.

If the parameter is not specified, the `CICSCLI` environment variable is used to locate the client initialization file. If `CICSCLI` is not set, the file `Client \BIN\CICSCLI.INI` is assumed.

**/o=*file***

Specifies the pathname of the converted file. The default is `Client \BIN\ctg.ini`.

To get help on using CTGCONV, enter: `CTGCONV /?`

## Using the configuration tool

During conversion, the input files are renamed so that the last three characters of the file extension become .BAK. Any file with that name is overwritten, and any output file that exists before the conversion is renamed in the same way.

Redundant parameters are removed from the old configuration files, and other parameters are given new names in the converted file.

---

## Editing the configuration file

Although it is recommended that you use the configuration tool you can perform the configuration by editing the configuration file.

The configuration file is used for both CICS Universal Clients and CICS Transaction Gateway, and contains the following sections:

1. GATEWAY (CICS Transaction Gateway only)
2. CLIENT
3. SERVER
4. DRIVER

These sections have the format:

```
SECTION sectionname [=value]
    property1=value
    property2=value
    ...
    propertyN=value
ENDSECTION
```

You must restart the CICS Universal Client to pick up any changes to the configuration file.

The following sections give examples of the properties in each section. For information on the properties, refer to the descriptions of the corresponding settings in the configuration tool.

### CLIENT section

```
SECTION CLIENT = *
#-----
# Client section - This section defines the local CICS client. There
#                      should only be one Client section.

        MaxServers = 1           # Only allow one server connection
        MaxRequests = 90         # Limit the maximum server interaction
        MaxBufferSize = 32       # Allow for a 32K maximum COMMAREA
#        CCSID=850
```

```
#      CPName = ABCD1234.EFGH5678      # The TCP62 client's fully qualified CP name
#      DomainNameSuffix = cicstcp.ibm.com      # Domain name suffix for TCP62 server

ENDSECTION
```

### SERVER section

```
SECTION SERVER = cicssna1
#-----
# Server section - This section defines a server to which the client may
#                   connect. There may be several Server sections.

      Description = A SNA Server      # Arbitrary description for the server
      Protocol = SNA                  # Matches with a Driver section below
      NetName = ABCDEFGH              # The server's Alias Name
      LUAliasNames = Y
      LocalLUName = WXYZ9999          # The client's local LU name
      ModeName = LU62PS               # The SNA communications mode name

ENDSECTION
```

### DRIVER section

```
SECTION DRIVER = SNA                  ; TCP/IP, NETBIOS, SNA, TCP62
#-----
#      Driver section - This section defines a communications protocol DLL
#                   used to communicate with a server. There may be
#                   several Driver sections.

      DriverName = CCLIBMSN           #CCLIBMNB, CCLIBMSN, etc.

ENDSECTION
```

**Note:** The DRIVER section does not correspond to any settings in the configuration tool. The configuration tool selects the correct protocol drivers automatically.

---

## Customizing the keyboard

The keyboard mapping for emulator operation is defined in a keyboard mapping file. The file supplied is `cicskey.ini` in the Client \BIN directory by default. It is however, recommended that you create your own customized mapping file.

The keyboard mapping file can be identified by:

- The `/k` option of the `CICSTERM` command, which identifies a keyboard mapping file with a particular terminal (see page “colorfile parameter” on page 95).
- The `CICSKEY` environment variable. For example:

```
SET CICSKEY=C:\CUSTOM\MYKEYS.INI
```

## Customizing the keyboard

You can set this environment variable in the CONFIG.SYS file.

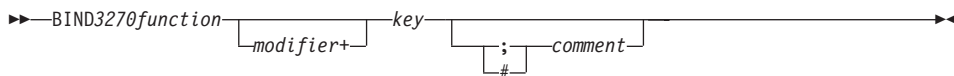
If neither of these is specified, a filename of cicskey.ini in the Client \BIN directory is assumed.

You can change the keyboard mapping file at any time, although changes do not take effect until the next time the terminal emulator is started.

## Keyboard mapping file syntax

This section describes the syntax of the keyboard mapping file. A statement must be provided for each key that is needed, because there are no default assignments (except for the alphabetic and numeric keys). There is no case sensitivity, so keywords and values may be entered in uppercase, lowercase or mixed case. Each binding must be on a separate line, and of the following form:

## Keyboard mapping file syntax



For example, to map the 3270 function EraseEof to the Ctrl+Delete keys pressed together the binding would be as follows:

bind	EraseEof	Ctrl+Delete ;erase to end of field
------	----------	------------------------------------

## The keyboard mapping file

In the mapping file, *3270function* can be any one of the following:

backspace	pa1	pf1	pf13
backtab	pa2	pf2	pf14
clear	pa3	pf3	pf15
cursordown		pf4	pf16
cursorleft	printscreen	pf5	pf17
cursorright	reset	pf6	pf18
cursorselect	tab	pf7	pf19
cursorup		pf8	pf20
delete	ignore	pf9	pf21
enter		pf10	pf22
eraseeof		pf11	pf23
eraseinput		pf12	pf24
home			
insert			
newline			

The value of `ignore` is provided to permit unwanted control keys on the keyboard to be ignored. (Unexpected glyphs are not generated.)

The *Modifier* can be any one of:

```
Alt      (IBM keyboards only)
Ctrl
Shift
```

The *Key* can be any one of the keys shown in Table 10, but some combinations of modifier+key are not supported:

*Table 10. CICS Universal Client keys that can be mapped*

Group	Keys
Escape key	Escape
Function keys	f1 f2 f3 f4 f5 f6 f7 f8 f9 f10 f11 f12
Numeric keys	0 1 2 3 4 5 6 7 8 9
Alphabetic keys	a b c d e f g h i j k l m n o p q r s t u v w x y z
Tab key	Tab
Movement keys	newline backspace insert home pageup delete end pagedown up left down right
Keypad keys	keypad/ keypad* keypad- keypad7 keypad8 keypad9 keypad4 keypad5 keypad6 keypad+ keypad1 keypad2 keypad3 keypad0 keypad. keypadenter

### Keys specific to particular keyboards

Some keys are specific to particular types of keyboard.

The following additional key is unique to IBM keyboards:

```
rightctrl
```

### Key combinations

The following combinations of modifier and key can be mapped:

**No modifier**                      All keys available for mapping.

## Customizing the keyboard

<b>Alt modifier</b>	Only function keys, numeric keys, movement keys, and alphabetic keys can be mapped.
<b>Ctrl modifier</b>	Only function keys, movement keys, alphabetic keys, tab key, and keypad keys can be mapped.
<b>Shift modifier</b>	Only function keys, numeric keys, tab key, and alphabetic keys can be mapped.

A sample key mapping file is shown in Figure 10 on page 73.

```

;*****
;* IBM CICS Client - 3270 Emulator Keyboard Binding File          *
;*****

; Format:
;   bind 3270_key [modifier+]workstation_key
;
; Refer to the "IBM CICS Clients Administration" book for full details
; of available values.

; Note: There are no default key bindings, ensure all required 3270
;       keys are mapped. "Enter" and "Clear" are particularly
;       important.

; Miscellaneous keys
bind Enter      RightCtrl
bind Enter      KeypadEnter
bind Enter      Ctrl+KeypadEnter
bind Clear      Escape
bind Reset      Alt+R
bind Newline    Newline
bind Tab        Tab
bind Backtab    Shift+Tab
bind Backspace  Backspace
bind Delete     Delete
bind Delete     Keypad.
bind Insert     Insert
bind Insert     Keypad0
bind EraseEof    Ctrl+Delete
bind EraseInput Alt+End
bind Ignore     End
bind PrintScreen Alt+P

; Cursor movement
bind CursorUp    Up
bind CursorUp    Keypad8
bind CursorDown  Down
bind CursorDown  Keypad2
bind CursorLeft  Left
bind CursorLeft  Keypad4
bind CursorRight Right
bind CursorRight Keypad6
bind CursorSelect Alt+F3
bind Home        Home
bind Home        Keypad7

```

Figure 10. Sample keyboard mapping file for CICS Universal Clients (Part 1 of 2)

## Customizing the screen colors

```
; PF and PA keys
bind Pf1      F1
bind Pf2      F2
bind Pf3      F3
bind Pf4      F4
bind Pf5      F5
bind Pf6      F6
bind Pf7      F7
bind Pf7      Keypad9
bind Pf7      PageUp
bind Pf8      F8
bind Pf8      Keypad3
bind Pf8      PageDown
bind Pf9      F9
bind Pf10     F10
bind Pf11     F11
bind Pf12     F12
bind Pf13     Shift+F1
bind Pf14     Shift+F2
bind Pf15     Shift+F3
bind Pf16     Shift+F4
bind Pf17     Shift+F5
bind Pf18     Shift+F6
bind Pf19     Shift+F7
bind Pf20     Shift+F8
bind Pf21     Shift+F9
bind Pf22     Shift+F10
bind Pf23     Shift+F11
bind Pf24     Shift+F12
bind Pa1      Alt+Insert
bind Pa2      Alt+Home
bind Pa3      Alt+PageUp
```

*Figure 10. Sample keyboard mapping file for CICS Universal Clients (Part 2 of 2)*

---

## Customizing the screen colors

The screen colors and attributes are defined in a color mapping file. A sample is provided for you to tailor. The file supplied is `cicscol.ini` in the `Client \BIN` directory by default. It is recommended that you create your own customized mapping file.

The color mapping file can be identified by:

- The `/c` option of the `CICSTERM` command, which identifies a color mapping file with a particular server (see page “colorfile parameter” on page 94).
- The `CICSCOL` environment variable: For example:

```
SET CICSCOL=C:\CUSTOM\MYCOLS.INI
```

You can set this environment variable in the `CONFIG.SYS` file



If neither of these is specified, a filename of `ciccol.ini` in the current directory is assumed.

A color mapping file is used to provide alternative representations in hardware environments where it is not possible to exactly replicate 3270 screen attributes, for example, blinking or underscore. The color mapping file therefore defines how 3270 screen attributes are emulated on the client hardware.

The color mapping file is optional. However, for most hardware environments a mapping file is required if blinking or underscore support is required by the emulator.

### Notes:

1. If the color mapping file specifies a mapping for an attribute, this mapping is used even if the hardware upon which the client is running actually supports the screen attribute.
2. If an application requests a 3270 field to be displayed with, for example, underscore, and no emulation setting has been specified, and the hardware cannot display underscore, then the field is displayed without any highlighting at all.

You can change the color mapping file at any time, although changes do not take effect until the next time the terminal emulator is started.

## Color mapping syntax

The syntax of the color mapping file is as follows. There is no case sensitivity—keywords and values may be entered in uppercase, lowercase or mixed case. Each binding must be on a separate line, and of the form:

### Color mapping file syntax

►—`BIND3270attribfg_color`—`/bg_color`—`;`—`comment`—`#`—►

## The color mapping file

In the color mapping file, `3270attrib` can be any one of the following:

## Customizing the screen colors

normal_protected	intensified_protected	
normal_unprotected	intensified_unprotected	
default	blinking_default	underscored_default
blue	blinking_blue	underscored_blue
green	blinking_green	underscored_green
cyan	blinking_cyan	underscored_cyan
red	blinking_red	underscored_red
magenta	blinking_magenta	underscored_magenta
white	blinking_white	underscored_white
yellow	blinking_yellow	underscored_yellow
default_highlight		
operator_information_area		

Each of *fg\_color* and *bg\_color* (foreground color and background color) can be any one of the following:

black	light_gray
blue	light_blue
brown	yellow
cyan	light_cyan
green	light_green
magenta	light_magenta
red	light_red
gray	white

If *bg\_color* is omitted, a default value of black is taken.

The sample color mapping file is shown in Figure 11 on page 77.

```

;*****
;* IBM CICS Client - 3270 Emulator Color Binding File *
;*****

; Format:
;   bind 3270_field foreground_color/background_color
;
; Refer to the "IBM CICS Clients Administration" book for full details
; of available values.

; Operation information area at the bottom of the screen.
bind operator_information_area  black/green

; Color used as default for a terminal defined as monochrome, or to display
; characters displayed in the default color when character or field
; attributes are used
bind default                    light_green/black

; Color used for an intense field when either the terminal is defined as
; monochrome, or the screen has been formatted with extended attributes but
; the data to be displayed has default color and highlight.
bind default_highlight         white/black

; Colors used for datastream formatted with field attributes when no
; extended character or field attributes are in use
bind normal_unprotected        light_green/black
bind intensified_unprotected    light_red/black
bind normal_protected          light_cyan/black
bind intensified_protected      white/black

; Colors used for datastream formatted with extended field attributes
; or datastream formatted with character attributes.
bind red                       light_red/black
bind green                     light_green/black
bind blue                      light_blue/black
bind magenta                   light_magenta/black
bind cyan                     light_cyan/black
bind yellow                    yellow/black
bind white                     white/black

; The following bindings provide a mapping for 3270 blink and under-
; score attributes, as most workstation displays do not provide blink
; or underscore capabilities. For those that do these bindings could
; be deleted.
bind blinking_default          light_green/gray
bind blinking_red              light_red/gray
bind blinking_green            light_green/gray
bind blinking_blue             light_blue/gray
bind blinking_magenta          light_magenta/gray
bind blinking_cyan             light_cyan/gray
bind blinking_yellow           yellow/gray
bind blinking_white            white/gray

```

Figure 11. Sample color mapping file (Part 1 of 2)

## Customizing the screen colors

```
bind underscored_default      light_green/light_gray
bind underscored_red          light_red/light_gray
bind underscored_green        light_green/light_gray
bind underscored_blue         light_blue/light_gray
bind underscored_magenta      light_magenta/light_gray
bind underscored_cyan         light_cyan/light_gray
bind underscored_yellow       yellow/light_gray
bind underscored_white        white/light_gray

;* End of file *
```

*Figure 11. Sample color mapping file (Part 2 of 2)*

---

## Testing the client/server link

After you have prepared the configuration file, and customized the keyboard and screen color attributes, you can test the client/server link.

1. Start the CICS server in the normal manner, as described in the documentation for the server.
2. On the client workstation, enter the command `CICSCLI /s=servername`, (or click on the equivalent icon), where `servername` is the name of the server from the configuration file.
3. After a few moments, enter the command `CICSCLI /1`, or click on the equivalent icon. This displays information on the status of the servers to which the client is connected or attempting connection. Verify that the server is available.

If the connection to the server cannot be made, examine any messages and the client error log to determine the cause of the error (see “Chapter 7. Problem determination” on page 105).

---

## Chapter 6. Operation

**This chapter describes**

- “An introduction to the CICS Universal Clients commands”
- “CICS Universal Clients icons” on page 80
- “The CICSCLI command” on page 80
- “The CICSTERM command” on page 92
- “The CICSPRNT command” on page 97
- “CICS Client telnet terminal support” on page 102

This chapter describes the commands for controlling CICS Universal Clients, their terminal emulation, and printer support. There is a detailed reference section for each command.

---

### An introduction to the CICS Universal Clients commands

You can use the following commands to provide control of client operation:

- The CICSCLI command

This command is used to start and stop the client process, check the availability of servers, and set other options.

- The CICSTERM command

This command starts a terminal emulation session with particular options.

- The CICSPRNT command

This command starts a printer terminal session with particular options.

These commands are the basis of icons supplied with CICS Universal Clients.

The CICS Universal Clients start automatically when any of their functions (EPI, ECI, or 3270 terminal emulation) are invoked at the workstation. It is not necessary to first use the CICSCLI command to start the client; you can simply run CICSTERM and CICSPRNT or double-click on their icons and the client is started with the necessary server connections.

You must explicitly terminate any server connections initiated by ECI calls. To do this, use the CICSCLI /x=*servername* or CICSCLI /i=*servername* command, see “CICSCLI command reference” on page 88.

### CICS Universal Client for OS/2 operation

The CICS Universal Client for OS/2 folder contains icons for controlling client operation (see “CICS Universal Clients icons”). These icons are based on the CICSCLI, CICSTERM, and CICSPRNT commands, which you enter at the OS/2 command prompt.

---

### CICS Universal Clients icons

All CICS Universal Clients are supplied with the following icons:

#### **Start Client**

Starts the client according to the definitions in the configuration file.

This is equivalent to the command `CICSCLI /s`.

#### **Stop Client**

Stops the client.

This is equivalent to the command `CICSCLI /x`.

#### **CICS Terminal**

Starts a 3270 terminal emulation session according to definitions in the configuration file.

This is equivalent to the command `CICSTERM`.

#### **CICS Printer**

Starts a CICS print terminal session according to definitions in the configuration file.

This is equivalent to the command `CICSPRNT`.

#### **Client Status**

Lists connected servers.

This is equivalent to the command `CICSCLI /l`.

You can tailor the properties of these icons, and add new icons according to the command variations you require for your client).

---

### The CICSCLI command

You use the `CICSCLI` command to:

- Start the client control process, and start communication with CICS servers (/s parameter)
- Stop the client control process (/i and /x parameters)
- Restart the client (/j and /y parameters)

- Specify the configuration file to use (/f parameter)
- Turn client trace on (/d parameter)
- Turn client trace off (/o parameter)
- Specify the client components to be traced (/m parameter)
- Set up security (/c, /u, and /p parameters)
- List connected CICS servers (/l parameters)
- Enable the display of pop-up messages (/e parameter)
- Disable the display of pop-up messages (/n parameter)

The following sections provide examples of using the CICSCLI command. Full details of the command syntax are given in “CICSCLI command reference” on page 88.

### Starting the client control process

To start the client control process, enter:

```
CICSCLI /s
```

To start the client control process and start communication with a CICS server, enter:

```
CICSCLI /s=servername
```

where *servername* is the name of a CICS server.

**Note:** You can start the CICS Universal Client by double-clicking on the **Start Client** icon.

### Starting connections with additional servers

You can start connections to servers when the CICS Universal Client is already running. When the CICS Universal Client has been started using the /s option, or when connections to additional servers are to be started, enter:

```
CICSCLI /s=servername
```

where *servername* is the name of a CICS server.

### Stopping the client control process

This section describes the different ways in which you can stop the client control process with the CICSCLI command.

## The CICSCLI command

### Stopping in a controlled manner

To stop the client control process for all connected servers after all outstanding units of work have completed, enter:

```
CICSCLI /x
```

To terminate the session with a particular server, enter:

```
CICSCLI /x=servername
```

where *servername* is the name of a CICS server. This only stops the session with the named server, it does not stop the client control process.

**Note:** You can stop the CICS Client by double-clicking on the **Stop Client** icon.

### Stopping immediately

To stop the client control process for all connected servers without completing outstanding units of work, enter:

```
CICSCLI /i
```

To terminate the session with a particular server, enter:

```
CICSCLI /i=servername
```

where *servername* is the name of a CICS server. This only stops the session with the named server, it does not stop the client control process.

## Restarting the client

This section describes the different ways in which you can restart the client control process with the CICSCLI command.

### Restarting in a controlled manner

To stop the client control process for all connected servers after all outstanding units of work have completed, and then start it again, enter:

```
CICSCLI /y
```

CICSCLI /y is equivalent to CICSCLI /x followed by CICSCLI /s. Server connections are not re-established when the client is restarted.

### Restarting immediately

To stop the client control process for all connected servers without completing outstanding units of work, and then start it again, enter:

```
CICSCLI /j
```



CICSCLI /j is equivalent to CICSCLI /i followed by CICSCLI /s. Server connections are not re-established when the client is restarted.

### Specifying which configuration file to use

To use a configuration file other than the default ctg.ini, enter:

```
CICSCLI /s /f=filename
```

where *filename* is the name of the file. The /f parameter can be used only with the /s parameter, and only if the client is not already started.

### Turning on client trace

To turn on the trace for a client, enter, for example:

```
CICSCLI /d=nnn
```

where *nnn* is the maximum size of the data areas to be traced.

If you need to trace the client from the startup sequence, you can specify the /s and /d parameters together.

The trace is written to the cicscli.bin file in the Client \BIN directory, which can then be formatted with the CICSFTRC utility.

### Specifying the trace components

To specify which client components to trace, enter, for example:

```
CICSCLI /m=TRN,API.2
```

which would cause the transport layer and full API trace information to be produced. For more information on which components you can trace, see “CICSCLI command reference” on page 88.

### Turning off client trace

To turn off the trace for a client, enter:

```
CICSCLI /o
```

### Setting up security

CICS servers may require that a userid and password be provided by the CICS client before they permit a client connection, terminals to be installed, or ECI or EPI activity to be performed. This is dependent on server and protocol security settings.

## The CICSCLI command

The CICS client maintains a default userid and password per server connection, and these can be set by any of the following methods:

- CICSCLI security commands
- The ESI function **CICS\_SetDefaultSecurity**
- Client security pop-ups

The default values are used for any request to a CICS server where a userid and password are required, and where no values have been supplied by an application.

The CICS client also maintains a userid and password per EPI terminal, and these can be set by one of the following methods:

- In the **CICS\_EpiAttributes\_t** structure on a **CICS\_EpiAddExTerminal** call.
- The EPI function **CICS\_EpiSetSecurity**. Note that this function can also be used for terminals added using the **CICS\_EpiAddTerminal** call.

Default and terminal userids and passwords do not appear in client traces.

The **Use upper case security** setting in the configuration file determines whether userids and passwords are converted to uppercase. If this setting is enabled, you can enter userids and passwords in either uppercase or lowercase. This setting is disabled by default.

**Note:** Userids and passwords must not contain DBCS characters.

### CICSCLI security commands

You can use the following commands after first starting the client process with the CICSCLI /s commands.

To identify a server to which security information is to be associated, enter:

```
CICSCLI /c=servername
```

where *servername* is the name of a CICS server.

To set a userid to be used when accessing this server, enter:

```
CICSCLI /c=servername /u=userid
```

where *userid* is the userid.

To set a password to be used when accessing this server, enter:

```
CICSCLI /c=servername /p=password
```

where *password* is the password.

You can enter the /u and /p parameters together.

### Using the ESI to set security details

You can use the ESI function **CICS\_SetDefaultSecurity** to specify a default userid and password to be used for CICS client requests to a CICS server.

The functions **CICS\_VerifyPassword** and **CICS\_ChangePassword** enable an application to check security values used in ECI and EPI calls, if the CICS server and protocol support password expiry management.

For more information, refer to *CICS Family: Client/Server Programming*.

### Security popups

The CICS client prompts for a userid and password via a security popup on:

- a CICS client connection to a CICS server,
- a terminal emulator install,
- or an EPI terminal install.

if the default userid and password have not been set for that server, and no values have been supplied by an application.

**Note:** Security popups are not displayed directly by ECI or EPI transaction requests, although they may be prompted by these requests, if the request causes a server connection to be established, or a terminal to be installed.

If the server and protocol support password expiry management, the client prompts for a new password if required, and indicates the specific problem with any incorrect values entered. If password expiry management is not supported, the client prompts if the entered values are incorrect, but without providing any details of the specific problem.

**Note:** CICSCLI /n suppresses prompting for userid and password, and returns security errors to the application.

### ECI security

The ECI application may provide a userid and password via the ECI parameter block. If these are not provided in the parameter block when the ECI application runs, the default values are used if they have been set. Otherwise, the call fails with a security return code.

## The CICSCLI command

Note that you are not prompted if the userid and password supplied on the ECI parameter block are invalid, and the values passed in the ECI parameter block are not saved for future use.

If the userid and password used to communicate with the server are incorrect, the ECI call fails with a security return code.

### EPI terminal security

A userid and password may be required for the install of EPI terminals and for transactions started against those terminals. This depends on the server and protocol security settings and on whether the terminal is defined as signon capable or signon incapable. (For more information, refer to the appropriate section in the *CICS Family: Client/Server Programming* book.)

If a terminal userid and password are not supplied, the default userid and password set for the CICS server are used, or the user is prompted via a security popup on a terminal install, otherwise, a security error is returned to the application.

**Note:** The userid and password entered on the signon screen of a signon capable terminal are treated by the CICS client simply as 3270 data, therefore the values appear in traces.

### Terminal emulator (CICSTERM) security

A terminal userid and password can not be specified for a CICSTERM emulator. By default, CICSTERM requests a signon capable terminal. If a userid and password are required for the terminal install, the server default values are used, if set, or the user is prompted via a security popup.

### Password expiry management

For CICS clients the management of expired passwords can be handled by the ESI functions **CICS\_ChangePassword** and **CICS\_VerifyPassword**. Also, security popups will prompt specifically for any unknown userid, incorrect password, or expired password, after the user has initially entered a userid and password via a security popup.

The ESI functions can only be used with CICS servers that support the PEM (password expiry management) service transaction:

- Transaction Server for OS/390 Release 1 or later.
- CICS for MVS/ESA Version 4.1

Password management using PEM is supported for CICS Universal Client for OS/2, CICS Universal Client for Windows 98, and CICS Universal Client for Windows NT communicating over TCP62 and SNA.

### Listing the connected servers

To list all the servers connected to a client and their status, enter:

```
CICSCLI /l
```

A list like the following is displayed:

```
CCL8001I CICSCLI - CICS Client Control Program
CCL0002I (C) Copyright IBM Corporation 1994,1998. All rights reserved.
CCL8041I The CICS client is using the following servers:
CCL8042I Server 'CICS0S2' (using 'TCP/IP' to 'CICS0S2') is available
CCL8043I Server 'CICSNT' (using 'NETBIOS' to 'CICSNT') is unavailable
```

**Note:** You can list the connected servers by double-clicking on the **Client Status** icon.

### Disabling the display of messages

To disable the display of all messages output with the command, enter, for example:

```
CICSCLI /s /q
```

### Enabling and disabling the display of pop-up messages

There may be times when you do not want messages (client error and security) to appear in pop-up windows. (These pop-up messages include those not associated with the client control process.) For example, if you leave CICS Universal Clients running unattended (overnight perhaps), you would not want messages appearing when the pop-up window cannot be closed.

To disable the display of pop-up messages, enter:

```
CICSCLI /n
```

To enable the display of pop-up messages again, enter:

```
CICSCLI /e
```

You can specify the /n parameter together with the /s parameter.

The display of pop-up messages is enabled by default.

### Displaying the command parameters

To display the parameters of the CICSCLI command, enter:

```
CICSCLI /?
```

## The CICSLI command

### CICSLI and applications

You can call CICSLI from within applications, as long as this is supported by the programming language. You can therefore enter CICSLI at the command line, then run an application to call CICSLI with additional parameters.

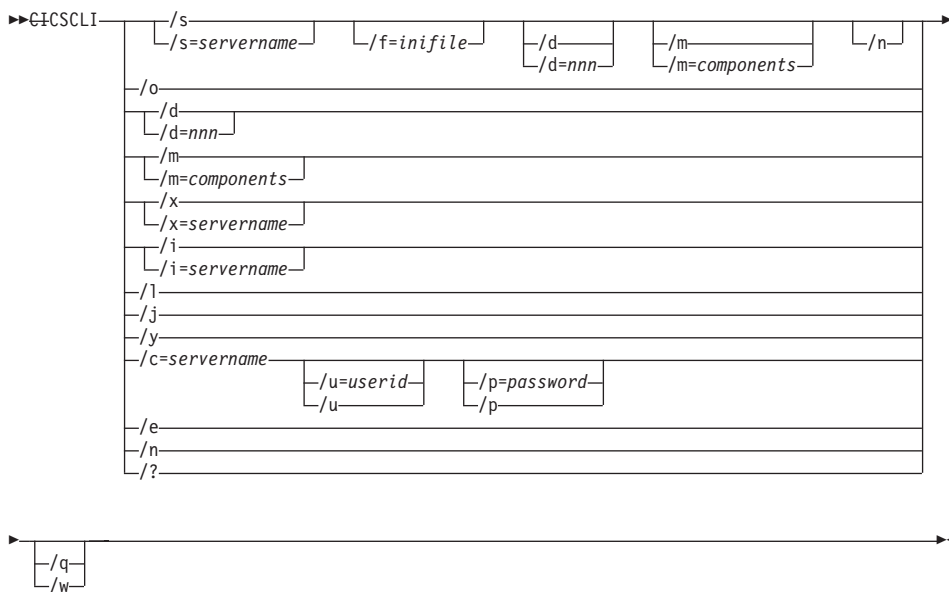
### CICSLI command reference

To conform with other popular syntax conventions, this slash character may be replaced with a minus sign (-). The commands and options are not case-sensitive.

All options of the form */x=variable* may contain spaces in the variable part, if it is enclosed in double quotes. Double quotes within variables must be entered as `\` , that is with a backslash preceding the double quote.

For an explanation of syntax diagrams, see “Conventions and terminology used in this book” on page xiv.

#### CICSLI Command Syntax



The options are:

***/c=servername***

Identifies the name of the server to which security information in the form of a userid and password is to be associated. Some CICS servers

require that the user provide security information to the server before interacting with the server. The CICS Universal Client prompts the user at the workstation for a userid and password, unless these have already been provided via CICSCLI (see the descriptions of the /u and /p options).

### /d=[nnn]

Turns the debug trace on for the client. If tracing is required while the client is starting up, this option may be specified along with the /s option.

*nnn* is the maximum size of data areas to be traced in bytes. The range is 1 through 32 767 bytes, and the default value is 512 bytes.

Output from the trace option is directed to the cicscli.bin file in the Client \BIN directory, which can then be formatted with the CICSFTRC utility. The resultant trace file, cicscli.trc, is an ASCII file that can be read with a text editor. For more information see “Formatting the binary trace file” on page 114.

/e Enables the display of client error and security messages in pop-up windows.

### /f=inifile

Specifies the name of the configuration file. The default is ctg.ini. If the file is not in the Client \BIN directory, you must specify the full path name.

**Note:** If the file resides on a redirected network drive then problems may occur if the network drive is later detached.

/i Stops the client immediately. The options /i=*servername* and /i operate as for /x=*servername* and /x respectively *but* the client does not wait for outstanding units of work to complete. Stopping the client in this way may result in a loss of data at connected servers.

/j Stops the client immediately and then restarts the client.

A client restart involves shutting down the client, waiting for the client to shut down, and then starting up again. CICSCLI /j is equivalent to CICSCLI /i followed by CICSCLI /s. Server connections are not re-established when the client is restarted.

Using /y is the preferred way of restarting the client.

/l Causes a list of all connected servers to be displayed. For each server, the netname of the server as it is known to the client is also displayed, as well as the state of the connection to the server and the connection protocol. One line per server is displayed.

## The CICSCLI command

**/n** Disables the display of client error and security messages in pop-up windows.

Any messages that would have been logged are still logged.

**/m=[components]**

Specifies a comma-separated list of identifiers for components that will be traced when tracing is turned on. You can specify any of the following components:

**ALL** All components

**API.1** The client API layer (level 1)

**API.2** The client API layer (level 1 and 2)

**API** Synonymous with API.1

**CCL** CICS client daemon

**CCP** The C++ class libraries

**CLI** The CICSCLI command interface

**DEF** The default components, that is the API, CCL, and DRV components

**DRV** Protocol drivers (TCP, SNA, and so on)

**EMU** CICSTERM and CICSPRNT emulators

**RXX** REXX API

**TLD** CICSTELD and telnet daemon.

**TRN** Interprocess communication

The **/m** parameter does not turn on tracing in the client; it simply specifies the components to trace. You cannot use **/m** while tracing is turned off, so you must specify the **/d** parameter before you use **/m**.

If you specify **/m** with no parameters, a list of the possible component identifiers is displayed, with an 'x' beside each component that it is currently enabled for tracing.

You can also specify settings for trace components using the configuration tool (see "Using the configuration tool" on page 52). Any component tracing specified using CICSCLI overrides that specified using the configuration tool. If component tracing is not specified either by the CICSCLI command or by using the configuration tool, a default set of components is traced, namely: DRV, CCL, and API. Any component tracing specified using the configuration tool overrides the default set of components.



For the API component, you can specify the level of information to trace. Components API or API.1 specify that basic API-related information is traced, for example, before and after ECI and EPI calls. Component API.2 specifies that additional API trace entries are produced in addition to those of level 1.

Note that the CICSLI `/d=nnn` command is used to set the maximum size of the data areas to be traced. The trace data may be truncated if you set *nnn* lower than the size of data expected.

**/o** Turns off trace if it is already active.

**/p=password**

Sets the current password to be used when accessing the server specified by the `/c` parameter. (Or by the `/s` parameter.) This password is used if the server requires a password (and userid) before running transactions on the client's behalf.

For ECI applications, any userid and password specified in the ECI parameter block overrides values set via the CICSLI command.

Specifying `/p` or `/p=` (that is, no password is specified) resets the associated password to a null value.

**/q** Disables the display of all messages output with the CICSLI command.

**/s** Starts the client control process. No attempt is made to initiate communication with a server unless `/s=servername` is specified. In this case, the control process also connects to the server using information specified in the configuration file. The *servername* must exist in the configuration file.

**/u=userid**

Sets the current userid to be used when accessing the server specified by the `/c` parameter. (Or by the `/s` parameter.) This userid is used if the server requires a userid (and password) before running transactions on the client's behalf.

For ECI applications, any userid and password specified in the ECI parameter block overrides values set via the CICSLI command.

Specifying `/u` or `/u=` (that is, no userid is specified) resets the associated userid to a null value.

**/x** Stops the client in a controlled manner. If `/x=servername` is specified, then when all outstanding units of work on the specified server have completed, the connection to the server is terminated. If other server connections are active, these remain unchanged.

## The CICSLI command

If /x is specified without a servername, the client waits for all outstanding units of work to complete, terminates all connections to servers, and ends the control process. Using /x or /x=servername is the preferred way of stopping the client.

|        /y        Restarts the client in a controlled manner.

|        A client restart involves shutting down the client, waiting for the  
|        client to shut down, and then starting up again. CICSLI /y is  
|        equivalent to CICSLI /x followed by CICSLI /s. Server connections  
|        are not re-established when the client is restarted.

|        Using /y is the preferred way of restarting the client.

|        /?        Causes the command syntax to be displayed.

---

## The CICSTERM command

The CICSTERM command is used to control 3270 terminal emulation. You can start emulator sessions, specify terminal emulator characteristics, and the names of the keyboard mapping and color mapping files.

You can have multiple terminal emulation sessions running simultaneously.

The CICSTERM command detects whether the hardware on which the client is running is enabled for double-byte character set (DBCS) display. If it is, the emulator can display DBCS characters.

|        For CICSTERM emulators, the maximum screen size is 27 rows by 132  
|        columns. This is because CICS Universal Clients support the ASCII-7 subset of  
|        the 3270 data stream architecture, which only supports 12 bit addressing. For  
|        more information, see *CICS Family: Client/Server Programming*.

|        **Note:** Some uses of servers and protocols require a model terminal definition  
|        for the emulator that explicitly specifies that the client wants to display  
|        DBCS.

### Using CICSTERM

You use the CICSTERM command to:

- Start a 3270 terminal emulator (/s or /r parameter)
- Specify the initial transaction (/t parameter)
- Specify the name of the keyboard mapping file (/k parameter)
- Specify the name of the color mapping file (/c parameter)
- Define the 3270 terminal emulator characteristics (/n and /m parameters)
- Specify whether a terminal emulator is signon capable (/a parameter)
- Determine the print file processing (/p parameter)
- Specify a file to which print files are appended (/f parameter)

You issue the CICSTERM command *once* with all the parameters you require.

The following is an example of a CICSTERM command:

```
CICSTERM /s=CICSOS2 /t=CESN /k=mykeys.ini /c=mycols.ini  
/n=cicsv123 /f=clprint.txt /q
```

In this example:

**/s=CICSOS2**

Specifies that a 3270 terminal emulator is started for the server CICSOS2.

**/t=CESN**

Specifies that the initial transaction is CESN.

**/k=mykeys.ini**

Specifies that the keyboard mapping file is named as mykeys.ini.

**/c=mycols.ini**

Specifies that the color mapping file is named as mycols.ini.

**/n=cicsv123**

Specifies that the 3270 terminal emulator characteristics are defined by the terminal definition cicsv123.

**/f=clprint.txt**

Specifies that the print file will be appended to the file clprint.txt.

**/q** Specifies that the display of messages output by the command is disabled.

All parameters of CICSTERM are optional. That is, you can enter the CICSTERM command without any parameters, and defaults are taken from the configuration file. (This is equivalent to double-clicking on the CICS Terminal icon.) Full details of the parameters are given in “CICSTERM command reference” on page 94.

### Stopping a terminal emulator

To stop a terminal emulator, enter the string specified by the Terminal exit configuration setting. This is EXIT by default.

### CICSTERM and user exits

CICSTERM can be used to drive EPI user exits.

The EPI user exits and how CICSTERM can use them are described in the *CICS Family: Client/Server Programming* book.

## The CICSTERM command

### CICSTERM and RETURN TRANSID IMMEDIATE

When an application running from a CICSTERM session issues either of the commands:

```
EXEC CICS RETURN TRANSID(name) IMMEDIATE
EXEC CICS RETURN TRANSID(name) IMMEDIATE INPUTMESSAGE(data-area)
```

then the transaction named in the TRANSID option starts straight away without any user input. When the INPUTMESSAGE option is specified, the contents of the *data-area* are passed to the new transaction, and the screen is not updated with the *data-area* contents.

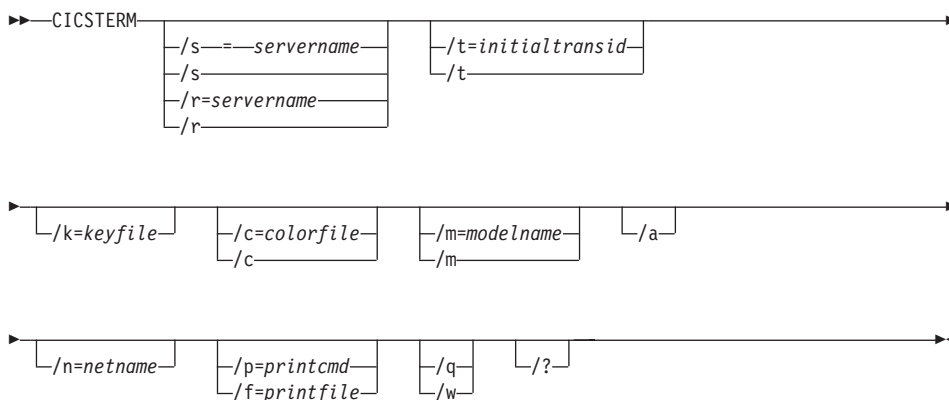
This behavior with the RETURN TRANSID(*name*) IMMEDIATE command was not supported for CICSTERM before CICS Universal Clients Version 3.1, and is not supported for CICSPRNT. (It is however, supported by the EPI).

**Note:** Issuing these EXEC CICS commands from CICSTERM does not result in the StartTranExit or ReplyExit user exits being driven. The EPI user exits and how CICSTERM can use them are described in the *CICS Family: Client/Server Programming* book.

### CICSTERM command reference

For an explanation of syntax diagrams, see “Conventions and terminology used in this book” on page xiv.

#### CICSTERM Command Syntax



The options are:

**/a** Specifies that the terminal emulator is *not* signon capable. By default, CICSTERM creates terminal emulators that are signon capable.

For information on signon capability, refer to the EPI chapter in the *CICS Family: Client/Server Programming* book.

### */c=colorfile*

Identifies the name of a color mapping file (see “Customizing the screen colors” on page 74) to be used with the emulator. If this parameter is omitted, the environment variable CICSCOL is assumed to identify the color mapping file. If CICSCOL is not defined, a filename of cicscol.ini in the Client \BIN directory is assumed.

If the parameter is specified as /c=, (that is, the color mapping filename is omitted), the emulator runs without any color definitions.

### */f=printfile*

Specifies the name of a file to which the output of print requests is appended. If the name of the file contains embedded blanks, it must be surrounded by double quotes (“”). Any double quotes within the name of the file must be entered as backslash double quote (\”).

If neither of the /f or /p parameters is provided, the **Print command** or **Print file** configuration settings define the command, file, or default action to take with print requests.

### */k=keyfile*

Identifies the name of a keyboard mapping file (see “Keyboard mapping file syntax” on page 70) to be used with the emulator. If this parameter is omitted, the environment variable CICSKEY is assumed to identify the key mapping file. If CICSKEY is not defined, a filename of cicskey.ini in the Client \BIN directory is assumed.

### */m=modelname*

Specifies the name of a model terminal definition, as known at the server to which the emulator is to connect, to be used to define the terminal characteristics. If neither this parameter nor /n=netname is specified, any Model terminal definition value from the configuration file is used. If no Model terminal definition value has been specified in the configuration file, the server’s default terminal definition is assumed.

If the parameter is specified as /m= (that is, the modelname is omitted), any Model terminal definition value specified in the configuration file is ignored, and the server’s default terminal definition is assumed.

This option is case-sensitive.

### */n=netname*

Specifies the name of a particular terminal definition at the server that this emulator is to be installed as. The precise interpretation of

## The CICSTERM command

netname varies between servers, for example, on CICS for OS/2 it references a termid defined in the CICS tables, on TXSeries for AIX it is a netname.

This option is case-sensitive.

### **/p=printcmd**

Specifies an operating system command used to process the temporary print file generated when print requests are received by the terminal emulator. If the command contains embedded blanks, then the command must be surrounded by double quotes (""). Any double quotes within the command must be entered as backslash double quote (\").

If neither of the /f or /p parameters is specified, the **Print command** or **Print file** setting in the configuration file defines the command, file, or default action to take with print requests.

The temporary print file is post-processed by appending the filename to the command, and executing the resultant command. Thus print output may simply be copied to a local printer, copied into a permanent file, processed further for inclusion into a document, and so on. If the temporary file is to be processed by a print command, the command is responsible for deleting the temporary file.

**/q** Disables the display of all messages output by the command.

### **/s=servername or /r=servername**

Specifies the name of the server that the terminal emulator is to be connected to. This servername must correspond to an entry in the configuration file. You can specify /s, or /r, but not both.

If neither parameter is specified, the first server entry in the configuration file is used.

If the parameter is specified as /s or /r (that is, no servername is provided) then, if the configuration file identifies more than one potential server to which the client can connect, the user is prompted to select from a list of available servers. These prompts are generated even if the /q parameter is specified.

If there is only one potential server identified in the configuration file, that server is used and the user is not prompted.

### **/t=initialtransid**

Identifies the initial transaction to be invoked for this terminal. If this option is omitted, any initial transaction specified in the configuration file is run. The string may be up to 128 characters long, specifying both a transaction name, and parameters to be passed to the

transaction. The transaction name is the first four characters or the characters up to the first blank in the string. The rest of the string is the parameter data.

If the parameter is specified as /t= (that is, the initialtransid is omitted), any initial transaction specified in the configuration file is ignored.

This option is case-sensitive.

**Note:** Be careful that transactions that you specify either here or in the configuration file do not require terminal input to complete.

- /w Prompts the user, before the command completes, to press the Enter key, to confirm that messages output to the screen (both informational and error) have been read.
- /? Causes the parameter syntax to be listed; any other options specified are ignored.

---

## The CICSPRNT command

The CICSPRNT command is used to control 3270 printer terminal emulation.

Applications running on the server can direct output to a printer in one of two ways:

1. An application running from a terminal can initiate printing by sending a map or data with the PRINT indicator set
2. The user can start at the client a 3270 Print Terminal Emulator using the CICSPRNT command. A 3270 Print Terminal Emulator must be started for a netname or model terminal definition (or termid in the case of CICS for OS/2) predefined in the server's terminal tables. Output is directed to such a device by starting a transaction against the printer device.

For CICSPRNT emulators, the maximum screen size is 27 rows by 132 columns. This is because CICS Universal Clients support the ASCII-7 subset of the 3270 data stream architecture, which only supports 12 bit addressing. For more information, see *CICS Family: Client/Server Programming*.

**Note:** At client workstations you can use the PrintScreen key (as defined by the keyboard mapping file). Note however, that any lines that contain only null characters are not printed. For a 'blank' line to be printed, it must contain at least one space character.

## The CICSPRNT command

### Using CICSPRNT

You use the CICSPRNT command to:

- Start a 3270 print terminal emulator (/s or /r parameter)
- Specify the initial transaction (/t parameter)
- Define the 3270 printer terminal emulator characteristics (/n and /m parameters)
- Determine the print file processing (/p parameter)
- Specify a file to which print files are appended (/f parameter)

You issue the CICSPRNT command once with all the parameters you require.

The following is an example of a CICSPRNT command:

```
CICSPRNT /s=CICSOS2 /n=P123 /t=XPRT /f=clprint.txt /q
```

In this example:

**/s=CICSOS2**

Specifies that a 3270 print terminal emulator is started for the server CICSOS2.

**/n=P123**

Specifies that the 3270 print terminal emulator characteristics are defined by the terminal definition v123 (in the terminal control table for CICS for OS/2 in this case.)

**/t=XPRT**

Specifies that the initial transaction is XPRT.

**/f=clprint.txt**

Specifies that the print file to which print requests are appended is clprint.txt.

**/q** Specifies that the display of messages output by the command is disabled.

All parameters of CICSPRNT are optional, except that you must specify either **/n=netname** or **/m=modelname**. That is, you can enter the CICSPRNT command with just the **/n** or the **/m** parameter, or both, and defaults for other parameters are taken from the configuration file. Full details of the parameters are given in “CICSPRNT command reference” on page 99.

If the system upon which the client is executing supports DBCS, it is assumed that the printer attached to the processor also supports DBCS. Conversely, if the system does not support DBCS, the client will not send DBCS data to the printer.



## CICS Printer icon

For CICS Universal Client for OS/2, when you double-click on the **CICS Printer** icon, the Specify Parameters dialog box is displayed. This allows you to enter the required parameters for CICSPRNT.

For CICS Universal Clients, you should edit the settings for the icon to change the parameters according to your usage.

The CICSPRNT process runs as a minimized window, and the window can be enlarged to view the current status of the printer. You can use an action available from a pulldown menu to terminate the print function.

## CICSPRNT and user exits

CICSPRNT can be used to drive EPI user exits.

The EPI user exits and how CICSPRNT can use them are described in the *CICS Family: Client/Server Programming* book.

## CICSPRNT and RETURN TRANSID IMMEDIATE

Unlike CICSTERM, CICSPRNT does not support the commands:

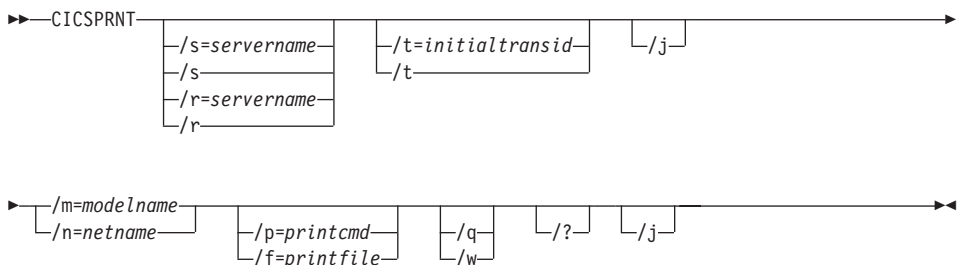
```
EXEC CICS RETURN TRANSID(name) IMMEDIATE
EXEC CICS RETURN TRANSID(name) IMMEDIATE INPUTMESSAGE(data-area)
```

For more information, refer to “CICSTERM and RETURN TRANSID IMMEDIATE” on page 94.

## CICSPRNT command reference

For an explanation of syntax diagrams, see “Conventions and terminology used in this book” on page xiv.

### CICSPRNT Command Syntax



## The CICSPRNT command

The options are:

### */f=printfile*

Specifies the name of a file to which the output of print requests is appended. If the name of the file contains embedded blanks, it must be surrounded by double quotes ("). Any double quotes within the name of the file must be entered as backslash double quote (\").

If neither of the /f or /p parameters is provided, the **Print command** or **Print file** setting in the configuration file defines the command, file, or default action to take with print requests.

### */j*

Specifies that CICSPRNT should concatenate all EXEC CICS SEND PRINT commands issued on a server transaction into a single print job. This print job is issued when the transaction terminates. Otherwise CICSPRNT will generate a separate print job for every EXEC CICS SEND PRINT command issued for a server transaction.

### */m=modelname*

Specifies the name of a model terminal definition, as known at the server to which the 3270 Print Terminal emulator is to connect, to be used to define the terminal characteristics. If this parameter is not specified, any Model terminal definition value from the configuration file is used. If no Model terminal definition value has been specified in the configuration file, the server's default terminal definition is assumed.

You must specify either the /m or the /n option, or both.

This option is case-sensitive

### */n=netname*

Specifies the name of a particular terminal definition at the server that this 3270 Print Terminal emulator is to be installed as. The precise interpretation of netname varies between servers. For example, on CICS for OS/2 it references a termid defined in the terminal control table (TCT), on TXSeries for AIX it is a netname.

You must specify either the /m or the /n option, or both.

This option is case-sensitive.

### */p=printcmd*

Specifies a command used to process the temporary print file generated when print requests are received by the terminal emulator. If the command contains embedded blanks, then the command must be surrounded by double quotes ("). Any double quotes within the command must be entered as backslash double quote (\").

If neither of the `/f` or `/p` parameters is specified, the **Print command** or **Print file** setting in the configuration file defines the command, file, or default action to take with print requests.

The temporary print file is post-processed by appending the filename to the command, and executing the resultant command. Thus print output may simply be copied to a local printer, copied into a permanent file, processed further for inclusion into a document, and so on. If the temporary file is to be processed by a print command, the command is responsible for deleting the temporary file.

**/q** Disables the display of all messages output by the command.

**/s=servername** or **/r=servername**

Specifies the name of the server that the printer is to be connected to. This servername must correspond to an entry in the configuration file. You can specify `/s`, or `/r`, but not both.

If neither parameter is specified, the first server entry in the configuration file is used.

If the parameter is specified as `/s` or `/r` (that is, no servername is provided) then, if the configuration file identifies more than one potential server to which the client can connect, the user is prompted to select from a list of available servers. These prompts are generated even if the `/q` parameter is specified.

If there is only one potential server identified in the configuration file that server is used and the user is not prompted.

**/t=initialtransid**

Identifies the initial transaction to be invoked for this printer. If this option is omitted, any initial transaction specified in the configuration file is run. The string may be up to 128 characters long, specifying both a transaction name, and parameters to be passed to the transaction. The transaction name is the first four characters or the characters up to the first blank in the string. The rest of the string is the parameter data.

If the parameter is specified as `/t=` (that is, the initialtransid is omitted), any initial transaction specified in the configuration file is ignored.

**Note:** Be careful that transactions that you specify either here or in the configuration file do not require terminal input to complete.

This option is case-sensitive.

## The CICSPRNT command

- /w** Prompts the user, before the command completes, to press the Enter key, to confirm that messages output to the screen (both informational and error) have been read.
- /?** Causes the parameter syntax to be listed; any other options specified are ignored.

---

## CICS Client telnet terminal support

Support is provided for multiple tn3270 terminals to connect to CICS servers via a single CICS Universal Client. This function is available for all CICS servers that support the EPI.

You must first start a CICS telnet daemon by using the following command:

```
CICSTELD [/p=port] [/s=server] [/c=client codepage] [/l=server codepage]  
          [/t=tranid] [/m=model]
```

where:

**/p=port**

Specifies the numeric Transmission Control Protocol (TCP) port number that CICSTELD listens on for incoming telnet client requests. If the port number is not valid, CICSTELD issues an explanation message. If you do not specify a port number, CICSTELD assumes the default CICS port number of 1436.

**/s=server**

Specifies the applid of the CICS server.

**/c=client codepage**

Specifies the codepage of the tn3270 client.

The default codepage is 037 (EBCDIC US English).

**/l=server codepage**

Specifies the codepage that is used by the CICS server's EPI data stream.

The default codepage is 437 (ASCII US English).

**/t=tranid**

Specifies an initial transaction.

**/m=model**

Specifies a model terminal name on the CICS server.

**/?** Causes the parameter syntax to be listed; any other options specified are ignored.

When you have started the CICS telnet daemon, tn3270 terminals can then connect to the CICS server host.

You can use any Telnet 3270 client to access CICS through the CICS telnet daemon. To connect a tn3270 terminal, you can use tn3270 (or an equivalent program, if tn3270 is not installed by default on your system). For example:

```
tn3270 hostname -p [port] -ext
```

where *hostname* is the hostname of the machine running the telnet Daemon, and *port* is the port number of the telnet daemon, if you are not using the default CICS port number. The -ext parameter enables extended datastream support. If extended datastream is not enabled you will get a blank screen. For more information, refer to the tn3270 documentation.

To stop the daemon, press the q key.

You may experience problems if the number of Telnet 3270 clients connected to CICSTELD exceeds the MAXTASKS setting of the CICS server. To prevent this situation, use the CICSTELD parameter /n to limit the number of connections to CICSTELD. For example, CICSTELD /n=10 will allow no more than 10 Telnet 3270 connections to CICSTELD. There is no default limit on the number of CICSTELD users.

CICSTELD works only with models that use the standard terminal size of 24x80. If you use a model (specified with the /m parameter), you must ensure that it is of the correct terminal size.

## Telnet terminal support

---

## Chapter 7. Problem determination

### **This chapter discusses problem determination**

- “Preliminary checks and information”
- “Program support” on page 106
- “CICS Universal Client problem determination” on page 112
- “CICS server problem determination” on page 120
- “Communication problem determination” on page 120
- “Troubleshooting” on page 123

This chapter is an introduction to the process of problem determination in CICS Universal Client environments.

Problem determination involves starting with a symptom (or a set of symptoms) and tracing it back to its cause, to allow resolution of the problem. This chapter therefore discusses facilities for collecting information that will help you and your service organization solve the problem. The chapter concentrates on problem determination of CICS Universal Clients themselves. It also gives pointers to tools that are used in problem determination of communication products.

---

### **Preliminary checks and information**

Before you begin the problem determination process, consider the following questions to see whether there is an obvious cause of the problem.

- Are there any messages explaining the failure (on the CICS Universal Client workstation, CICS server, or both)?
- Has the connection run successfully before?
- Are specific parts of the network affected by the problem?
- Can you reproduce the error?

As you go through the questions, take note of anything relevant to the problem. Even if your observations do not at first suggest a cause, they may be useful in a detailed problem determination exercise.

---

### Program support

If you need to contact your support organization, you should first refer to the Service and Support card supplied with the product for details of the support available to you, or visit our Web site at:

<http://www.ibm.com/software/ts/cics/>

and follow the **Support** link.

This section provides guidance on reporting problems, detailing the information you should collect to assist your support organization. It also explains how the problem is handled through to resolution and the provision of a fix.

### Reporting problems

Before reporting your problem to the support organization try to ensure that the error is actually occurring within your CICS Universal Clients system, even if you are unsure whether the problem is due to CICS Universal Clients itself.

In practice, many errors reported to support organizations turn out to be user errors, or they cannot be reproduced, indicating just how difficult it can be to determine the precise cause of a problem. User errors are mainly caused by faults in application programs or by errors in setting up systems.

The ability of the support organization to resolve your problem quickly is dependent on the quality of the information you provide to them in relation to the problem. For this reason it is good practice to collect and organize problem data before making the initial contact.

Summarize your problem and the documentation/information you collect on a problem reporting sheet, ideally using the suggested structure and headings shown in Table 11 on page 109. This sheet provides a valuable summary to the support organization, and a copy will be useful for your own records.

Although you are unlikely to have all of the data listed in the sheet when you first contact the support organization, you should be able to describe the symptoms of your problem and supply details of your system configuration (see “Documentation of problems” on page 108 for details).

### Level-1 support

If this is your first contact regarding this problem a unique incident number will be assigned. A Problem Management Record (PMR) is opened on the



RETAIN<sup>®</sup> database system, where all activity associated with your problem is recorded. The PMR remains “open” until it is solved.

Make a note of the incident number on your copy of the problem reporting sheet. You will be expected to quote the incident number in all future correspondence relating to this problem.

Your initial communication with the support organization is through a Level-1 representative, who uses the keywords that you have provided to search the RETAIN database for problems with similar symptoms. If your problem is found to be one already known to the support organization, and a fix has been devised for it, you will be advised of the availability of corrective service software that you can install to overcome the problem, as described in “Obtaining the fix” on page 112. If the RETAIN search is unsuccessful, the problem is passed on to a Level-2 representative who will contact you for more information about your problem, to start a more detailed investigation of the cause.

### Level-2 support

Let the Level-2 representative know if any of the following events occurred before the problem appeared:

1. Changes in level of CICS Universal Clients, compiler or associated licensed programs
2. Corrective service software and fixes applied to workstation software
3. Problem Tracking Fixes (PTFs) applied to other associated products
4. Additional features used
5. Application programs changed
6. Unusual operator action.

At this stage you may be asked to supply more details, such as your problem reporting sheet, or some of the documentation listed in it (see “Documentation of problems” on page 108 for more details). If this happens, the PMR is updated to show any documentation you have submitted.

Based on the information you supply, the investigation then carried out can determine whether the cause of your problem is new to the support organization, or is already known.

If the problem is a valid new one, an APAR may be raised, to be dealt with by the CICS service team, as described in “APARs and fixes” on page 111. If, however, the problem is already known, and a fix has been developed, you can obtain it as described in “Obtaining the fix” on page 112.

### Documentation of problems

In a communication environment, where many clients may be connected to several servers, you must obtain information from both the client and the server sides.

To facilitate problem determination, try to reduce your environment to only one client workstation and one server to narrow down the range of possibilities that may cause the error.

Listed below are the sources of information likely to be needed by the support organization when tackling problem determination for CICS Universal Clients. The list is also summarized on the example problem reporting sheet (see Table 11 on page 109), where it can be used as a checklist of the information you have available.

- A description of the problem and the circumstances under which it occurs, that is, what you were trying to do.
- An operating system dump.
- A dump file for the CICS server
- A trace file for both CICS Universal Client (binary) and the server
- A trace for any communication product you were using
- Listings of relevant application programs
- A sample short program that demonstrates the problem
- CICS Universal Clients and Server message logs wherever possible. These contain information that is often overlooked.
- A list of APARs applied to the system
- Installation and customization parameters in effect at time of failure (for example, the configuration file or the .NDF file of Communications Server).
- Details of any user environment variables and hardware configurations.
- Operating Systems, compiler and other application release levels.
- A list of all the information you are providing, and the form in which it is being provided.

Table 11. Sample problem reporting sheet

<b>PROBLEM REPORTING SHEET</b>		
Date	Severity	Problem Number
Company name		Contact name
Address		
Telephone		
<b>System configuration—CICS Client</b>		
CICS Client Rel	COBOL compiler level	Operating System
Client service level	C compiler level	Communications prod.
Fixes applied?	PL/I compiler level	
<b>System configuration—CICS Server</b>		
CICS Server Rel	COBOL compiler level	Operating System
Server service level	C compiler level	Communications Prod.
Host CICS Rel	PL/I compiler level	Database Server
Fixes applied?	JDK/JRE level	
<b>Problem/Enquiry</b>		
Abend/Trap	Incorrout	
Wait	Message	
Loop	Other	
Performance		
<b>Documentation available—CICS Client</b>		
ctg.ini	Program output	Operating system dump
Message Logs	Application source	CICS Client dump
CICS Client Trace	Translator output	Testcases
Symptom string	Compiler output	User Env. variables
Comms product trace		
<b>Documentation available—CICS Server</b>		
Abend code	Program output	Operating system dump
Message Logs	Application source	CICS Server dump
CICS Server Trace	Translator output	Testcases
Symptom string	Compiler output	User Env. variables
Comms product trace		
<b>Actions</b>		
Date	Name	Activity
<b>Resolution</b>		
APAR	Fix	Other

## Locating and compiling information

The advice listed below will help you compile the information required. If you are in any doubt about how to gather any of the items listed above, wait until you can seek advice from your support organization.

## Program Support

- Try to establish which program in your system seems to be the cause of the problem. As you are reading this book, it is likely that you already believe CICS Universal Clients to be the problem source.

You also need to provide the version and release number, for example Ver 3 Rel 1 Mod 0, together with the service level, for example Version 3.0.0, Current service level: UNnnnnn (where *nnnnn* is a valid PTF number provided with the service fix). With CICS Universal Clients, you can find the current version and service level by selecting **Details** in the Installation and Maintenance Utility. You then select **Product status** and the **Service level** button.

- Obtain OS/2 operating system details by using the SYSLEVEL command at a command prompt. It is often useful to redirect this information to a file and print it as part of your problem documentation. For example, to output to a file SYSTEM.LEV, enter `SYSLEVEL > SYSTEM.LEV`
- You should find details of your compiler level on the product media labels or the associated documentation. Alternatively, check the panel displayed on the screen at compile time.
- Give the problem a severity level. Severity levels have the following meanings:
  - Severity level 1 indicates that you are unable to use a program, resulting in a critical condition that needs immediate attention.
  - Severity level 2 indicates that you are able to use the program, but that operation is severely restricted.
  - Severity level 3 indicates that you are able to use the program, with limited functions, but the problem is not critical to your overall operation.
  - Severity level 4 indicates that you are able to use the program with the problem causing negligible hindrance.

When deciding the severity of the problem, take care to categorize it correctly, as the procedures followed by the support organization depend on the severity level supplied.

- Also, try to classify the error and give a brief description of the problem. Include keywords associated with the problem, such as ABEND, WAIT, LOOP, PERFORMANCE, INCORROUT, and MESSAGE, corresponding to the problem classification types used in the RETAIN database. Strings containing other keywords are also useful. These are not predefined, and might include such items as a message or message number, an abend code, any parameters known to be associated with the problem, or, for example, STARTUP, INITIALIZATION, or TRANSIENT DATA.
- Finally, you should include your address, relevant contact names, and details of the other products at your installation.

Keep a copy of the problem reporting sheet which summarizes all the information relevant to the problem, together with any available documentation such as dumps, traces and output from programs, translators, and compilers.

### Submitting the documentation

If you are asked to provide documentation, it will help the support organization during their investigation if you adhere to the following rules when preparing it:

- Provide as much information as possible in softcopy format.
- Write any notes and documents in English.
- If you are asked to supply any additional trace output, provide an unformatted binary version, rather than the viewable output from the formatter.
- If you upload files to a mainframe system, do so in binary rather than ASCII format.

### APARs and fixes

Once a reported problem is confirmed as being both new and valid, An “authorized program analysis report” (APAR) is raised from the Problem Management Record (PMR) and becomes a permanent record describing your error, and in time, its solution. An APAR is the means of reporting to the appropriate product service team any problems you find with an IBM program.

#### The APAR process

The first step in the APAR process is that a Level-2 representative from the support organization enters an APAR containing a description of your problem into the RETAIN system. If you have a means of getting round the problem, details of this are entered as well. Your name is also entered, so that your support organization knows who to contact if the service team need to ask anything further about the APAR documentation.

At this stage, you are given an APAR number. This number is always associated with the APAR and its resolution and, if a code change is required, is associated with the fix as well.

The service team may ask for additional backup documentation, normally via the Level-2 representative. The specific documentation required will vary from problem to problem, and depends on what information has already been supplied during the PMR stage.

## Program Support

During the investigation, you can ask your support organization at any time about how your APAR is progressing, particularly if it is a problem of high severity.

### Obtaining the fix

When the service team has found a fix for your problem, you may be asked to test it on your system. If so, you may be sent a Program Temporary Fix (PTF) in the form of individual replacement modules. This is normally done through the Level-2 organization and feedback from your testing will be requested.

Each PTF may contain several APAR fixes. If an individual APAR fix within a PTF is found to be in error, it may still be advisable to apply the PTF, to obtain the other APAR fixes.

In time, formal corrective service software is made available, which you can order through your support organization. Traditionally, corrective service software has been supplied on Corrective Service Diskettes (CSD) ordered by CSD number. It may now be supplied on CD-ROM or via the Internet.

Fixes for CICS Clients are available from the Personal Software Services Web page at: <http://ps.software.ibm.com/>.

Personal Software Services provides technical information and assistance for the Personal Systems Support Family (PSSF) of products, which includes CICS.

---

## CICS Universal Client problem determination

In a CICS Universal Client environment there are two sources of information for problem determination:

1. CICS Universal Client trace
2. CICS Universal Client error log, see “CICS Universal Client error processing” on page 119

### CICS Universal Clients trace

Client trace is a very useful problem determination tool for communication problems. You can use the trace functions to collect detailed information on the execution of a certain function or transaction. A trace can show you how the execution of a particular activity is affected by, for example, the execution of other tasks in a CICS system. Each trace entry has a time stamp attached to it, which provides information on the time taken to perform certain activities.

You can specify which components of CICS Universal Client you want to trace. You control this with the CICSCLI /m command, (see “CICSCLI command reference” on page 88), or by specifying a list of components using the configuration tool.

The output from the trace function is a binary trace file called, by default, cicscli.bin in the Client \BIN directory. Using the configuration tool, you can specify the filename for this file, but not the extension, which is always .BIN. Using the **Maximum Client wrap size** configuration setting, you can specify that the binary trace file should wrap into a second trace file, and you can also specify the maximum size of these files.

To read the trace, you must run the CICSFTRC utility to convert the binary file(s) into a text file, cicscli.trc by default. The following summarizes the trace files (default names are shown):

<b>cicscli.bin</b>	The binary trace file produced by running the client trace.
<b>cicscli.wrp</b>	The second binary trace file if wrapping of client trace is enabled.
<b>cicscli.trc</b>	The name of the text trace file produced when the binary trace file is converted to a text file using the CICSFTRC utility.
<b>cicscli.bak</b>	The backup file of the binary trace file. A backup file is produced from any existing .BIN file when you turn tracing on.

See “Formatting the binary trace file” on page 114 for information on the trace conversion utility.

### Starting and stopping client trace

To start client trace, enter the CICSCLI command with the /d option, for example:

```
CICSCLI /d=nnn
```

where *nnn* is optional, and is the maximum size in bytes of the data areas to be traced. (The default value is 512.)

**Note:** It is recommended that you set at least /d=1000 to ensure that all relevant data is included in the trace before sending it to your support organization.

If you need to trace the client immediately from startup, you can specify the /s and /d parameters together in the same CICSCLI command. For example,

## CICS Universal Clients trace

the following command starts the connection to a CICS server with the name CICSTCP, enables the trace function, and sets the maximum data area size to be traced to 128 bytes:

```
CICSCLI /s=CICSTCP /d=128
```

To stop client trace, enter the CICSCLI command with the /o option (or click on the equivalent icon), for example:

```
CICSCLI /o
```

The trace is also automatically stopped if you stop the client by using the CICSCLI /x command.

### Wrapping of client trace

You can control the size of the binary trace file by specifying that it wraps into a second trace file. You turn on wrapping of trace using the **Maximum Client wrap size** configuration setting, where you specify the maximum size of the wrapping trace (in kilobytes). If this value is 0 (the default), wrapping trace is not turned on.

When wrapping trace is turned on, two files (called cicscli.bin and cicscli.wrp) are used. Each file can be up to half the size of the **Maximum Client wrap size** value.

### Formatting the binary trace file

You use the Binary Trace Formatter utility CICSFTRC to convert the binary trace file cicscli.trc to ASCII text. The utility has the following parameters:

**/m=*list of components***

Specifies that only trace points from the listed components are written to the text file. The components you can specify are the same as for CICSCLI /m. If /m is not specified, all trace points in the binary trace are written to the text file.

**/w[=*filename*]**

Indicates that there are two binary trace files to format and then concatenate (that is, the binary files were created with a wrapping trace). If no filename is specified with the /w parameter, CICSFTRC assumes that the name of the second trace file is cicscli.wrp.

**/n** Indents entry and exit points in the test trace file to make it more readable. By default, indentation is turned off.

**/p** Specifies that the TIDs and PIDs are to be shown in the trace. By default, this parameter is turned off.



**/i=filename**

Specifies the name of the input (binary) trace file, which is cicscli.bin by default.

**/o=filename**

Specifies the name of the output (text) trace file. If no /o parameter is specified, the name of the text trace file is assumed to be cicscli.trc.

**/d=data size**

Specifies the maximum amount of data to write to disk for each dump. If this option is not specified, the maximum data size is 512 bytes.

If wrapping of trace is turned on, *data size* can be a maximum of half of the wrapping size.

**CICS Universal Clients trace analysis**

The CICS Universal Clients trace file records detailed information on all actions taken during the execution of a particular activity. You can use this information in your problem determination activities, and to help understand how the CICS Universal Clients perform a particular function, for example, establishing a connection to a CICS server.

If you cannot interpret the trace yourself, you should contact your support organization and forward the unformatted binary trace file.

Some sample traces are shown in “Sample CICS Universal Clients trace” on page 116.

**Format of trace entries**

The entries in the clients trace file have the following format:

```
date time [number] component trace message
data
```

where:

**date**

The date the entry was written.

**time**

The time the entry was written, to millisecond accuracy.

**[number]**

A number to aid your support organization in the diagnosis of serious problems.

**[component]**

The component of the product to which this entry applies.

## CICS Universal Clients trace

### *trace message*

The trace message number and text. These trace messages are explained in *CICS Universal Clients Messages*.

### *data*

Some trace entries include a dump of key data blocks in addition to the trace message.

**Sample CICS Universal Clients trace:** Figure 12 on page 117 shows the trace information recorded during the successful connection of a CICS Universal Client to a CICS server using the TCP/IP protocol. The trace was generated using the commands:

```
CICSCLI -s=cicstcp -d  
CICSCLI -o
```

The trace was generated for CICS Universal Client for AIX but is applicable for other CICS Universal Clients

```

11:01:47.129 [1007] TRC: CCL1042 *** CICS Client for AIX v3.1 Service Level 00 - service trace started
11:01:47.130 [2183] CCL: CCL2048 Maximum trace data size set to 512
11:01:47.133 [2114] CCL:CCL2142 GetNextTimeout timeout is 0 seconds
11:01:47.135 [2030] CCL:CCL2106 Comms Event : LINK-UP
11:01:47.135 [2019] CCL:CCL2055 Connection with server established (LinkId=1)
11:01:47.135 [2035] CCL:CCL2109 Send server TCS data
11:01:47.136 [3214] CCL: CCL3251 Comms Allocate request (LinkId=1, Tran='CCIN')
11:01:47.136 [3217] CCL: CCL3238 Comms Allocate completed (LinkId=1, ConvId=1, Rc=0)
11:01:47.136 [2127] CCL:CCL2143 CommsBegin - OK
11:01:47.138 [3100] CCL: CCL3113 CCIN install request: ApplId='*', Codepage=819
11:01:47.138 [3225] CCL:CCL3242 Comms Send request: Length=38 (ConvId=1)
11:01:47.139 [3229] CCL:CCL3241 Comms Send completed (ConvId=1, Rc=0)
11:01:47.139 [2131] CCL:CCL2143 CommsSendData - OK
11:01:47.139 [3230] CCL:CCL3254 Comms Wait request (ConvId=1)
11:01:47.140 [4410] DRV:CCL4411 TCP/IP (to CICSTCP) send data: Length=90
50ef0 0000005A00000001 000B90A0280502FF ...Z.....[.....].....
50f00 0003D1004004C3C3 C9D5180B02404040 ...@.....@@@ ..J. .CCIN...
50f10 404040404040400B 01BF000000000000 @@@@.....
50f20 00000083002612FF 0000000C01010000 .....&..... ..C.....
50f30 0000000300000008 03F8F1F900000008 .....819....
50f40 0401000000000006 0C80 .....
11:01:47.140 [3234] CCL:CCL3243 Comms Wait completed (ConvId=1, Rc=0)
11:01:47.140 [2114] CCL:CCL2142 GetNextTimeout timeout is 0 seconds
11:01:47.163 [2029] CCL:CCL2107 Comms Private Event : COMMS
11:01:47.163 [4411] DRV:CCL4412 TCP/IP (to CICSTCP) receive data: Length=12
40220 0000003D00000001 00039001 ...=.....
11:01:47.164 [4409] DRV:CCL4412 TCP/IP (to CICSTCP) receive data: Length=49
50efc 003112FF0000000C 0102000000000003 .1.....
50f0c 0000000D017CF0E9 F8C1C1C1C1000000 .....|..... ..@Z8AAAA...
50f1c 0B03F8F8F5F960F1 0000000909010100 ...../..... ..8859-1.....
50f2c 00 .....
11:01:47.164 [2114] CCL:CCL2142 GetNextTimeout timeout is 0 seconds
11:01:47.165 [2033] CCL:CCL2106 Comms Event : CONV-DATA
11:01:47.166 [2022] CCL:CCL2058 Incoming conversation data (ConvId=1)
11:01:47.166 [3235] CCL:CCL3244 Comms Receive request (ConvId=1)
11:01:47.166 [3239] CCL:CCL3256 Comms Receive completed (last): Length=49 (ConvId=1, Reason=0, Rc=0)
50efc 003112FF0000000C 0102000000000003 .1.....
50f0c 0000000D017CF0E9 F8C1C1C1C1000000 .....|..... ..@Z8AAAA...
50f1c 0B03F8F8F5F960F1 0000000909010100 ...../..... ..8859-1.....
50f2c 00 .....
11:01:47.167 [3102] CCL: CCL3114 CCIN install response: ApplId='@Z8AAAA', Codepage=8859-1, Rc=0
11:01:47.167 [3241] CCL:CCL3255 Comms Complete request (ConvId=1)
11:01:47.168 [3244] CCL:CCL3246 Comms Complete completed (ConvId=1, Rc=0)
11:01:47.168 [3218] CCL:CCL3252 Comms Deallocate request (ConvId=1)
11:01:47.168 [3221] CCL:CCL3239 Comms Deallocate completed (ConvId=1, Reason=0, Rc=0)
11:01:47.169 [2042] CCL:CCL2114 Processed TCS Reply - Server connected
11:01:47.169 [2114] CCL:CCL2142 GetNextTimeout timeout is 0 seconds
11:02:51.974 [2114] CCL:CCL2142 GetNextTimeout timeout is 0 seconds
11:02:51.975 [1004] TRC: CCL1043 *** Service trace ended ***

```

Figure 12. Sample CICS Universal Clients trace

Message ID	Explanation
------------	-------------

# CICS Universal Clients trace

CCL1042	Start of trace message. The trace file is overwritten each time a trace is started. You can delete the file when required. Check the date and time stamp to ensure that you are reading the correct trace.
CCL2048	Maximum trace data size is at the default size of 512 bytes. You can modify this size by specifying the size value in the start command for the client trace (see “Starting and stopping client trace” on page 113).
CCL3251	The client sends a CCIN transaction to the server to install its connection definition on the server.
CCL3238	Reply to message CCL3238, includes the conversation ID for this conversation.
CCL3113	The client sends a CCIN transaction to the server with Appl ID set to * to install its application. The Appl ID is specified in the configuration file as Client=*. This requests the server to dynamically generate an Appl ID that is unique within the CICS server system.
CCL3114	This is the response to message CCL3114 with the dynamically generated Appl ID.
CCL1043	End of trace message.

Figure 13 shows trace information recorded when we tried to connect to a CICS server over TCP/IP using an invalid port number. The port number specified in the ctg.ini file was not defined in the services file of the server. Hence, the connection could not be established.

```
11:04:48.116 [1007] TRC:CCL1042 *** CICS Client for AIX v3.1 Service Level 00 - service trace started ***
11:04:48.117 [2183] CCL:CCL2048 Maximum trace data size set to 512
11:04:48.120 [2114] CCL:CCL2142 GetNextTimeout timeout is 0 seconds
11:04:55.054 [2114] CCL:CCL2142 GetNextTimeout timeout is -1 seconds
11:04:55.054 [3207] CCL:CCL3249 Comms Open request (Server=CICSTCP, Driver=CCLIBMIP)
11:04:55.055 [4408] DRV:CCL4413 TCP/IP (to CICSTCP) address=9.20.19.134, port=1089, socket=3
11:04:55.056 [3210] CCL:CCL3236 Comms Open completed (Server=CICSTCP, LinkId=1, Rc=0)
11:04:55.056 [2114] CCL:CCL2142 GetNextTimeout timeout is 3660 seconds
11:04:55.058 [1004] TRC:CCL1043 *** Service trace ended ***
```

Figure 13. CICS Universal Clients trace: using an invalid port number

Message ID	Explanation
CCL4413	Shows the port number used for this connection request.

You must check your definitions in the SIT on the server, the configuration file on the workstation, and the services file for the port number specified.

You must provide a valid port number or use the default value.

### CICS Universal Client error processing

There are two categories of messages in IBM CICS Universal Clients:

1. Messages displayed to the user
2. Errors written to the CICS Universal Clients error log and trace file.

The *CICS Universal Clients Messages* book explains all of these messages.

#### Error log messages

Any errors on the client workstation that are not caused by incorrect use of the API are written to the client error log.

The error log (cicscli.log) is an ASCII text file that you can browse using a standard text editor.

The error messages are explained in *CICS Universal Clients Messages* and also the help text for all messages is provided in two ASCII text files in the Client \BIN subdirectory. You can view these using any standard text editor:

<b>CCLMSG.HLP</b>	Help for the end user messages, in the language you chose at installation time.
-------------------	---

<b>CCLLOG.HLP</b>	Help for the trace and log messages. This is provided in English only.
-------------------	--

Errors resulting from incorrect use of the APIs simply result in error return codes to the application. It is the responsibility of the application to notify the end user of the error and provide guidance on corrective actions.

#### Pop-up messages

Errors generated from within the client process are displayed in a pop-up window. You must click on **OK** in the pop-up window before the client can continue processing. There may be times when you do not want messages (client error and security) to appear in pop-up windows; for example, if you leave a CICS Universal Client running unattended overnight. To disable the display of pop-up messages, enter the following command:

```
CICSCLI /n
```

When the display of pop-up messages is disabled, the messages are still written to the client error log. To re-enable the display of pop-up messages, enter the following command.

```
CICSCLI /e
```

## CICS Universal Client error processing

You can specify the /n parameter together with the /s parameter. The display of pop-up messages is enabled by default.

---

## CICS server problem determination

The most important facilities for problem determination on CICS servers are:

- Traces
  - auxiliary
  - internal
- Dumps
- CICS message logs
- Statistics information
- Monitoring information
- Execution diagnostic facility (EDF)
- CICS-supplied transactions, CEBR and CECI
- Independent software vendor (ISV) tools

Information about these facilities is given in the *Problem Determination* books for the individual products (see “CICS publications” on page xvi). You may need to contact the server system administrator for more information, for example, about a CICS server error log.

The following shows the error message prefixes for CICS products:

<b>CCL</b>	CICS Universal Clients and CICS Transaction Gateway
<b>FAA</b>	CICS for OS/2
<b>DFH</b>	CICS on System/390
<b>ERZ</b>	Transaction Server for Windows NT Version 4.0, and TXSeries
<b>AEG</b>	CICS for OS/400.

---

## Communication problem determination

Even a small telecommunications network is a very complex system in which all components depend on one another. If one component fails and presents incorrect information to the other components, the latter may fail even more severely than the former. Sometimes the failure may be considerably delayed, and the error indicator may be lost before the error is detected. Thus, it is sometimes very difficult to analyze a problem in the communication part of a system.

The CICS Universal Clients generate various messages associated with the use of the supported communication protocols and the associated products. Refer to *CICS Universal Clients Messages* for a list of these messages and their explanations.

The communication products themselves generate error messages. For details of these, and information on troubleshooting, you should refer to the documentation for the communication product. The following sections summarize useful commands and utilities for problem determination.

### NetBIOS-providing products

For information on error messages related to NetBIOS, refer to your operating system documentation.

### TCP/IP-providing products

#### IBM TCP/IP for OS/2

TCP/IP for OS/2 provides utilities to help in problem determination when communication problems occur. For a detailed discussion of these utilities, and problem determination in a TCP/IP environment, refer to the *TCP/IP for MVS, VM, OS/2 and DOS Troubleshooting Guide*.

The utilities are:

- |                 |   |
|-----------------|---|
| <b>PING</b>     | Sends an Internet Control Message Protocol (ICMP) message to a destination IP address or IP network and reports the response time if that destination can be reached. Use this command to determine whether you can reach a destination on an IP network. If the PING command completes successfully, you installed TCP/IP correctly. |
| <b>NETSTAT</b>  | Helps you obtain information about your own IP interfaces, for example, to list IP addresses and TCP/IP routing tables in use at your workstation.  |
| <b>IFCONFIG</b> | Initializes an IP interface at your workstation and allows you to query its status. This is helpful if you have to find out if an IP interface is active.   |
| <b>ARP</b>      | Displays the IP-to-hardware address mapping table at your workstation. You can manually add a hardware address to your ARP table, for instance if you lose connection to a router.  |

**Note:** Be careful when updating the ARP table. Entering a wrong IP-to-hardware address pair results in communication errors.

## Communication problems

**HOSTNAME** Displays the IP address of your workstation

**NSLOOKUP** Displays the default server name and IP address.

### APPC-providing products

This section describes problem determination in products involved in APPC communication.

#### VTAM Buffer Trace

You can use VTAM buffer tracing to record the flow of data between logical units in the CICS environment. The trace entries include the netname of the terminal (logical unit) to which they relate. For details on VTAM buffer tracing and other VTAM problem determination functions, see the appropriate book in the VTAM library.

#### eNetwork Communications Server

The *eNetwork Communications Server Problem Determination Guide*, explains the APPC return codes, and further information is given in the *eNetwork Communications Server APPC Programming Guide and Reference*.

**eNetwork Communications Server trace:** eNetwork Communications Server provides a broad range of problem determination functions covering different types of APIs, connections, and services. The trace selections for eNetwork Communications Server include the following:

**API** Contains APIs supported by eNetwork Communications Server, for example, APPC or EHLLAPI.

**DLC** Contains data link control (DLC) types that pass through eNetwork Communications Server. When eNetwork Communications Server encounters DLC of the type you selected, it copies the data to a trace record. For example, if you select *IBMTRNET* from the DLC list, *IBMTRNET DLC* is traced every time it passes through eNetwork Communications Server.

#### Events

Contains functions that are performed within eNetwork Communications Server. When eNetwork Communications Server encounters the event you selected, it copies the data to a trace record. For example, if you selected *APPC SNDRCV* from the events list, a trace record is written every time APPC sends or receives data through the network.

You can enable the APPC API trace by doing one of the following:

- Selecting APPC from the API column of the Problem Determination Aids—Trace Services window



- Using the CMTRACE command
- Specifying the APPC\_API parameter on the DEFINE\_TRACE common services verb

**eNetwork Communications Server dumps:** The eNetwork Communications Server dump service is a diagnostic tool that helps you identify the source of certain types of communication problems. You can use dump services to copy the data stored in a portion of memory being used by a communications component.

### The APING utility

In an APPC environment, you can use the APING utility to test a connection. APING exchanges data packets with a partner computer over APPC and times how long the data transfer takes. It can be used to get a first estimate of the session setup time between two computers and the throughput and turnaround time on that APPC session. You can use APING to determine whether a session can be set up between two computers and to display extensive error information if session allocation fails. APING consists of two transaction programs: APING, which runs on the client side, and APINGD, which runs on the server side.

### Other tools and utilities

Apart from the trace and dump functions described above, other diagnostic tools and techniques are available.

### First Failure Support Technology/2

First Failure Support Technology/2 (FFST/2) is a software problem diagnosis tool for OS/2 system software and applications. It is designed to facilitate the capture of problem diagnosis information when a problem occurs.

Unless you are an advanced user, your primary use of FFST/2 will be the system error log and the message log formatter. Refer to the *eNetwork Communications Server Problem Determination Guide* for a complete description of the tasks you can perform with FFST/2.

---

## Troubleshooting

This section provides information to help you solve some common problems with CICS Universal Clients. The problems are discussed under the following headings:

- “Starting clients and terminals” on page 124
- “NetBIOS communication problems” on page 124

## Troubleshooting

- “TCP/IP communication problems” on page 125
- “APPC communication problems” on page 126.

For information on error messages, refer to *CICS Universal Clients Messages*.

### Starting clients and terminals

The following are solutions to problems that can occur when starting clients and terminals:

#### **CICS for OS/2 server reports “transaction CCIN failed”**

CCIN and CTIN are two transactions required on the CICS for OS/2 server to enable a CICS Universal Client (or another CICS for OS/2 server) to attach using TCP/IP. For TCP/IP support, either CICS for OS/2 Version 2.0.1, or CICS for OS/2 Version 2.0 with CSD4 are required. Although the two Versions are essentially identical, CCIN and CTIN must be defined in the program control table for Version 2.0. Both transactions call the same program—FAATPPCI.

#### **A CICSTERM request has gone to the wrong server**

If you do not specify the `/s=servername` option on the CICSTERM command, a CICSTERM request is issued against the first server listed in the configuration file, even if it is not yet activated. The *servername* should be as specified in the configuration file.

### NetBIOS communication problems

The following are solutions to problems that can occur when communicating over NetBIOS:

#### **There are only a few NetBIOS resources when running CICS for OS/2 server and CICS Universal Client for OS/2 on the same workstation**

The CICS for OS/2 server is probably using all of the NetBIOS resources for itself and leaving none for CICS Universal Client for OS/2. To avoid restricting the number of commands and sessions, reduce the values of *n1*, *n2*, and *n3* in the `\IBMLAN\IBMLAN.INI` file, or `PROTOCOL.INI` file.

```
[networks]
net1 = netbeui$, 0, LM10, n1,n2,n3
```

However, if you are using the LAN services program, leave sufficient resources for the LAN server/requester. Also in `IBMLAN.INI`, the `wrkheuristics` entry should contain 0 at position #18.

**The NetBIOS connection fails. The trace reports:  
CCL4217 NetBIOS command failed for conversation 0/0: Cmd=0090, Rc=0014**

This is a general return code returned by NetBIOS. Check that:

- the **Maximum NetBIOS Systems** value is greater than 0 in the System Initialization Table for CICS for OS/2
- If using CICS Universal Client for Windows NT you have configured your NetBIOS for the correct LAN type, for example, Novell or IBM. (The default is Novell.)

**Message FAA5573W received from CICS for OS/2:  
NetBIOS Listener Name exists elsewhere on adapter 0**

If the application ID specified in the CICS for OS/2 System Initialization Table is not unique in the network, message FAA5573W is displayed during startup of CICS for OS/2. This usually happens if you use the default value (CICSOS2) for the application ID.

To solve this problem, select **Continue** in the CICS for OS/2 console window and change the application ID according to your standards. You must shut down and restart CICS for OS/2 for the changes to take effect.

### **TCP/IP communication problems**

The following are solutions to problems that can occur when communicating over TCP/IP:

**CICS Universal Client for OS/2: error CCL4401 TCP/IP not available**

For CICS Universal Client for OS/2 ensure that TCP/IP is one of the protocols listed in the LAN Adapter and Protocol Support (LAPS) current configuration list. LAPS is part of Network Transport Services/2 (NTS/2). For more information, see the NTS/2 documentation.

**Message CCL4404 TCP/IP (to 'CICSTCP') unable to resolve name: RC=2**

The CICS server, (in this example, CICSTCP) could not be resolved by the TCP/IP protocol driver. Ensure that your domain name server and router address information is correct, and that any names and IP addresses in the TCP/IP ETC\HOSTS file are correct.

**The TCP/IP connection to a CICS for OS/2 server failed**

Check that:

- the **Maximum TCP/IP Systems** value is greater than zero in the SIT for the CICS for OS/2.

## Troubleshooting

- both the client and server are using the same port number.
- CCIN and CTIN have been defined on your CICS for OS/2 server. Refer to the CCIN problem in “Starting clients and terminals” on page 124 for more information.

### APPC communication problems

The following are solutions to problems that can occur when communicating over APPC:

#### **CCIN not recognized, CTIN not recognized**

The CCIN transaction installs your CICS Universal Clients definition on the CICS server. The CTIN transaction installs your client terminal definition on the CICS server. These transactions must be available on the CICS server if it supports the EPI, because the EPI implies CICS 3270 terminal emulation and CICS 3270 printer emulation. For information on which CICS servers support EPI and hence CICS 3270 emulation, see Table 2 on page 10. You can ignore these messages if your CICS server does not support the EPI.

#### **A CICSTERM command for a mainframe CICS server failed**

CICSTERM and CICSPRNT use CICS 3270 emulation. Some mainframe CICS servers do not support CICS 3270 emulation. For information on which CICS servers support CICS 3270 emulation, see Table 2 on page 10.

---

## Chapter 8. Distributing CICS Universal Clients in a network

**This chapter discusses**

- “Configuration, distribution, installation”
- “Performing installation and maintenance” on page 128.

CICS Universal Client for OS/2 is a CID-enabled product, that is, it conforms to IBM’s configuration, installation, and distribution (CID) architecture. Such products can be automatically installed and maintained. Indeed, CICS Universal Client for OS/2 can be tailored and automatically distributed to workstations on a LAN, without any action being required at those workstations.

---

### Configuration, distribution, installation

The CID architecture is a set of standards that allows installation of software on workstations from code residing at a LAN server. This saves you the bothersome task of responding to prompts and messages at each client workstation.

Installation can be ***lightly attended***, meaning that entering the install command is all that is required at the workstation. The lightly attended installation process is described in this chapter.

Installation can also be ***unattended***, meaning that you can automatically install CICS Universal Clients on many workstations without any end user action being required. This makes installation and maintenance of CICS Universal Clients much easier. Unattended CID is performed under the control of a ***software distribution manager (SDM)***, which remotely installs the CID-enabled product. Examples of SDMs are NetView Distribution Manager (NVDM/2), Start/2, and SystemView for OS/2. Unattended CID is beyond the scope of this book; refer to the documentation for your SDM-providing product for a full description of unattended CID.

For both lightly attended and unattended CID, the installation and maintenance options are supplied in a ***response file***. You invoke the installation program with parameters that identify the source and target

## Configuration, distribution, installation

directories, and the response file. The progress of the installation is recorded in an error log and history log. Return codes are notified to the software distribution manager.

---

### Performing installation and maintenance

The procedure for lightly attended CID is as follows:

1. Copy the CICS Universal Clients files to a local or LAN drive, for example:  

```
XCOPY e:\*.* /s /e
```
2. Edit the supplied response file CICSCLI.RSP or create a new response file. The supplied file CICSCLI.RSP can handle only install and delete actions.
3. Start the installation, by entering INSTALL together with the required parameters. For example:

```
INSTALL /A:I /R:F:\INSTCICS\CICSCLI.RSP  
/S:F:\INSTCICS  
/L1:F:\INSTCICS\CICSCLI.OUT  
/L2:F:\INSTCICS\CICSCLI.HIS /X
```

In this example, the CICS Universal Client is installed from the redirected drive F: on the server, according to the options supplied in the response file CICSCLI.RSP. The response file specifies the drive and directory in which to install the client. Errors are logged in CICSCLI.OUT and the history log is contained in CICSCLI.HIS. The /X parameter specifies that the installation is noninteractive.

**Note:** The base level of the CICS Universal Client is installed using the INSTALL command, while corrective service software is installed using the EPFINSTS command.

4. The installation proceeds automatically.

### The installation and maintenance parameters

The installation and maintenance parameters are as follows:

```
INSTALL /A:action  
/G:include-path  
/L1:error-log  
/L2:history-log  
/P:"product-name"  
/R:response-file  
/S:source-location  
/T:install-target-directory  
/TU:update CONFIG.SYS directory  
/X
```

**Note:** You can enter the parameters in any order. Equals signs (=) can be used instead of colons (:) in the parameters. Values can be uppercase or lowercase.

### */A:action*

Specifies the action to be performed by the installation program. If you specify this parameter, the main window of the installation program is not displayed. Valid values for *action* are:

- D** Delete an installed CICS Universal Client system
- I** Install a new CICS Universal Client system
- R** Restore a backed-up CICS Universal Client system
- U** Update an installed CICS Universal Client system

These actions are the same as those provided by the installation utility (see for example, “Updating your CICS Universal Client system” on page 22).

**Note:** The installation program is not supplied with corrective service software. Therefore, if you are updating, restoring, or deleting a CICS Universal Client, you must have access to the installation program used to install the CICS Universal Client.

### */G:include-path*

Specifies the drive and path of a general response file to be included by the specific response file. For more information about response files, see “Response files” on page 130.

### */L1:error-log*

Specifies the drive, path, and file name of the error log file. The error log contains messages associated with installation, including confirmations and error messages. Messages are written to the error log even if you specify the */X* parameter.

You should specify the drive and path where the installation program is running. If you do not specify the */L1* parameter, no error log is maintained. If you specify an error log that already exists, it is appended to.

Example:

```
/L1:F:\CICSCLI\CICSCLI.LOG
```

### */L2:history-log*

Specifies the drive, path, and file name of the history log file. The history log contains an entry for each file transferred, each workplace shell object icon created, and each Installation Exit run.

## Performing installation and maintenance

You should specify the drive and path where the installation program is running. If you do not specify the /L2 parameter, no history log is maintained. If you specify a history log that already exists, it is appended to.

Example:

/L2:F:\CICSCLI\CICSCLI.HIS

**/P:***“product-name”*

Specifies the catalog name of the product to perform the action on.

**/R:***response-file*

Specifies the drive, path, and file name of a response file, (see “Response files”).

Example:.

/R:L:\CICSINS\CICSCLI.RSP

**/S:***source location*

Specifies the drive and path containing the source files to be installed or updated.

**/T:***install-target-directory*

Specifies the drive and path that CICS Universal Client files are installed into. If you specify this parameter, it overrides the FILE path specified in the response files.

**/TU:***update target CONFIG.SYS directory*

Specifies the drive and path of the target CONFIG.SYS (OS/2)

**/X** Specifies that the installation is fully automatic.

When you specify this parameter, no progress indicator panel is shown and error messages are logged in the error log file. (You specify the path name of the error log file using the /L1 parameter.) If you do not specify all of the information required for the action to complete, an error occurs.

If you do not specify the /X parameter, the user is prompted for any information that the installation program needs to complete the action. In this interactive mode of operation, progress indication is shown and error messages are displayed in secondary windows.

## Response files

A response file is an ASCII text file containing answers to the options that you select when you install or maintain a CICS Universal Client system. This allows installation and maintenance to be performed automatically, without interaction.

In a response file you can specify:



- Whether environment information should be updated automatically
- The CICS Universal Clients components to be installed
- The path for installation or maintenance
- Whether existing files should be overwritten
- Whether only backup versions of CICS Universal Clients should be deleted

You can use the response file supplied with CICS Universal Client for OS/2 to install or delete CICS Universal Clients. To perform other actions you must prepare your own response files using a suitable editor.

Normally, you will have only one response file, specified by the /R parameter of the installation program. However, you may sometimes need to install or update a CICS Universal Client on a workstation with different options. In this case you can use one response file as a general response file to be included in a specific response file. The specific response file would contain options specific to a particular workstation, and the general response file would contain options common to all workstations. For example, you might use a general response file and a specific response file to install a particular component on only certain workstations. You use the /R parameter to specify the specific response file, and the /G parameter to specify the location of the general response file.

### Response file keywords

The following keywords are supported in response files for CICS Universal Clients:

<b>CFGUPDATE</b>	Specifies whether the installation program automatically updates the environment information in CONFIG.SYS. Valid values for this keyword are:  <b>AUTO</b> Automatically updates the environment information  <b>MANUAL</b> Does not update the environment information  <b>Actions:</b> INSTALL, UPDATE, DELETE.
<b>COMP</b>	Specifies the name of a component to perform an action on. Valid values are: <ul style="list-style-type: none"><li>• Program Files</li><li>• Programming Samples</li><li>• Toolkit</li><li>• PDF Documentation</li></ul>

## Performing installation and maintenance

You do not need to enclose the component names within quotes. The name is not case-sensitive.

**Actions:** ALL actions.

### DELETEBACKUP

Specifies whether to delete only the backup version of CICS client or the entire product. Valid values for this keyword are YES (delete backup version only) and NO (delete current version).

**Actions:** DELETE.

### FILE

Specifies the drive and directory in which to install CICS Universal Client; for example, the C:\CICSLI directory. This keyword applies to the installation action only.

**Actions:** INSTALL.

### INCLUDE

Specifies which general response files to include with a specific response file. The format of this keyword is:

INCLUDE = *filespec*

where *filespec* specifies the general response file to be included. If the file specification contains any global characters (\* or ?), the first file found that matches the specification is included. If the specification is not valid, no general response file is included.

**Note:** You should not have more than five levels of included response files.

**Actions:** All actions.

### OVERWRITE

Specifies whether to overwrite files automatically during installation. Valid values for this keyword are YES and NO.

**Actions:** INSTALL, UPDATE.

### SAVEBACKUP

Specifies whether to save a backup version of the CICS Universal Client when it is updated. Valid values for this keyword are YES and NO.

**Actions:** UPDATE.

### CICSCLIINI

Specifies the name of a customized CICSCLI configuration file. If no customized configuration file is to be provided, the value of the parameter must be set to an asterisk (\*).

Before the installation is started, the customized .INI file must be placed in (or below) the source directory from which the installation is taking place. The value of the parameter must specify the path relative to this source directory. The .INI file is copied to the \BIN subdirectory on the target, and the CICSCLI environment variable set in CONFIG.SYS. Thus, if the customized file (MYCLI.INI) is placed in the subdirectory \MYFILES under the source directory, the value of the parameter would be CICSCLIINI = MYFILES\MYCLI.INI.

**Actions:** INSTALL.

### CICSCOLINI

Specifies the name of a customized CICSCOL color mapping file used to tailor the colors for the CICSTERM terminal emulator. If no customized mapping file is to be provided, the value of the parameter must be set to an asterisk (\*).

The customized file is copied into the \BIN subdirectory on the target system, and the CICSCOL environment variable set.

See the CICSCLIINI keyword description for details of how to specify the file name, and where the customized file should be placed before installation.

**Actions:** INSTALL.

### CICSKEYINI

Specifies the name of a customized CICSKEY keyboard mapping file used to tailor the keys for the CICSTERM terminal emulator. If no customized mapping file is to be provided, the value of the parameter must be set to an asterisk (\*).

The customized file is copied into the \BIN subdirectory on the target system, and the CICSKEY environment variable set. See the CICSCLIINI keyword description for details of

## Performing installation and maintenance

how to specify the file name, and where the customized file should be placed before installation.

**Actions:** INSTALL.

### MESSAGELANG

Specifies the language of messages sent to the CICS Universal Clients. Valid values are:

<b>FR</b>	French
<b>IT</b>	Italian
<b>JP</b>	Japanese
<b>KR</b>	Korean
<b>ES</b>	Spanish
<b>US</b>	US English

**Actions:** INSTALL.

### Response file structure

There are two kinds of line in a response file:

- **Comment lines**

Comment lines are either blank or start with an asterisk (\*) or a semicolon (;).

- **Response lines**

Response lines are used to determine the options and configurations to install on the target system.

Response lines have the following syntax:

keyword = value

Keyword-value pairs can be in any order. However, there can be only one pair per line.

You can enter keywords in uppercase or lowercase letters. You cannot include spaces within keywords.

The maximum line length in a response file is 255 characters.

### Return codes

All return codes are two-byte hexadecimal values as follows:

**00 00** Successful termination. No messages were logged.

- 00 04** Successful termination. Warning messages were logged. Restarting the workstation operating system is not required.
- 00 08** Successful termination. Error messages were logged. Restarting the workstation operating system is not required.
- 00 12** Successful termination. Severe error messages were logged. Restarting the workstation operating system is not required.
- FE 00** Successful termination. Restarting the workstation operating system is required. Do not invoke the installation and maintenance program again.
- FE 04** Successful termination. Warning messages were logged. Restarting the workstation operating system is required. Do not invoke the installation and maintenance program again.
- FE 12** Successful termination. Severe error messages were logged. Restarting the workstation operating system is required. Do not invoke the installation and maintenance program again.
- FE yy** Successful termination. yy can vary within the range 00 to FF. Restarting the workstation operating system is required. Do not invoke the installation and maintenance program again.

Messages are logged only when the /X parameter is used.

---

### Using SystemView for OS/2 to install CICS Universal Clients

You can use SystemView for OS/2 to install CICS Universal Client for OS/2 on unattended workstations using the statements included in a file called a *software profile* and a response file. A software profile is used by SystemView for OS/2 to invoke an application installation program. A response file is used by the installation program to get all of the data it needs during the installation process.

Using SystemView for OS/2, you specify the information for the software profiles and response files when you customize the application definition file (.ADF), which defines how to install and configure CICS Universal Clients. This file is shipped with CICS Universal Client for OS/2 as CICSCLI.ADF.

For a complete description of software profiles and response files, refer to the SystemView for OS/2 online help or the *SystemView for OS/2 Up and Running!*, SH19-4184.

### Overview of SystemView for OS/2 installation

To install CICS Universal Clients on workstations using SystemView for OS/2, you need to:

## Using SystemView for OS/2 to install CICS Universal Clients

1. Copy CICS Universal Clients files to the code server, that is, the workstation from which you will install CICS Universal Clients
2. Add CICS Universal Clients information to the SystemView for OS/2 software library
3. Configure CICS Universal Clients in the SystemView for OS/2 software library
4. Distribute and install the resulting software object on target server workstations using SystemView for OS/2.

The first step is described in this book. For information on the other steps, refer to the SystemView for OS/2 online help or to the SystemView for OS/2 documentation.

### Copying CICS Universal Clients to the code server

To set up the code server so that you can install CICS Universal Clients using SystemView for OS/2, use the XCOPY command.

For example, to copy the files for CICS Universal Client for OS/2 from the CD-ROM in the E: drive to the D: drive, you would enter this command:

```
XCOPY E:\CLIENTS\I86_CLNT\CLIENTS\OS2\ D:\CID\IMG\CLIENTS\OS2 /S
```

This command would copy the files from the \CLIENTS\OS2 subdirectory on the CD-ROM in the E: drive to the D:drive, putting the files under the directory \CID\IMG\CLIENTS\OS2.

The /S indicates that all of the subdirectories and their contents, and the contents of the root directory should be copied.

---

## Chapter 9. Migration issues

This chapter discusses issues concerning the migration of applications between different versions of CICS Clients.

---

### Application compatibility

CICS Universal Clients do not support the *development* of 16-bit applications, but old 16-bit applications can still be run.

#### ECI and EPI compatibility

OS/2 ECI and EPI applications written for Distributed Feature clients of CICS for OS/2 are portable at the binary level to CICS Universal Client for OS/2.

##### Notes:

1. The header files are now called `cics_eci.h` (C), `cicseci.cbl` (COBOL), `cics_eci.inc` (PL/I); and `cics_epi.h` (C), `cicsepi.cbl` (COBOL), `cics_epi.inc` (PL/I). References to `faaecih.h` and `faaepih.h`, or `faaeciw.cbl` and `faaepiw.cbl` must be changed.
2. The CICS ECI entry point is now called `CICS_ExternalCall()` for C and PL/I, and `CICSEXTERNALCALL` for COBOL.
3. The CICS EPI entry point is now called `CICSEXTERNALCALL` for 32-bit COBOL. Any calls to `FaaExternalCall()` must be changed.
4. The correct link library is now called `CCLOS232.LIB` (for 32-bit OS/2). Attempts to link with `FAACLIB.LIB` or `FAACIC32.LIB` will fail.

ECI and EPI client applications are portable between members of the CICS Universal Clients family, provided the application does not contain operating-system-specific calls.

For more information, refer to the *CICS Family: Client/Server Programming* book.

## **Application compatibility**



---

## Appendix A. CICS Universal Clients data conversion

The ECI and EPI allow non-CICS applications running in a client system to gain access to CICS facilities and data managed by a CICS server system.

Character data may have to be converted as it is passed between client and server; for example data is encoded in ASCII on a CICS Universal Client system and in EBCDIC on a CICS/390 server system. Data conversion is performed by the server system.

The possible ASCII and EBCDIC encoding schemes are described in detail in SC09-2190, the Character Data Representation Architecture Reference and Registry (CRDA). In summary, each encoding scheme is identified by a Coded Character Set Identifier (CCSID) which defines a set of graphic characters and the Code Page Global Identifier (CPGID) which specifies the code points used to represent the graphic characters.

The data managed by the server system may be accessed from several client systems each of which uses a different ASCII encoding scheme. To support such access, each client system must be able to supply a CCSID 'tag' in order that the data is converted correctly.

---

### Supported conversions

The method used to perform data conversion depends on the server platform. The range of data conversions supported also depends on the platform. The following table has been extracted from *Communicating from CICS on System/390*, SC33-1697 ASCII and EBCDIC CCSIDs are assigned to geographic or linguistic groups.

Data conversion is supported between ASCII and EBCDIC where both CCSIDs belong to the same group. The intent is that other CICS servers will provide equivalent support.

#### Arabic:

	CCSID	CPGID	
ASCII	00864	00864	PC data: Arabic
	01089	01089	ISO 8859-6: Arabic
	01256	01256	MS Windows: Arabic
EBCDIC	00420	00420	Host: Arabic

#### Baltic Rim:

## Data conversion

	CCSID	CPGID	
ASCII	00921	00921	PC data: Latvia, Lithuania
	01257	01257	MS Windows: Baltic Rim
EBCDIC	01112	01112	Host: Latvia, Lithuania

### Cyrillic:

	CCSID	CPGID	
ASCII	00866	00866	PC data: Cyrillic
	00915	00915	ISO 8859-5: Cyrillic
	01251	01251	MS Windows: Cyrillic
EBCDIC	01025	01025	Host: Cyrillic multilingual

### Estonia:

	CCSID	CPGID	
ASCII	00922	00922	PC data: Estonia
	01257	01257	MS Windows: Baltic Rim
EBCDIC	01122	01122	Host: Estonia

### Greek:

	CCSID	CPGID	
ASCII	00869	00869	PC data: Greece
	00813	00813	ISO 8859-7: Grek
	01253	01253	MS Windows: Greek
EBCDIC	00875	00875	Host: Grek

### Hebrew:

	CCSID	CPGID	
ASCII	00856	00856	PC data: Hebrew
	00916	00916	ISO 8859-8: Hebrew
	01255	01255	MS Windows: Hebrew
EBCDIC	00424	00424	Host: Hebrew

### Latin-1:

	CCSID	CPGID	
ASCII	00437	00437	PC data: USA, many other countries
	00819	00819	ISO 8859-1: Latin-1 countries
	00850	00850	PC data: Latin-1 countries
	01252	01252	MS Windows: Latin-1 countries
EBCDIC	00037	00037	Host: USA, Canada, etc
	00273	00273	Host: Austria, Germany
	00277	00277	Host: Denmark, Norway
	00278	00278	Host: Finland, Sweden
	00280	00280	Host: Italy
	00284	00284	Host: Spain, Latin America (Spanish)
	00285	00285	Host: United Kingdom
	00297	00297	Host: France
	00500	00500	Host: International Latin-1
	00871	00871	Host: Iceland
	01047	01047	Host: Latin-1 Open Systems

### Latin-1 including euro support:

	CCSID	CPGID	
ASCII	00858	00858	PC data: Latin-1 countries
	05348	01252	MS Windows: Latin-1 countries, version 2
EBCDIC	01140	01140	Host: USA, Canada, etc
	01141	01141	Host: Austria, Germany
	01142	01142	Host: Denmark, Norway
	01143	01143	Host: Finland, Sweden
	01144	01144	Host: Italy
	01145	01145	Host: Spain, Latin America (Spanish)
	01146	01146	Host: United Kingdom
	01147	01147	Host: France
	01148	01148	Host: International Latin-1
	01149	01149	Host: Iceland

**Latin-2:**

	CCSID	CPGID	
ASCII	00852	00852	PC data: Latin-2 multilingual
	00912	00912	ISO 8859-2: Latin-2 multilingual
	01250	01250	MS windows: Latin-2
EBCDIC	00870	00870	Host: Latin-2 multilingual

**Latin-5:**

	CCSID	CPGID	
ASCII	00857	00857	PC data: Latin-5 (Turkey)
	00920	00920	ISO 8859-9: Latin-5 (Turkey)
	01254	01254	MS Windows: Turkey
EBCDIC	01026	01026	Host: Latin-5 (Turkey)

**Latin-9:**

	CCSID	CPGID	
ASCII	00923	00923	ISO 8859-15: Latin-9
EBCDIC	00924	00924	Host: Latin-9

**Japanese:**

	CCSID	CPGID	
ASCII	00932	00897	PC data: SBCS
		00301	PC data: DBCS
	00942	01041	PC data: extended SBCS
		00301	PC data: DBCS
	00943	00897	PC data: SBCS
		00941	PC data: DBCS for Open environment
	00954	00895	G0: JIS X201 Roman
		00952	G1: JIS X208-1990
		00896	G2: JIS X201 Katakana
		00953	G3: JIS X212
EBCDIC	00930	00290	Host: Katakana, extended SBCS
		00300	Host: DBCS
	00931	00037	Host: Latin-1, SBCS
		00300	Host: DBCS
	00939	01027	Host: Latin-1, extended SBCS
		00300	Host: DBCS

## Data conversion

### Korea:

	CCSID	CPGID	
ASCII	00934	00891	PC data: SBCS
		00926	PC data: DBCS
	00944	01040	PC data: extended SBCS
		00926	PC data: DBCS
	00949	01088	PC data: SBCS, IBM KS code
		00951	PC data: DBCS, IBM KS code
	00970	00367	G0: ASCII
		00971	G1: KSC X5601-1989
EBCDIC	01363	01126	MS Windows: Korean SBCS
		01362	MS Windows: Korean DBCS
	00933	00833	Host: extended SBCS
		00834	Host: DBCS

### Simplified Chinese:

	CCSID	CPGID	
ASCII	00946	01042	PC data: extended SBCS
		00928	PC data: DBCS
	01381	01115	PC data: extended SBCS, IBM GB
		01380	PC data: DBCS, IBMGB
	01383	00367	G0: ASCII
		01382	G1: GB 2312-80 set
	01386	01114	PC data: SBCS, S-Chinese GBK, T-Chinese IBM BIG-5
		01385	PC data: DBCS, S-Chinese GBK
EBCDIC	00935	00836	Host: extended SBCS
		00837	Host: DBCS

### Traditional Chinese:

	CCSID	CPGID	
ASCII	00938	00904	PC data: SBCS
		00927	PC data: DBCS
	00948	01043	PC data: extended SBCS
		00927	PC data: DBCS
	00950	01114	PC data: SBCS, IBM BIG-5
		00947	PC data: DBCS
	00964	00367	G0: ASCII
		00960	G1: CNS 11643 plane 1
EBCDIC	00937	00961	G2: CNS 11643 plane 2
		00037	Host: Latin-1 SBCS
		00835	Host: DBCS

---

## Appendix B. The CICS Transaction Gateway and CICS Universal Clients library

This chapter lists all the CICS Transaction Gateway, CICS Universal Clients, and related books, and discusses the various forms in which they are available.

The headings in this chapter are:

- “CICS Transaction Gateway books”
- “CICS Universal Clients books” on page 144
- “CICS Family publications” on page 144
- “Book filenames” on page 145
- “Sample configuration documents” on page 145
- “Other publications” on page 146
- “Viewing the online documentation” on page 146

---

### CICS Transaction Gateway books

- *CICS Transaction Gateway for OS/2 Administration, SC34-5590*  
This book describes the administration of the CICS Transaction Gateway for OS/2.
- *CICS Transaction Gateway for Windows Administration, SC34-5589*  
This book describes the administration of CICS Transaction Gateway for Windows 98 and CICS Transaction Gateway for Windows NT.
- *CICS Transaction Gateway for AIX Administration, SC34-5591*  
This book describes the administration of the CICS Transaction Gateway for AIX.
- *CICS Transaction Gateway for Solaris Administration, SC34-5592*  
This book describes the administration of the CICS Transaction Gateway for Solaris.
- *CICS Transaction Gateway for OS/390 Administration, SC34-5528*  
This book describes the administration of the CICS Transaction Gateway for OS/390.
- *CICS Transaction Gateway Messages*  
This online book lists and explains the error messages that can be generated by CICS Transaction Gateway.  
You cannot order this book.

## The CICS Transaction Gateway and CICS Universal Clients library

- *CICS Transaction Gateway Programming*, SC34-5594

This book provides an introduction to Java programming with the CICS Transaction Gateway.

There are also additional HTML pages that contain programming reference information.

---

### CICS Universal Clients books

- *CICS Universal Client for OS/2 Administration*, SC34-5450

This book describes the administration of the CICS Universal Client for OS/2.

- *CICS Universal Client for Windows Administration*, SC34-5449

This book describes the administration of the CICS Universal Client for Windows 98 and CICS Universal Client for Windows NT.

- *CICS Universal Client for AIX Administration*, SC34-5348

This book describes the administration of the CICS Universal Client for AIX.

- *CICS Universal Client for Solaris Administration*, SC34-5451

This book describes the administration of the CICS Universal Client for Solaris.

- *CICS Universal Clients Messages*

This online book lists and explains the error and trace messages that can be generated by CICS Universal Clients.

You cannot order this book.

- *CICS Universal Clients C++ Programming*, SC33-1923

This book describes how to write object oriented programs for the ECI and EPI in the C++ language.

- *CICS Universal Clients COM Automation Programming*, SC33-1924

This book describes how to write object oriented programs for the ECI and EPI according to the Component Object Model (COM) standard.

---

### CICS Family publications

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

This book describes the programming interfaces associated with CICS client/server Programming— the External Call Interface (ECI), the External Presentation Interface (EPI), and the External Security Interface (ESI). It is intended for application designers and programmers who wish to develop client applications to communicate with CICS server systems.

---

## Book filenames

Table 12 show the softcopy filenames of the CICS Transaction Gateway and CICS Universal Client books.

*Table 12. CICS Transaction Gateway and CICS Universal Clients books and file names*

Book title	File name
CICS Universal Clients Messages	CCLHAB
CICS Universal Client for AIX Administration	CCLHAD
CICS Universal Client for OS/2 Administration	CCLHAE
CICS Universal Client for Windows Administration	CCLHAF
CICS Universal Client for Solaris Administration	CCLHAG
CICS Transaction Gateway for OS/390 Administration	CCLHAI
CICS Transaction Gateway Messages	CCLHAJ
CICS Transaction Gateway Programming	CCLHAK
CICS Transaction Gateway for Windows Administration	CCLHAL
CICS Transaction Gateway for OS/2 Administration	CCLHAM
CICS Transaction Gateway for AIX Administration	CCLHAN
CICS Transaction Gateway for Solaris Administration	CCLHAO
CICS Universal Clients C++ Programming	CCLHAP
CICS Universal Clients COM Automation Programming	CCLHAQ
CICS Family: Client/Server Programming	DFHZAD
<b>Note:</b> The File names in this table do not include the 2-digit suffix.	

---

## Sample configuration documents

A number of sample configuration documents are available in the Portable Document Format (PDF) format.

These documents provide step-by-step guidance to help you, for example, in configuring your CICS Universal Clients for communication with CICS servers, using various protocols. They provide detailed instructions that extend the information in the CICS Transaction Gateway and CICS Universal Client libraries.

As more sample configuration documents become available, you can download them from our Web site; go to:

<http://www.ibm.com/software/ts/cics/>

## Other publications

and follow the **Library** link.

---

## Other publications

The following International Technical Support Organization (ITSO) Redbook publication contains many examples of client/server configurations:

- *Revealed! CICS Transaction Gateway with more CICS Clients Unmasked, SG24-5277*

This book supersedes the following book:

- *CICS Clients Unmasked, GG24-2534*

You can obtain ITSO Redbooks from a number of sources. For the latest information, see:

<http://www.ibm.com/redbooks/>

You can find information on CICS products at:

<http://www.ibm.com/software/ts/cics/>

---

## Viewing the online documentation

You can access all of the documentation provided with CICS Transaction Gateway and CICS Universal Client in our online library. You need Adobe Acrobat Reader and a suitable Web browser to use the online library (and you may need to configure these).

To get to the online library:

- On Windows and OS/2, select the **Documentation** icon.
- On AIX and Solaris, run the **ctgdoc** script.

and the library home page is displayed.

The online library allows you to link to:

- CICS Transaction Gateway and CICS Universal Clients books in PDF format.
- Programming reference documentation in HyperText Markup Language (HTML) files (provided for CICS Transaction Gateway only).
- README files.
- Sample configuration documents in PDF format.
- Translated books in PDF format. (You may find that not all books are translated for your language.)
- The CICS Web site.



Guidance information on using Acrobat Reader is also provided.

Updated versions of the books may be provided from time to time, check our Web site at:

<http://www.ibm.com/software/ts/cics/>

and follow the **Library** link.

### Viewing PDF books

The PDF information provides powerful functions for:

- Navigating through the information. There are hypertext links within PDF documents, and to other PDF documents and Web pages.
- Searching for specific information.
- Printing all or part of PDF documents on a PostScript printer.

You can find out more about Acrobat Reader at the Adobe Web site:

<http://www.adobe.com/acrobat/>

## Viewing the online documentation

---

## Appendix C. Notices

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

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

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

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

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

**The following paragraph does not apply in the United Kingdom or any other country where such provisions are inconsistent with local law:**  
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore this statement may not apply to you.

This information 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 information. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this information at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM United Kingdom Laboratories, MP151, Hursley Park, Winchester, Hampshire, England, SO21 2JN. Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information 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.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

---

## Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

AIX	Anynet
CICS	CICS/MVS
DB/2	eNetwork
IBM	Interspace
LAN Distance	MVS
MVS/ESA	OS/2
OS/390	OS/400
RETAIN	System/390
TXSeries	VisualAge

VSE/ESA

VTAM

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

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

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Other company, product, and service names may be trademarks or service marks of others.



---

## Glossary

This glossary defines special terms used in this book.

If you cannot find the word you are looking for, try the *IBM Dictionary of Computing*.

### A

**APING.** A command used to verify an APPC link.

**APPC.** Advanced program-to-program communication. An implementation of the SNA/SDLC LU 6.2 protocol that allows interconnected systems to communicate and share the processing of programs. The APPC protocol is used by IBM CICS Universal Clients to communicate with CICS server systems.

**application definition file.** In software preparation, a file that contains information about how to install a particular CID-enabled software product, and how to gather configuration parameters to be used in the response file and software object.

**application server.** A CICS run-time process that executes CICS tasks.

**ATI.** See *automatic transaction initiation*.

**Attach.** In SNA, the request unit that flows on a session to initiate a conversation.

**Attach Manager.** The component of APPC that matches Attaches received from remote computers to accepts issued by local programs.

**autoinstall.** A facility that enables terminal definitions to be created automatically in a CICS server on request. The definition is deleted when the client terminal is logged off.

**automatic transaction initiation (ATI).** The process whereby a transaction request made internally within a CICS system or network leads

to the scheduling of the transaction. An ATI request from a CICS Universal Client can cause the initiation of a transaction in a CICS server system.

### B

**BIND.** In SNA, a request to activate a session between two logical units (LUs).

**business logic.** The part of a distributed application that is concerned with the application logic rather than the user interface of the application. See *presentation logic*.

### C

**CICS Gateway for Java.** In CICS Universal Clients Version 2, an interface between a Java program and CICS that allows a Java-enabled browser to dynamically download a Java Applet and transparently access CICS data. This has been replaced by the CICS Transaction Gateway.

**CICS on open systems.** A term used to refer generically to the products: CICS for HP-UX, CICS for Sun Solaris, TXSeries for AIX, TXSeries for HP-UX, and TXSeries for Solaris.

**CICS on System/390.** A term used to refer generically to the products: CICS for MVS/ESA, CICS/MVS, and CICS for VSE/ESA.

**cicsprnt.** A command that allows you to define a printer terminal on the CICS Universal Client workstation. The command enables CICS applications running on the server to direct output to the client-attached printer.

**cicsteld.** A command that starts a CICS telnet daemon. The telnet daemon allows a telnet 3270 client to connect to a CICS server.

## Glossary

**cicsterm.** A command that provides 3270 emulation and enables connection to a CICS region.

**CID.** Configuration, installation, and distribution.

**CID methodology.** An IBM-specified way to install and configure products on, or remove products from, remote workstations and hosts. CID-enabled products use response files and redirected installation and configuration to eliminate or reduce user interaction with the CID-enabled product.

**Change-Number-of-Sessions (CNOS).** An internal transaction program that regulates the number of parallel sessions between the partner LUs with specific characteristics.

**client/server.** A distributed application design model, in which the client sends a request to its partner, the server, which executes the request and returns the results to the client. This design model is associated typically with cooperative processing.

**client configuration file.** A file containing configuration information used to inform the CICS Universal Client of the CICS servers it can connect to, and the communication protocols to be used.

**CNOS.** See *Change-Number-of-Sessions*.

**color mapping file.** A file used by IBM CICS Universal Clients to customize the 3270 screen color attributes on client workstations.

**COMMAREA.** A communication area that is used for passing data both between programs within a transaction and between transactions.

**connection.** In data communication, an association established between functional units for conveying information. In Open Systems Interconnection architecture, an association established by a given layer between two or more entities of the next higher layer for the purpose of data transfer. In TCP/IP, the path between two protocol application that provides reliable data stream delivery service. In Internet,

a connection extends from a TCP application on one system to a TCP application on another system.

**control table.** In CICS, a storage area used to describe or define the configuration or operation of the system.

**conversation.** The communication between a CICS Universal Client and a CICS server. In APPC, a connection between two transaction programs over an LU-LU session that allows them to communicate with each other while processing a transaction. See also *session*.

**conversation security.** In APPC, a process that allows validation of a user ID or group ID and password before establishing a connection.

**cooperative processing.** A subset of *distributed processing*, in which at least the user interface (presentation) aspect of the application runs on a programmable workstation.

## D

**data link control (DLC).** A set of rules used by nodes on a data link (such as an SDLC link or a token ring) to accomplish an orderly exchange of information.

**DBCS.** See *double-byte character set*.

**dependent LU.** A logical unit (LU) that can receive, but not send a BIND, and which supports only single sessions. See *BIND*.

**distributed application.** An application for which the component application programs are distributed between two or more interconnected processors.

**Distributed Feature client.** The CICS clients supplied with CICS for OS/2 Version 2.0. These are functionally inferior to IBM CICS Universal Clients.

**distributed processing.** The processing of different parts of the same application in different systems, on one or more processors.



**domain.** In the Internet, a part of a naming hierarchy in which the domain name consists of a sequence of names (labels) separated by periods (dots).

**domain name.** In TCP/IP, a name of a host system in a network.

**Domain Name Server.** In TCP/IP, a server program that supplies name-to-address translation by mapping domain names to internet addresses. Synonymous with name server.

**dotted decimal notation.** The syntactical representation for a 32-bit integer that consists of four 8-bit numbers written in base 10 with periods (dots) separating them. It is used to represent IP addresses.

**double-byte character set (DBCS).** A set of characters in which each character is represented by 2 bytes. Languages such as Japanese, Chinese and Korean, which contain more symbols than can be represented by 256 code points, require double-byte character sets. Because each character requires 2 bytes, the typing, display, and printing of DBCS characters requires hardware and programs that support DBCS. Contrast with *single-byte character set*.

## E

**ECI.** See *external call interface*.

**emulator, emulation program.** A program that allows a host system to communicate with a workstation in the same way as it would with the emulated terminal. In IBM CICS Universal Clients the terminal emulation function allows client workstations to run CICS transactions that use 3270 data flows.

**EPI.** See *external presentation interface*.

**Ethernet.** A 10-megabit baseband local area network that allows multiple stations to access the transmission medium at will without prior coordination, avoids contention by using carrier sense and deference, and resolves contention by using collision detection and transmission.

Ethernet uses carrier sense multiple access with collision detection (CSMA/CD).

**external call interface (ECI).** A facility that allows a non-CICS program to run a CICS program. Data is exchanged in a COMMAREA as for normal CICS interprogram communication.

**external presentation interface (EPI).** A facility that allows a non-CICS program to appear to CICS as one or more standard 3270 terminals. 3270 data can be presented to the user by emulating a 3270 terminal or by using a graphical user interface.

**external security interface (ESI).** A facility that enables client applications to verify and change passwords for user IDs on CICS servers.

## G

**gateway.** A device that connects two dissimilar LANs, or that connects a LAN to a wide area network (WAN), midrange computer, or mainframe computer. A gateway device has its own processor and memory and may perform protocol conversion. A gateway handles multiple communication sessions simultaneously.

## H

**host.** In the Internet suite of protocols, an end system. The end system can be any system; it does not have to be a mainframe.

**host address.** An IP address that is used to identify a host in an internet.

**host ID.** In TCP/IP, that part of the Internet address that defines the host on the network. The length of the host ID depends on the type of network or network class (A, B, or C).

**host name.** In the Internet suite of protocols, the domain name of a host.

## Glossary

### I

**independent LU.** A logical unit (LU) that can both send and receive a BIND, and which supports single, parallel, and multiple sessions. See *BIND*.

**internet.** A collection of networks interconnected by a set of routers that allow them to function as a single, large network. Contrast with *Internet*.

**Internet.** The internet administered by the Internet Architecture Board (IAB), consisting of large national backbone networks and many regional and campus networks all over the world. The Internet uses the Internet suite of protocols.

**Internet address.** See *IP address*.

**Internet Architecture Board.** The technical body that oversees the development of the internet suite of protocols known as TCP/IP.

**Internet Packet Exchange (IPX).** In Novell NetWare LANs, a communication protocol that sends data packets to requested destinations.

**Internet Protocol (IP).** In TCP/IP, a protocol that routes data from its source to its destination in an Internet environment.

**Interoperability.** The capability to communicate, execute programs, or transfer data among various functional units in a way that requires the user to have little or no knowledge of the unique characteristics of those units.

**IP.** Internet Protocol.

**IP address.** The 32-bit address defined by the internet protocol, standard 5, Request for Comment (RFC) 791. It is usually represented in dotted decimal notation.

**IPX.** Internet Packet Exchange.

### K

**keyboard mapping file.** A file used by IBM CICS Universal Clients to customize the keyboard mapping of client workstations.

### L

**LAN.** See *local area network*.

**link support layer (LSL).** The part of Novell's Open Data-Link Interface (ODI) that allows the LAN driver for a network adapter to service more than one protocol. See *Open Data-Link Interface (ODI)*.

**local area network (LAN).** A network of workstations, or terminals, or both, where all the connected systems are relatively near to each other. See *wide area network*.

**logical unit (LU).** In SNA, a port through which an end user accesses the SNA network in order to communicate with another end user and through which the end user accesses the functions provided by system services control points (SSCP). An LU can support at least two sessions, one with an SSCP and one with another LU, and may be capable of supporting many sessions with other logical units. See *network addressable unit*, *primary logical unit*, *secondary logical unit*.

**LU-LU session.** In SNA, a session between two logical units (LUs) in an SNA network. It provides communication between two end users, or between an end user and an LU services component.

**LU-LU session type 6.2.** In SNA, a type of session for communication between peer systems. Synonymous with APPC protocol.

**LU 6.2.** See *APPC*.

### M

**MAC.** Medium access control.

**medium access control (MAC) sublayer.** One of two sublayers of the ISO Open Systems Interconnection data link layer proposed for local area networks by the IEEE Project 802 Committee on Local Area Networks and the European Computer Manufacturers Association (ECMA). It provides functions that depend on the topology of the network and uses services of the physical layer to provide services to the logical link control (LLC) sublayer. The OSI data link layer corresponds to the SNA data link control layer.

**mode.** In SNA, a set of parameters that defines the characteristics of a session between two LUs.

## N

**name server.** In TCP/IP, synonym for Domain Name Server. In Internet communications, the host that translates host names into their respective internet addresses when requested by the hosts on the network. A physical device, and its associated software, that enables a processor or controller to be connected to a network.

**NDIS.** Network Driver Interface Specification.

**NetBIOS.** Network Basic Input/Output System. The standard interface to networks, IBM personal computers (PCs), and compatible PCs, that is used on LANs to provide message, print server, and file-server functions. The IBM NetBIOS application program interface (API) provides a programming interface to the LAN so that an application program can have LAN communication without knowledge and responsibility of the data link control (DLC) interface.

**network address.** In SNA, an address, consisting of subarea and element fields, that identifies a link, link station, or network addressable unit (NAU). Subarea nodes use network addresses; peripheral nodes use local addresses. The boundary function in the subarea node to which a peripheral node is attached transforms local addresses to network addresses and vice versa. See also *network name*.

**network addressable unit (NAU).** In SNA, a logical unit, a physical unit, or a system services

control point. The NAU is the origin or the destination of information transmitted by the path control network. See also *logical unit*, *network address*, *network name*.

**network name.** In SNA, the symbolic identifier by which end users refer to a network addressable unit (NAU), link station, or link. See also *network address*.

### Network Driver Interface Specification

**(NDIS).** A specification developed jointly by Microsoft and 3Com that lets an OS/2 system support one or more network adapters and protocol drivers. The device driver functions at the medium access control (MAC) layer. This kind of driver is also called an NDIS MAC driver.

**node type.** In SNA, a designation of a node according to the protocols it supports and the network addressable units (NAUs) it can contain. Four types are defined: 1, 2, 4, and 5. Type 1 and type 2 nodes are peripheral nodes; type 4 and type 5 nodes are subarea nodes.

## O

**ODI.** Open Data-Link Interface.

**Open Data-Link Interface (ODI).** Novell's architecture that allows multiple LAN drivers and protocols to coexist in network systems. The ODI specification defines the set of interface and software modules used to decouple device drivers from protocol stacks and to enable multiple protocol stacks to share transparently the network hardware.

## P

**pacing.** A technique by which a receiving station controls the rate of transmission of a sending station to prevent overrun.

**packet internet groper (PING).** In Internet communications, a program used in TCP/IP networks to test the ability to reach destinations

## Glossary

by sending the destinations an Internet Control Message Protocol (ICMP) echo request and waiting for a reply.

**parallel session.** In SNA, two or more concurrently active sessions between the same two LUs using different pairs of network addresses. Each session can have independent session parameters.

**PING.** Packet internet groper. The command that sends an ICMP Echo Request packet to a gateway, router, or host with the expectation of receiving a reply.

**partner logical unit (PLU).** In SNA, the remote participant in a session.

**partner transaction program.** The transaction program engaged in an APPC conversation with a local transaction program.

**PLU.** Primary logical unit. Partner logical unit.

**port.** An endpoint for communication between devices, generally referring to a logical connection. A 16-bit number identifying a particular Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) resource within a given TCP/IP node.

**presentation logic.** The part of a distributed application that is concerned with the user interface of the application. See *business logic*.

**primary logical unit (PLU).** In SNA, the logical unit (LU) that contains the primary half-session for a particular LU-LU session. Contrast with secondary logical unit. See also *logical unit*.

**Note:** A particular logical unit may contain primary and secondary half-sessions for different active LU-LU sessions.

**protocol boundary.** The signals and rules governing interactions between two components within a node.

## R

**request unit (RU).** In SNA, a message unit that contains control information such as a request code, or function management (FM) headers, end-user data, or both.

**request/response unit.** A generic term for a request unit or a response unit. See also *request unit* and *response unit*.

**response file.** A file that contains a set of predefined answers to questions asked by a program and that is used in place of a user dialog. See *CID methodology*.

**response unit (RU).** A message unit that acknowledges a request unit; it may contain prefix information received in a request unit.

**RU.** Request unit. Response unit.

## S

**SAA.** See *Systems Application Architecture*.

**SBCS.** See *single-byte character set*.

**SDLC.** See *Synchronous Data Link Control*.

**secondary logical unit (SLU).** In SNA, the logical unit (LU) that contains the secondary half-session for a particular LU-LU session. Contrast with primary logical unit. See also *logical unit*.

**session.** In SNA, a logical connection between two network addressable units (NAUs) that can be activated, tailored to provide various protocols, and deactivated as requested. The session activation request and response can determine options for such things as rate and concurrency of data exchange, control of contention and error recovery, and characteristics of the data stream. Sessions compete for network resources, such as links within the path control network. For routing purposes, each session is identified by the network or local addresses of the session partners. See *LU-LU session*, *LU-LU session type*.

**session limit.** In SNA, the maximum number of concurrently active LU-LU sessions that a particular logical unit (LU) can support.

**side information.** System-defined values required by CPI-C to initialize a conversation. These values include the partner LU name, mode name, and transaction program name. See *Common Programming Interface for Communication (CPI-C)*.

**single-byte character set (SBCS).** A character set in which each character is represented by 1 byte. Contrast with double-byte character set.

**SNA gateway.** A gateway that handles multiple APPC communication sessions simultaneously. See *gateway*.

**SNA sense data.** An SNA-defined encoding of error information. In SNA, the data sent with a negative response, indicating the reason for the response.

**SNASVCMG mode name.** The SNA service manager mode name. This is the architecturally-defined mode name identifying sessions on which CNOS is exchanged. Most APPC-providing products predefine SNASVCMG sessions.

**subnet.** In TCP/IP, a part of a network that is identified by a portion of the Internet address. Synonym for subnetwork.

**subnet address.** In Internet communications, an extension to the basic IP addressing scheme where a portion of the host address is interpreted as the local network address.

**Synchronous Data Link Control (SDLC).** A communications protocol for managing synchronous, code-transparent serial-by-bit information transfer over a link connection.

**Systems Application Architecture (SAA).** A set of IBM software interfaces, conventions, and protocols that provide a framework for designing and developing applications across multiple computing environments.

**Systems Network Architecture (SNA).** The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through, and controlling the configuration and operation of, networks.

**SystemView for OS/2.** A set of system management applications that:

- Provides the administrator with functions to monitor and manage OS/2 managed systems on a LAN, keep track of hardware and software resources; and make software available to managed systems, and
- Provides the end user at the managed system with the functions to perform a variety of systems management tasks on the local system in cooperation with the SystemView manager.

## T

**TCP62.** A protocol mapper provided by AnyNet that allows partner LU62 applications to communicate over a TCP/IP network.

**TCP/IP.** Transmission Control Protocol/Internet Protocol.

**terminal emulation.** The capability of a microcomputer or personal computer to operate as if it were a particular type of terminal linked to a processing unit and to access data.

**token ring.** A network with a ring topology that passes tokens from one attaching device to another; for example, the IBM Token-Ring network.

**trace.** A record of data that provides a history of events that occurred in a system. The process of recording the sequence in which the statements in a program are executed and, optionally, are the values of the program variables used in the statements.

**transaction program.** A program that uses the APPC application program interface (API) to communicate with a partner application program in the same node or at a partner node.

## Glossary

**Transmission Control Protocol/Internet Protocol (TCP/IP).** A set of communications protocols that support peer-to-peer connectivity functions for both local and wide area networks. TCP/IP can be used for client/server links between IBM CICS Universal Clients and, for example, CICS for OS/2, CICS for Windows NT.

**type 2.0 node.** An SNA node that attaches to a subarea network as a peripheral node and provides full end-user services but no intermediate routing services.

**type 2.1 node.** An SNA node that can be configured as an endpoint or intermediate routing node in a network, or as a peripheral node attached to a subarea network.

## U

**user session.** Any APPC session other than a SNASVCMG session.

## V

**verb.** A reserved word that expresses an action to be taken by an application programming interface (API), a compiler, or an object program. In SNA, the general name for a transaction program's request for communication services.

## W

**wide area network (WAN).** A network that provides communication services to a geographic area larger than that served by a local area network or a metropolitan area network, and that may use or provide public communication facilities.



---

# Index

## Numerics

3270 data streams 92, 97

3745 cluster controller 8

## A

Adapter configuration setting 65  
adapters 15

advanced program-to-program  
communication (APPC) 13, 37

Anynet domain name suffix  
configuration setting 64

APAR (authorized program analysis  
report)

authorization 111

closing 112

documentation needed 108

process 111

submitting 111

APPC (advanced

program-to-program

communication) 13, 37

application compatibility 137

application definition file 135

Applid configuration setting 55

Applid configuration setting  
(NetBIOS) 65

automatic installation of CICS  
Universal Clients 127

## B

backing up CICS Universal  
Client 23

benefits of CICS Universal  
Clients 3

binary trace formatter 114

books 143

CICS Transaction Gateway and

CICS Universal Clients

library 143

online 146

PDF 147

printed 147

business logic 4

## C

CCIN 124

CCLLOG.HLP 119

CCLMSG.HLP 119

CF2 configuration file, eNetwork  
Communications Server 39

CFG configuration file, eNetwork  
Communications Server 38

CICS printer icon 99

CICS server problem  
determination 120

CICS server PTF requirements 17

CICS servers 1

CICS servers, connections 7

CICS telnet daemon 102

CICS Universal Client

trace analysis 115

CICS Universal Client for OS/2

installation 20

operation 80

CICS Universal Clients icons 80

CICS Universal Clients in Novell  
NetWare LANs 39

CICSCLI.ADF 135

cicscli.bin 112

CICSCLI command 80

cicscli.log 58, 119

CICSCLI.MRF 135

cicscli.trc 112

CICSCLI.VAR 135

CICSCOL environment variable 74

CICSFTRC utility 114

CICSKEY environment variable 70

CICSPRNT command 97

CICSTELD command 102

CICSTERM command 92

CID (configuration, installation and  
distribution) 127

client control 5

client control process, restarting 82

client control process, starting 81

client control process, stopping 81

client initialization file

specifying which to use 83

client/server communication, setting  
up 27

client/server connections 10

client trace 83, 113

client trace file 115

Client trace file configuration  
setting 66

coaxial connections 8

code pages 17

Codepage identifier override

configuration setting 57

coexistence with Novell's IPX 41

color mapping file 74, 92

commands

CICSCLI 80

CICSPRNT 97

CICSTERM 92

COMMAREA 4

communication

problems 120

setting up a client 27

communication, mobile 9

communication, setting up  
client/server connection 27

communication protocols

APPC 13

NetBIOS 13

TCP/IP 13

Communications Manager/2

configuring 38

dumps 123

trace 122

compatibility, ECI and EPI 137

CONFIG.SYS 20, 23

configuration

configuration file 52

configuration tool 52

configuration conversion tool 67

configuration examples 27

configuration file 52

configuration files, eNetwork

Communications Server 38

configuration installation and

distribution (CID) 127

configuration settings

Adapter 65

Anynet domain name suffix 64

Applid 55, 65

Client trace file 66

Codepage identifier override 57

Connection timeout 61

Description 58

Enable popups 58

Fully qualified CP name or

template 63

Hostname or IP address 60

Initial transaction 59

IP address mask for CP name

(optional) 64

- configuration settings (*continued*)
  - IP address mask for LU name template (optional) 62
  - Local LU name 61
  - Local LU name or template 62
  - Log file 58
  - Maximum buffer size 55
  - Maximum Client wrap size 66
  - Maximum logical SNA sessions 64
  - Maximum requests 56
  - Maximum servers 30, 56
  - Maximum SNA RU size 64
  - Mode name 61, 62
  - Model terminal definition 59
  - NetBIOS 60
  - Partner LU name 61, 62
  - Port 60
  - Print command 56
  - Print file 57
  - Remote node inactivity timeout 64
  - Send TCP/IP Keepalive packets 61
  - Server name 58
  - Server retry interval 58
  - SNA 59
  - SNA pacing size 64
  - TCP/IP 59
  - TCP62 59
  - Terminal exit 55
  - Trace 65
  - Trace settings 113
  - Use LU alias names 61
  - Use upper case security 59
- configuration tool 52
- connecting to CICS servers 81, 88
- Connection timeout configuration setting 61
- controller, 3745 8
- conversation, APPC 39
- Corrective Service Diskette (CSD) 112
- corrective service software 22
- creating icons for clients 25
- CSD (Corrective Service Diskette) 112
- ctgcfg command 52
- CTGCONV, conversion tool 67
- CTIN 124
- customizing
  - keyboard 69
  - screen colors 74

## D

- DBCS (double-byte character set) 16
- defining 3270 printer terminal emulator characteristics 98
- defining 3270 terminal emulator characteristics 92
- definition checklists 27
- Delete panel 24
- deleting a CICS Universal Client system 24
- Description configuration setting 58
- disabling the display of messages 87
- disabling the display of pop-up messages 87
- Disk space panel 21
- disk space required 15
- displaying command syntax 87
- documentation 143
  - HTML 146
  - PDF 147
- documentation of problems 108
- double-byte character set (DBCS) 16
- dynamic name generation (TCP62) 63

## E

- ECI (External Call Interface) 4
- ECI and EPI compatibility 137
- emulation of NetBIOS over IPX 45
- Enable popups configuration setting 58
- enabling the display of pop-up messages 87
- eNetwork Communications Server 38
- environment variables
  - CICSCOL 74
  - CICSKEY 70
- EPI (External Presentation Interface) 4
- error log 129
  - cicscli.log 119
  - client error log 119
  - server error log 120
- error processing 119
- ESI (External Security Interface) 5
- Euro support 57
- example list of CICS servers 87
- EXEC CICS RETURN TRANSID IMMEDIATE command 94, 99
- External Call Interface (ECI) 4

- External Presentation Interface (EPI) 4
- External Security Interface (ESI) 5

## F

- FAATPPCI 124
- free memory 55
- Fully qualified CP name or template configuration setting 63

## G

- gateway 8
- glossary of terms and abbreviations 153
- graphical user interface (GUI) 4
- GUI (graphical user interface) 4

## H

- hardcopy books 147
- hardware requirements 15
- history log 25, 129
- Hostname or IP address configuration setting 60
- HTML (HyperText Markup Language) 146
- HTML documentation, viewing 146
- HyperText Markup Language (HTML) 146

## I

- icons, client 25
- INI configuration file, eNetwork Communications Server 38
- initial transaction 92
- Initial transaction configuration setting 59
- initialization files
  - cicscol.ini 74
  - cicskey.ini 70
- Install-directories panel 21
- Install panel 20
- installation
  - automatic 24
  - CICS Universal Client 20
- installation and maintenance program 22
- installation parameters 128
- Internet Packet Exchange (IPX) protocol 40
- IP address, TCP/IP 30
- IP address mask for CP name (optional) configuration setting 64
- IP address mask for LU name template (optional) configuration setting 62
- IPX (Internet Packet Exchange) protocol 40



## K

KeepAlive packets 37  
keyboard mapping file 69, 92  
keys specific to particular  
keyboards 71  
keywords for response files 131

## L

LAN adapter and protocol support  
(LAPS) 46  
LAN Distance 9  
language support for client  
applications 17  
LAPS (LAN adapter and protocol  
support) 46  
LAPS configuration file  
(PROTOCOL.INI) 38  
link support layer (LSL) 40  
listing connected servers 87  
Local LU name configuration  
setting 61  
Local LU name or template  
configuration setting 62  
Log file configuration setting 58  
logical unit (LU) 37  
LSL (link support layer) 40  
LU6.2 13

## M

maintenance program 22  
Maximum buffer size configuration  
setting 55  
Maximum Client wrap size  
configuration setting 66  
Maximum logical SNA sessions  
configuration setting 64  
Maximum requests configuration  
setting 56  
Maximum servers configuration  
setting 56  
Maximum SNA RU size  
configuration setting 64  
memory requirements 55  
messages, disabling the display  
of 87  
migration 137  
mobile communication 9  
mode definitions, APPC 37  
Mode name configuration  
setting 61, 62  
Model terminal definition  
configuration setting 59  
multiple protocol configurations 9

## N

NDF configuration file, eNetwork  
Communications Server 38  
NDIS (Network Driver Interface  
Specification) 41  
NetBIOS (Network Basic  
Input/Output System) 13, 29  
NetBIOS configuration setting 60  
NetBIOS emulation 45  
NetBIOS listener 125  
NetBIOS resources 30  
network adapters 15  
Network Basic Input/Output System  
(NetBIOS) 13  
Network Driver Interface  
Specification (NDIS) 41  
node definitions (.NDF) file,  
eNetwork Communications  
Server 38  
Novell  
client/server support 41  
IPX 41  
NetBIOS emulator 45  
Novell's IPX, coexistence with 41  
Novell NetWare LANs, CICS  
Universal Clients in 39

## O

object oriented programming 2  
ODI (Open Data-Link Interface) 40  
ODINSUP 41  
online books, PDF 147  
online documentation, HTML 146  
online help, for end user  
messages, 119  
online help, for trace and log  
messages 119  
Open Data-Link Interface (ODI) 40  
operation  
CICS Universal Client for  
OS/2 80  
options  
CICSCLI command 88  
CICSPRNT command 100  
CICSTERM command 94

## P

Partner LU name configuration  
setting 61, 62  
password expiry management  
(PEM) 86  
PDF (Portable Document  
Format) 147  
PDF books, viewing 147

PEM (password expiry  
management) 86  
planning 7  
PMR (Problem Management  
Record) 106  
pop-up messages, disabling the  
display of 87  
pop-up messages, enabling the  
display of 87  
Port configuration setting 60  
port number, TCP/IP 30  
Portable Document Format  
(PDF) 147  
PostScript books 147  
presentation logic 4  
Print command configuration  
setting 56  
print file, processing 92, 98  
Print file configuration setting 57  
print terminal emulator, starting 98  
printer support 5  
printer terminal emulator  
characteristics, defining 98  
problem determination 105  
Problem Management Record  
(PMR) 106  
problem reporting  
documentation of problems 108  
information needed 109  
report sheet 109  
support organization 106  
problems, common 124  
problems, communication 120  
program support 106  
programming languages 17  
protocol configurations, multiple 9  
PROTOCOL.INI, LAPS configuration  
file 38, 47  
PTF (Program Temporary Fix) 112  
publications, CICS Transaction  
Gateway and CICS Universal  
Clients library 143

## R

railroad diagrams xiv  
Remote node inactivity timeout  
configuration setting 64  
requirements, hardware 15  
requirements, software 16  
response files 24, 127, 130  
keywords 131  
structure 134  
restarting the client 82  
restrictions, using CICS servers 11  
RETAIN database 106

- RETAIN problem management system
  - APARs 111
  - Problem Management Record 106
- return codes 134

## S

- SDLC connections 8
- SDM (software distribution manager) 127
- SEC configuration file, eNetwork Communications Server 39
- security 59, 83
- Send TCP/IP Keepalive packets configuration setting 61
- Server name configuration setting 58
- Server retry interval configuration setting 58
- servers, listing 87
- service levels 22
- SERVICES file, TCP/IP 30
- setting up client/server communication 27
- signon capable terminals 17
- SNA (Systems Network Architecture) 13
- SNA configuration setting 59
- SNA gateway 7
- SNA pacing size configuration setting 64
- softcopy books, PDF 147
- software distribution manager (SDM) 127
- software profile 135
- software requirements 16
- starting a 3270 print terminal emulator 98
- starting a 3270 terminal emulator 92
- starting the client control process 81
- stopping a terminal emulator 93
- stopping the client control process 81
- structure of response files 134
- support organization
  - problem reporting 109
  - program support 106
  - sending documentation to 111
- syntax notation xiv
- systems network architecture (SNA) configuration files, eNetwork Communications Server 38

- Systems Network Architecture (SNA) 13
- SystemView for OS/2 135

## T

- TCP/IP (Transmission Control Protocol/Internet Protocol) 13, 30
- TCP/IP configuration setting 59
- TCP62 configuration setting 59
- TCP62 protocol 35
- TCP62 support 14
- telnet support 102
- terminal emulation 3
- terminal emulator, starting 92
- terminal emulator, stopping 93
- terminal emulator characteristics, defining 92
- Terminal exit configuration setting 55
- testing the client/server link 78
- token ring connections 8
- TOKEN.SYS 41
- Trace configuration setting 65
- trace file 112
- Trace settings, configuration tool 113
- tracing IBM CICS Universal Clients 112
- transaction programs (TP), APPC 37
- transmission control protocol/internet protocol (TCP/IP) 13
- Transmission Control Protocol/Internet Protocol (TCP/IP) 13, 30
- troubleshooting 123, 124

## U

- unattended CID 135
- Update panel 23
- updating CICS Universal Client 22
- Use LU alias names configuration setting 61
- Use upper case security configuration setting 59

## V

- viewing online documentation 146
- VTAM
  - buffer trace 122





Program Number: 5648-B42



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

SC34-5450-01

