

CICS® Universal Client for Solaris™



Administration

Version 3.1

CICS® Universal Client for Solaris™



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 103.

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 SC34-5451-00.

© **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

Figures.	v	National language support	15
Tables.	vii	Using X-Windows from a remote system	16
About this book	ix	Chapter 4. Setting up client/server communication	19
Who should read this book	ix	TCP/IP configuration	19
Conventions and terminology used in this book	x	Definition checklists	20
Prerequisite and related information	xii	Verifying the TCP/IP installation	21
CICS Universal Clients publications	xii	APPC configuration	22
CICS publications	xii	Configuring APPC for CICS Universal Client for Solaris	23
APPC-related publications	xiii	Chapter 5. Customization	25
How to send your comments	xiii	Renaming the client configuration and mapping files	25
Obtaining books from IBM	xiv	Using the configuration tool	26
The configuration tool interface	26	Configuring Client settings	30
Configuring Server settings	33	Trace settings	35
The configuration conversion tool	37	Using the conversion tool	37
Editing the configuration file	38	CLIENT section	39
SERVER section	39	DRIVER section	39
Customizing the keyboard	40	Keyboard mapping file syntax	40
The keyboard mapping file	40	Customizing the screen colors	44
Color mapping syntax	45	The color mapping file	45
Testing the client/server link	48	Chapter 6. Operation	49
Chapter 1. Overview	1	An introduction to the CICS Universal Clients commands	49
An introduction to IBM CICS Universal Clients	1	The cicscli command	50
CICS Universal Clients communication	2	Starting the client control process	50
The benefits of using CICS Universal Clients	2	Starting connections with additional servers	50
CICS Universal Clients functions	3	Stopping the client control process	51
3270 terminal emulation	3	Restarting the client	51
The External Call Interface	3	Specifying which configuration file to use	52
The External Presentation Interface	4	Turning on client trace	52
The External Security Interface	4		
3270 client printer support	4		
Client control	5		
CICS telnet support	5		
Chapter 2. Planning before installation	7		
CICS Universal Clients and your network	7		
Connections to CICS servers	7		
Communication between CICS Universal Clients and CICS servers	8		
Restrictions on the use of CICS servers	9		
Communication protocols	10		
Which protocols can you use?	10		
Hardware requirements	11		
Software requirements	12		
CICS server PTF requirements	12		
DBCS Multibyte Characters	13		
Chapter 3. Installing CICS Universal Clients	15		
Installing CICS Universal Client for Solaris	15		

Specifying the trace components	52	CICS Universal Client problem determination	80
Turning off client trace	53	CICS Universal Clients trace	80
Setting up security	53	CICS Universal Client error processing	87
Listing the connected servers	55	CICS server problem determination	88
Disabling the display of messages.	56	Communication problem determination.	88
Enabling and disabling the display of console messages	56	Troubleshooting.	89
Displaying the command parameters.	56	Starting clients and terminals	89
cicscli and applications	56	TCP/IP communication problems.	90
cicscli command reference	57	APPC communication problems	91
The cicsterm command	61	Appendix A. CICS Universal Clients data conversion	93
Using cicsterm	61	Supported conversions	93
Stopping a terminal emulator	62	Appendix B. The CICS Transaction Gateway and CICS Universal Clients library.	97
cicsterm and user exits	62	CICS Transaction Gateway books	97
cicsterm and RETURN TRANSID IMMEDIATE.	62	CICS Universal Clients books	98
cicsterm command reference	63	CICS Family publications	98
The cicsprnt command	66	Book filenames	99
Using cicsprnt	66	Sample configuration documents	99
cicsprnt and user exits	67	Other publications	100
cicsprnt and RETURN TRANSID IMMEDIATE.	67	Viewing the online documentation	100
cicsprnt command reference.	68	Viewing PDF books	101
CICS Client telnet terminal support	70	Appendix C. Notices	103
Security considerations for trace and log files	71	Trademarks	104
Chapter 7. Problem determination	73	Glossary	107
Preliminary checks and information	73	Index.	115
Program support	74		
Reporting problems	74		
Documentation of problems.	76		
Locating and compiling information	77		
Submitting the documentation	79		
APARs and fixes	79		

Figures

1. Example messages returned by ping command	22	4. Sample color mapping file	47
2. The configuration tool	27	5. Sample CICS Universal Clients trace	85
3. Sample keyboard mapping file for CICS Universal Clients.	42	6. CICS Universal Clients trace: using an invalid port number	86

Tables

1.	Command syntax conventions	x	5.	Matching definitions: CICS Universal Client for Solaris to TXSeries for AIX using TCP/IP for Solaris	21
2.	Communication between CICS Universal Client for Solaris and CICS servers and functions.	8	6.	CICS Universal Client keys that can be mapped.	41
3.	Hardware requirements for CICS Universal Client for Solaris	11	7.	Sample problem reporting sheet.	77
4.	Software requirements for CICS Universal Client for Solaris	12	8.	CICS Transaction Gateway and CICS Universal Clients books and file names	99

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.
- 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.

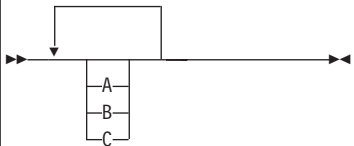
Conventions and terminology used in this book

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 <i>must</i> 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>Note: For CICS, unless otherwise stated, this representation means only that more than one item can be selected.</p>
<p>Punctuation and uppercase characters</p>	<p>Code exactly as shown.</p>
<p>Lowercase characters appearing like <i>this</i></p>	<p>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.</p>

A glossary is provided at the back of 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[®] 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 97. 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[®] 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[®] Problem Determination Guide, GC33-1693

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

CICS for OS/400:[®] 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/>

APPC-related publications

The following books are concerned with products providing APPC support:

Systems Network Architecture (SNA)

SNA Formats, GA27-3136

Systems Network Architecture Technical Overview, GC30-3073

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
- 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/pb1/pb1>

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 2
- “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:

- Transmission control protocol/Internet protocol (TCP/IP)
- Advanced program-to-program communication (APPC)

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 26).

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; for more information, see the *CICS Universal Clients C++ Programming* book.

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

The benefits of using CICS Universal Clients

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.

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.

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 8) 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 functions

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 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 8 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 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:

- 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 70.

Client control

Chapter 2. Planning before installation

This chapter describes the following:

- “CICS Universal Clients and your network”
- “Communication protocols” on page 10
- “Hardware requirements” on page 11
- “Software requirements” on page 12

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 8 shows which CICS servers the CICS Universal Clients can connect to, the protocols used, and the functions supported.

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.

CICS Universal Clients and your network

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 Solaris 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	○	○	●	○	●	○	●
Transaction Server for OS/390 Version 1.x	○	○	●	○	●	●	●
CICS for VSE/ESA Version 2.3	○	○	●	○	●	○	●
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. EPI Emulation is supported on CICS for MVS/ESA Version 4.1 with APAR PN68409.
4. APAR PN79262 is required for DBCS support in CICS for MVS/ESA.
5. 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 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:

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[®] operating system, but is now supported on virtually every platform.

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.

Note: CICS Universal Client for Solaris cannot use the Network Basic Input/Output System (NetBIOS) or TCP62 protocols.

The protocols that can be used for the various client/server connections are shown in Table 2 on page 8.

Hardware requirements

The following table lists the hardware requirements of CICS Universal Client for Solaris.

Table 3. Hardware requirements for CICS Universal Client for Solaris

Processor type
• SPARC
Terminals
Any type of terminal or X/Terminal that can attach to the processor is supported
Memory requirements
A minimum of 16MB
Disk space requirements
22MB

Software requirements

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 Solaris

Table 4. Software requirements for CICS Universal Client for Solaris

Operating system version
Sun Solaris Version 2.5.1 and later
TCP/IP support
TCP/IP support is provided by the Solaris operating system.
SNA support
APPC support is provided by: <ul style="list-style-type: none">• SunLink SNA/PTP Version 9.0 or later The following patches are required: <ul style="list-style-type: none">• For Solaris 2.5.1, the Recommended Patch Cluster for Solaris 2.5.1• For Solaris 2.6, the Recommended Patch Cluster for Solaris 2.6
Application development language support
The following language compilers are supported: <ul style="list-style-type: none">• Sun Workshop Compiler C/C++ Version 3.0, or later• Net Express Version 3.0 (Micro Focus Object COBOL Version 4.1)• IBM VisualAge Interspace Version 5.2

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.

DBCS Multibyte Characters

Some characters in certain codesets are represented with 3 or more bytes. CICS Universal Client for AIX and CICS Universal Client for Solaris do not support multibyte characters that are more than 2 bytes in length. If you try to display such characters on a CICS terminal, you will get unpredictable results.

If you are running on an AIX or Solaris-unique locale, you may experience problems when connecting to certain CICS servers. The table below lists the relevant client-server combinations.

CICS Client Codepage	CICS Server Platform	CICS Server Codepage
ja_JP (33722)	OS/2	932
ja_JP (33722)	Windows NT	932
ja_JP (33722)	AIX	932
ko_KR (970)	OS/2	949
ko_KR (970)	Windows	949
zh_TW (964)	OS/2	950
zh_TW (964)	AIX	950
zh_CN (1383)	OS/2	1381
zh_CN (1383)	Windows NT	1381

Software requirements

Chapter 3. Installing CICS Universal Clients

This chapter describes how to install CICS Universal Client for Solaris.

The CICS Universal Client for Solaris is distributed as a compressed tar file containing the library files, commands, messages and customisation files (.INI) in a single directory tree.

You can download CICS Universal Client for Solaris from the following Web site: <http://service.software.ibm.com/dl/cclient/ccli3-w>

You must adhere to the licensing conditions mentioned on this Web site.

Installing CICS Universal Client for Solaris

To install CICS Universal Client for Solaris

1. Log in as root.
2. Copy the compressed tar file from the SOL directory of your CICS Universal Client for Solaris distribution medium to your hard disk as cli-310s.tar.Z.
3. Uncompress the file and extract the files to the /opt/cicscli directory.

```
uncompress cli-310s.tar.Z
tar -xvf cli-310s.tar
```

Samples are provided in the samples subdirectory.

4. Set up the CICS Universal Client for Solaris by entering:
mkcicscli

Note: You can uninstall CICS Universal Client by using the `mkcicscli -u` command.

National language support

You can select the language in which user messages are displayed by CICS Universal Client for Solaris by entering the command:

```
mkclimsgs XX
```

where *XX* is the message language. To obtain a list of the available languages, enter the command without the *XX* parameter:

Installing CICS Universal Clients

CICS Client for Solaris - User messages

Language	Locale	Code Set
us US English	en_US	8859-1
fr French	fr_FR	8859-1
it Italian	it_IT	8859-1
jp Japanese	ja	eucJP
kr Korean	ko	eucKR
es Spanish	es_ES	8859-1

Example: `mkclimsgs us` for US English

This also shows the locale and code set associated with the language.

To use an alternate code set, you can use the `iconv` routine for the flat file `/opt/cicscli/CCLMSG.TXT`. For example, to convert `/opt/cicscli/CCLMSG.TXT` from code set ISO8859-1 to code set ISO-850 enter:

```
iconv -f ISO8859-1 -t ISO-850 /opt/cicscli/CCLMSG.TXT. > CCLMSG.NEW
```

When you have done this conversion, you can overwrite the `CCLMSG.TXT` file with the new file:

```
mv CCLMSG.NEW /opt/cicscli/CCLMSG.TXT.
```

Using X-Windows from a remote system

When using X-Windows from a remote system, for example, to access the configuration tool and iKeyman, you must set up the `DISPLAY` environment variable to allow the application to display its windows on that system.

On the display system (that is the one that will display the windows), enter the command:

```
xhost +appl
```

where *appl* is the network name of the system being used to run the application.

On the application system, before you run the application, enter the command:

```
DISPLAY=disp:0
```

followed by

```
export DISPLAY
```


| where *disp* is the network name of the system where the windows will be
| displayed (followed by a colon and the display id—normally 0). The
| application windows are then displayed on the *disp* system.

Chapter 4. Setting up client/server communication

This chapter describes the following:

- “TCP/IP configuration”
- “APPC configuration” on page 22

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

The 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 xii.

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

- Enable and set up TCP/IP or APPC communication
- Set up the required settings in the configuration file, (see “Using the configuration tool” on page 26)

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.

TCP/IP configuration

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

Definition checklists

The following shows example definitions for connecting CICS Universal Client for Solaris to a CICS for AIX server:

CICS server definitions

TCP/IP for Solaris Definitions	Ref. Key	Example
HOSTNAME	1	cicstcp.ibm.com
Internet ADDRESS	2	192.113.36.200
Network MASK	3	255.255.255.0
NAMESERVER Internet ADDRESS	4	192.113.36.250
DOMAIN Name	5	acme.ibm.com
TCP/IP service name	6	cicstcp
Port number	7	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	2	192.113.36.200
TCP service name	6	cicstcp

CICS Universal Client Definitions

TCP/IP for Solaris Definitions	Ref. Key	Example
HOSTNAME		myhost.ibm.com
Internet ADDRESS		192.113.36.78
Network MASK	3	255.255.255.0
NAMESERVER Internet ADDRESS	4	192.113.36.250
DOMAIN Name	5	acme.ibm.com
TCP/IP service name	6	cicstcp
Port number	7	0
Protocol type (TCP only)		TCP

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

Table 5. Matching definitions: CICS Universal Client for Solaris to TXSeries for AIX using TCP/IP for Solaris

Ref. Key	TCP/IP for Solaris (Server)	TXSeries for AIX	TCP/IP for Solaris (Client)	Client Configuration Settings	Example
1	HOSTNAME			Host name	cicstcp.ibm.com
2	Internet ADDRESS	TCP adapter address		IP address	192.113.36.200
3	Network MASK		Network MASK		255.255.255.0
4	NAMESERVER Internet ADDRESS		NAMESERVER Internet ADDRESS		192.113.36.250
5	DOMAIN Name		DOMAIN Name		acme.ibm.com
6	TCP/IP service name	TCP service name			cicstcp
7	Port number			Port	0

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.

TCP/IP configuration

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 1. 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.

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
- One or more **mode** definitions to specify sets of session properties that are used in binding APPC sessions
- A **transaction program (TP)** definition for the CRSR transaction. This is required if terminal emulation is supported by the CICS servers, and it is necessary to perform automatic transaction initiation (ATI) against the CICS Universal Client terminals.

Configuring APPC for CICS Universal Client for Solaris

To run the SNA transport, a SNA Server must be installed and configured. This is available within Sun SunLink SNA Server Version 9.

To run SunLink under the Solaris 7 operating system, you require the patches for 32-bit and 64-bit mode operation, as detailed in see “Software requirements” on page 12.

Detailed information on connecting CICS Universal Client for Solaris to CICS Transaction Server for OS/390 is given in any configuration document that is available, see “Sample configuration documents” on page 99.

To enable ATI against CICS Universal Client terminals, the transaction program CRSR must be defined on the SunLink Server. Follow the instructions in the SunLink SNA9.1 PU2.1 Server Configuration Guide. To define the TP using the sungmi utility:

1. Start the SNA administration application, **sungmi**.
2. Open the System and expand the PU2.1 Servers and pu211 entries.
3. Select **Transaction Programs** and **New Transaction Program**, then **Advanced entries**.
4. Enter CRSR as the TP Name.
5. Indicate Parameters are for invocation on any LU.
6. Enter the command path as `/usr/sbin/cc1c1nt CRSR`.
7. Select **OK** to save the new entry.

APPC configuration

Chapter 5. Customization

This chapter describes

- “Renaming the client configuration and mapping files”
- “Using the configuration tool” on page 26
- “Customizing the keyboard” on page 40
- “Customizing the screen colors” on page 44
- “Testing the client/server link” on page 48

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 /opt/cicscli/bin directory by default:

CTGSAMP.INI	A sample client configuration file. (The default configuration file name is CTG.INI.)
CICSKEY.INI	The keyboard mapping file
CICSCOL.INI	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
client configuration file	CICSCLI
keyboard mapping file	CICSKEY
color mapping file	CICSCOL

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 use the configuration tool you must export the display using the commands described in “Using X-Windows from a remote system” on page 16.

To start the configuration tool, 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.

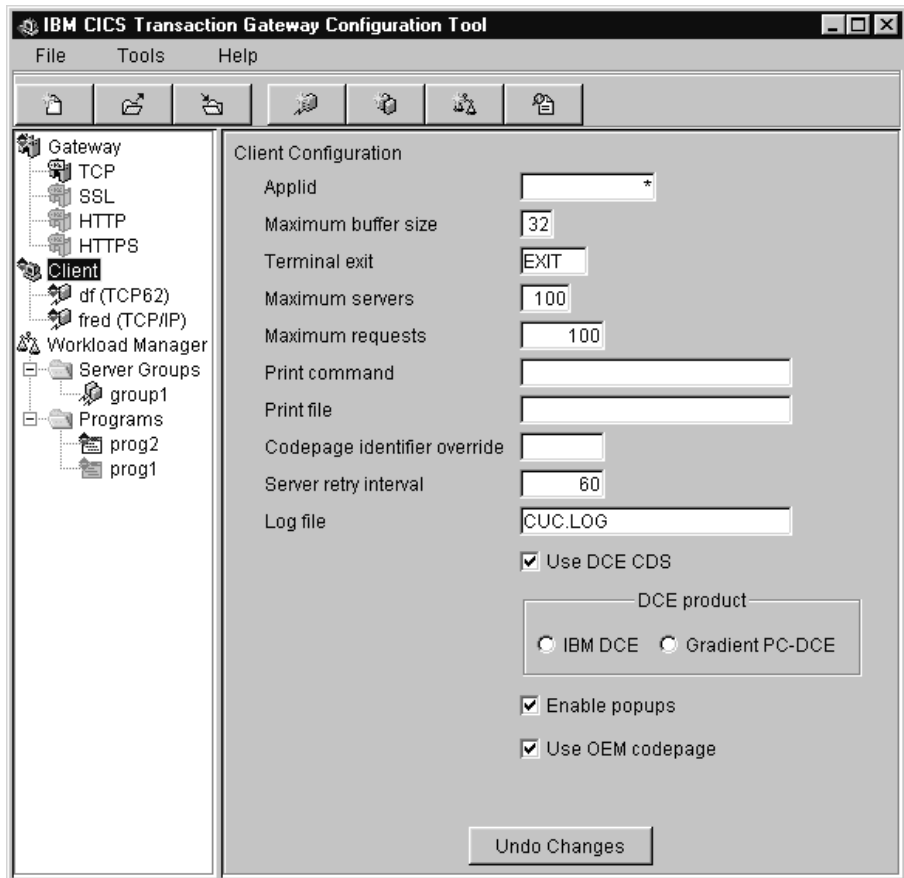


Figure 2. The configuration tool

Note: On your platform, the configuration tool may not appear exactly as shown in Figure 2.

Tree structure

The tree structure (see Figure 2) 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:

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

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

- | | |
|-----------------------|---|
| Trace Settings | Displays the Trace Settings dialog. |
| New Server | Starts the TaskGuide for creating new server definitions. |
| Delete Server | Deletes a server node from the tree structure. |

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

- | | |
|---------------------|--|
| Context Help | Displays online help information according to the current screen context, for example, for a particular configuration setting. |
| Contents | Displays the contents list for the online help information. |
| Index | 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.

Using the configuration tool

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.

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.

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.

It is the responsibility of the Print Command to delete the temporary print file after it has finished processing it.

Use a shell script with the print command (for example, **lpr**) followed by the command to delete (**rm**).

Using the configuration tool

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.

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 63 and “cicsprnt command reference” on page 68).

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 93.

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 /var/cicscli subdirectory.

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.

Using the configuration tool

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.

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, `cicssrv2.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.

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 directory should be used to locate the port number for the service CICS using a protocol of TCP.

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.

LU names are always aliases on AIX and Solaris, so this setting is ignored for these clients.

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

This must be an eight-character alias name.

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.

This must be an eight-character alias name.

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.

Trace settings

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

Using the configuration tool

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.

Trace everything	All components.
Client API level 1	The client API layer (level 1).
Client API level 2	The client API layer (level 1 and 2).
CICSCLI command line	The cicscli command interface.
CICSTELD	cicsteld process.
CICSTERM and CICSPRINT	cicsterm and cicsprnt emulators.
CPP classes	The C++ class libraries.
Client daemon	CICS Universal Client daemon.
Transport layer	Interprocess communication.
Protocol drivers	Protocol drivers (TCP, SNA, and so on).
Java Gateway	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 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 CICSCCLI.BIN file. This is in the /var/cicscli 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 80.

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:

CICSCCLI.INI	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 38.

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 /opt/cicscli/bin directory is assumed. The extension must be .INI.

Using the configuration tool

- 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 /opt/cicscli/bin/CICSCLI.INI is assumed.

-o=file

Specifies the pathname of the converted file. The default is /opt/cicscli/bin/CTG.INI.

To get help on using ctgconv, enter: ctgconv -?

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 = 256        # Limit the maximum server interaction
           MaxBufferSize = 32       # Allow for a 32K maximum COMMAREA
#           CCSID=850

ENDSECTION
```

SERVER section

```
SECTION SERVER = cicssnal
#-----
# 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
           LocalLUName = WXYZ9999     # The client's local LU name alias
           ModeName = LU62PS          # The SNA communications mode name

ENDSECTION
```

DRIVER section

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

           DriverName = CCLIBMSN      #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

Customizing the keyboard

The keyboard mapping for emulator operation is defined in a keyboard mapping file. The file supplied is CICSKEY.INI in the /opt/cicscli/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 64).
- The CICSKEY environment variable. For example:

```
set CICSKEY=/var/cicscli/MYKEYS.INI
```

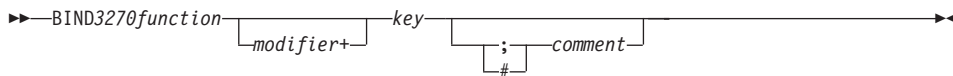
If this is not specified, a filename of CICSKEY.INI in the /opt/cicscli/bin directory is assumed.

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
cursor _{down}		pf4	pf16
cursor _{left}	printscreen	pf5	pf17
cursor _{right}	reset	pf6	pf18
cursor _{select}	tab	pf7	pf19
cursor _{up}		pf8	pf20
delete	ignore	pf9	pf21
enter		pf10	pf22
erase _{eof}		pf11	pf23
erase _{input}		pf12	pf24
home			
insert			
new _{line}			

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:

Ctrl
Shift

The *Key* can be any one of the keys shown in Table 6, but some combinations of modifier+key are not supported:

Table 6. 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

Customizing the keyboard

Note: For CICS Universal Client for Solaris, the Ctrl/Act, Print Screen, Scroll Lock and Pause keys are not available. The 3270 function Clear is assigned to the Esc key by default.

Key combinations

The following combinations of modifier and key can be mapped:

No modifier	All keys available for mapping.
Ctrl modifier	Only function keys, movement keys, alphabetic keys, tab key, and keypad keys can be mapped.
Shift modifier	Only function keys, numeric keys, tab key, and alphabetic keys can be mapped.

A sample key mapping file is shown in Figure 3.

```
;*****  
;* 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.
```

Figure 3. Sample keyboard mapping file for CICS Universal Clients (Part 1 of 3)

```
; Miscellaneous keys
bind Enter      Newline
bind Enter      KeypadEnter
bind Enter      Ctrl+KeypadEnter
bind Clear      Escape
bind Reset      Ctrl+R
bind Newline    Ctrl+N
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+E
bind EraseInput Ctrl+A
bind Ignore     End
bind PrintScreen Ctrl+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 Ctrl+F3
bind Home       Home
bind Home       Keypad7
```

Figure 3. Sample keyboard mapping file for CICS Universal Clients (Part 2 of 3)

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      Ctrl+Insert
bind Pa2      Ctrl+Home
bind Pa3      Ctrl+PageUp
```

Figure 3. Sample keyboard mapping file for CICS Universal Clients (Part 3 of 3)

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 `/opt/cicscli/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 63).
- The `CICSCOL` environment variable: For example:

```
set CICSCOL=/var/cicscli/MYCOLS.INI
```

If this is not specified, a filename of `CICSCOL.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:

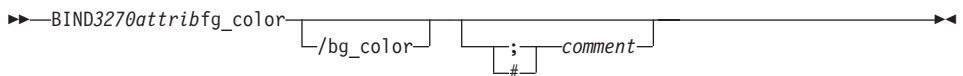
1. If the color mapping file specifies a mapping for an attribute, this mapping is used even if the hardware upon which the client is running actually supports the screen attribute.
2. If an application requests a 3270 field to be displayed with, for example, underscore, and no emulation setting has 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



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 4 on page 47.

Customizing the screen colors

```
*****
;* 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 4. 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 4. 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.
2. On the client workstation, enter the command:

```
cicscli -s=servername [-u=userid -p=password]
```

where you may have to include the *userid* and *password* if security is required. If you omit the password, a prompt is displayed, and you can enter the password without it being displayed.

3. After a few moments, enter the command `cicscli -l`. This displays information on the status of the servers to which the client is connected or attempting connection. Verify that the server is available.
4. Run `cicsterm` to start a terminal session and enter a transaction, for example, `CEMT INQUIRE TERMINAL`.

The window should be 80 columns by 25 rows (to allow for the status line provided with this CICS terminal emulator).

To exit from `cicsterm` use the command `EXIT`.

5. Follow the instructions in the `samplea.txt` file in the `samples` subdirectory on how to compile and run the sample programs.

Chapter 6. Operation

This chapter describes

- “An introduction to the CICS Universal Clients commands”
- “The cicscli command” on page 50
- “The cicsterm command” on page 61
- “The cicsprnt command” on page 66
- “CICS Client telnet terminal support” on page 70

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.

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 or an EPI or ECI application 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 57.

The cicscli command

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 console messages (-e parameter)
- Disable the display of console 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 57.

Note: With the cicscli command, you can enter parameters preceded by the minus (-) or by the slash (/) character.

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.

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.

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.

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.

Stopping the client process

If the client will not terminate completely, this may be because the client process **cclclnt** remains active. To stop this process, enter the command

```
kill -2 pid
```

where *pid* is the numeric process id of **cclclnt**.

Note: Do not use the `kill -9` command as this stops a process without allowing its resources to be released (these resources will remain until the system is restarted).

Restarting the client

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

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.

You can only use this parameter if you log in as root.

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 CICSCLL.BIN file in the /var/cicscli 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 57.

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 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.

The cicscli command

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. (You are prompted for the password.)

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.

You can also specify the security parameters when starting the client:

```
cicscli -s=servername -u=userid -p=password
```

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*.

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.

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, 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.

Password expiry management

For CICS clients the management of expired passwords can be handled by the ESI functions **CICS_ChangePassword** and **CICS_VerifyPassword**.

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 AIX, and CICS Universal Client for Solaris communicating over 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 'CICSAIX' (using 'TCP/IP' to 'CICSAIX') is available
```

The cicscli command

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 console messages

Client error messages (including those not associated with the client control process) are sent to the system console (/dev/console) by default, and also to the log file /var/cicscli/CICSCLI.LOG.

There may be times when you do not want messages appear on the system console.

To disable the display of console messages, enter:

```
cicscli -n
```

To enable the display of console messages again, enter:

```
cicscli -e
```

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

The display of console messages is enabled by default.

Displaying the command parameters

To display the parameters of the cicscli command, enter:

```
cicscli -?
```

cicscli and applications

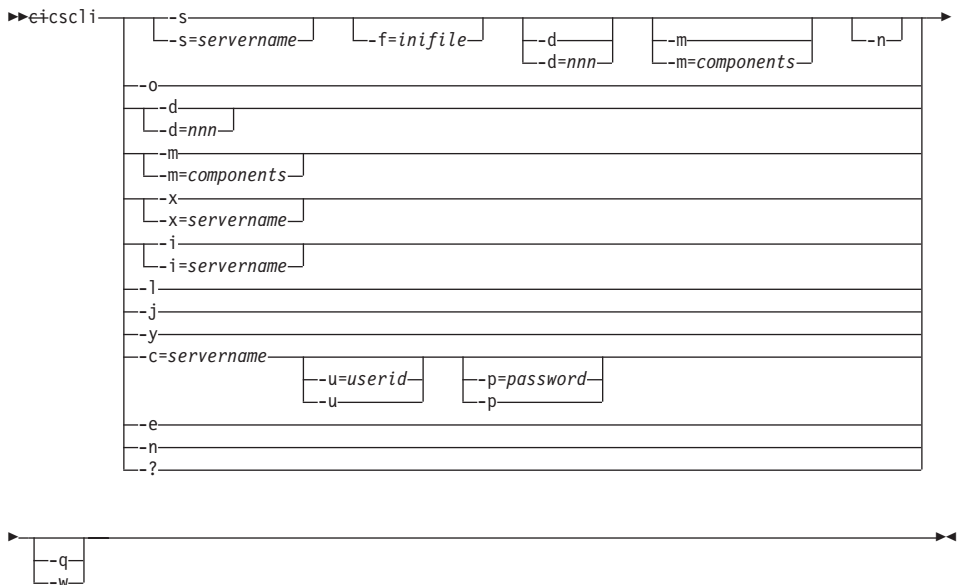
You can call cicscli from within applications, as long as this is supported by the programming language. You can therefore enter cicscli at the command line, then run an application to call cicscli with additional parameters.

cicscli command reference

All client control commands have options identified by a leading minus (-). Instead of minus signs you can use a slash for all parameters. However for the ? parameter you must use a minus sign. 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 x.

cicscli 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).

The cicscli command

-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 /var/cicscli directory, which can then be formatted with the cicsfrtc 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 82.

-e Enables the display of client error and security messages on the console.

-f=*inifile*

Specifies the name of the configuration file. The default is CTG.INI. If the file is not in the /opt/cicscli/bin directory, you must specify the full path name.

You can only use this parameter if you log in as root.

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.

-n Disables the display of client error and security messages on the console

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
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 26). 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.

The cicscli command

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.

-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. 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 cicscli 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 cicscli 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. This userid is used if the server requires a userid (and password) before running transactions on the client's behalf.

If you do not provide the -p parameter, you are prompted for the password.

For ECI applications, any userid and password specified in the ECI parameter block overrides values set via the cicscli 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.

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. `cicscli -y` is equivalent to `cicscli -x` 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.

`-?` 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.

For CICS Universal Client for AIX and CICS Universal Client for Solaris, 3270 field outlining is not supported for `cicsterm`.

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`

Note: With the `cicsterm` command, you can enter parameters preceded by the minus (`-`) or by the slash (`/`) character. However for the `?` parameter you must use a minus sign.

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=CICS0S2 -t=CESN -k=mykeys.ini -c=mycols.ini  
-n=cicsv123 -f=clprint.txt -q
```

In this example:

The cicsterm command

-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. Full details of the parameters are given in “cicsterm command reference” on page 63.

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.

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.

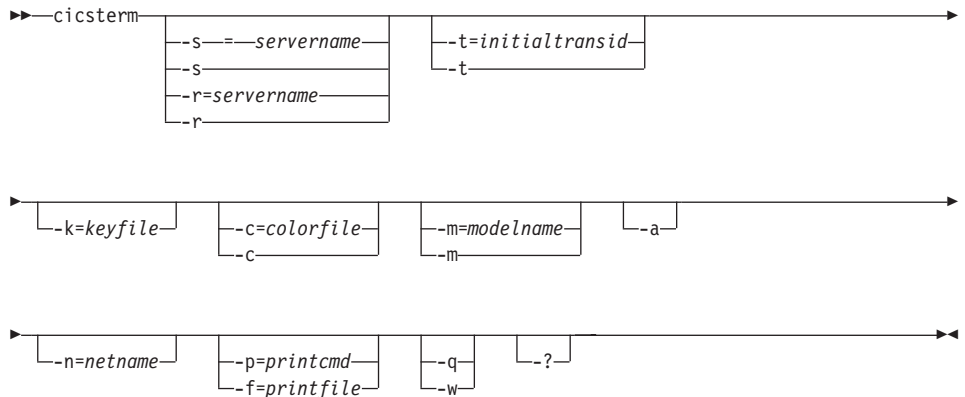
This behavior with the RETURN TRANSID(*name*) IMMEDIATE command was not supported for cicsterm before CICS Universal Clients Version 3.1, and is not supported for cicspnt. (It is however, supported by the EPI).

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 x.

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 44) 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 /opt/cicscli/bin directory is assumed.

The cicsterm command

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 40) 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 `/opt/cicscli/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 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.

The cicsterm command

- ? 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.

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.

Using cicsprnt

Note: With the cicsprnt command, you can enter parameters preceded by the minus (-) or by the slash (/) character. However for the ? parameter you must use a minus sign.

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=CICS0S2 -n=P123 -t=XPRT -f=c\print.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 68.

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.

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)
```

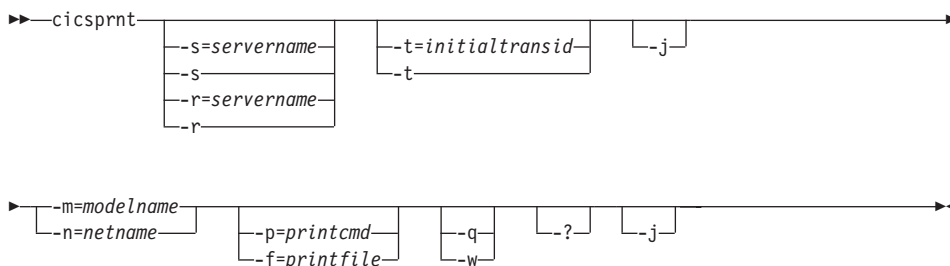
For more information, refer to “cicsterm and RETURN TRANSID IMMEDIATE” on page 62.

The cicsprnt command

cicsprnt command reference

For an explanation of syntax diagrams, see “Conventions and terminology used in this book” on page x.

cicsprnt Command Syntax



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

The cicsprnt command

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.

-? 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] [-b=file]
```

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 IBM-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 8859-1 (ASCII US English).

-t=tranid

Specifies an initial transaction.

-m=model

Specifies a model terminal name on the CICS server.

-b=file

Indicates that `cicsteld` is to run as a background task. All output is redirected to the specified file. If no file is specified then the output is written to `/var/cicscli/CICSTELD.OUT`. `cicsteld` stops when the number of users returns to zero.

-? 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.

Security considerations for trace and log files

CICS Universal Clients restricts access to the client trace and log files, which by default are normal files (not links) with names of `CICSCLI.BIN`, and `CICSCLI.LOG` in the `/var/cicscli` directory (you can specify the names of these files using the configuration tool).

The file permissions on `CICSCLI.BIN` are set so that only the owner (`root`) and group can write to the file, and only the owner can read it. Note however, that if a user has write access to the `/var/cicscli` directory they can delete the `CICSCLI.BIN` file regardless of the permissions on `CICSCLI.BIN`.

If you do not want unauthorised users to have access to the `CICSCLI.BIN` file, do not give them write access to the `/var/cicscli` directory. For example, enter a command such as:

Security considerations

```
chmod 755 /var/cicscli
```

will allow users to see files in /var/cicscli but not create, delete, or move them. The CICS Universal Clients ensure that if an unauthorized user has deleted and recreated CICSCLI.BIN, then tracing is disabled.

The file permissions on CICSCLI.LOG are set so that the owner (root) and group can read and write the file.

To improve security further:

- You can set the permissions on the /var/cicscli directory to restrict general access (chmod 0711 /var/cicscli). This means users cannot even see what files are in this directory.
- You can allow ECI, EPI programs, and terminals to start the client whilst restricting all other client administration (for example stopping the client, turning tracing on) to root:

```
chmod 0700 /opt/cicscli/bin/cicscli
```

Chapter 7. Problem determination

This chapter discusses problem determination

- “Preliminary checks and information”
- “Program support” on page 74
- “CICS Universal Client problem determination” on page 80
- “CICS server problem determination” on page 88
- “Communication problem determination” on page 88
- “Troubleshooting” on page 89

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 7 on page 77. 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 76 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[®] 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 80. 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 76 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 79. 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 80.

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 7 on page 77), 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 7. Sample problem reporting sheet

PROBLEM REPORTING SHEET		
Date	Severity	Problem Number
Company name		Contact name
Address		
Telephone		
System configuration—CICS Client		
CICS Client Rel	COBOL compiler level	Operating System
Client service level	C compiler level	Communications prod.
Fixes applied?	PL/I compiler level	
System configuration—CICS Server		
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	
Problem/Enquiry		
Abend/Trap	Incorrount	
Wait	Message	
Loop	Other	
Performance		
Documentation available—CICS Client		
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		
Documentation available—CICS Server		
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		
Actions		
Date	Name	Activity
Resolution		
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).

- 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.

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.

Program Support

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 87

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 57), 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 /var/cicscli 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):

CICSCLI.BIN	The binary trace file produced by running the client trace.
CICSCLI.WRP	The second binary trace file if wrapping of client trace is enabled.
CICSCLI.TRC	The name of the text trace file produced when the binary trace file is converted to a text file using the cicsftrc utility.
CICSCLI.BAK	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 82 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, 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
```

CICS Universal Clients trace

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.

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.

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.

CICS Universal Clients trace

Sample CICS Universal Clients trace: Figure 5 on page 85 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

CICS Universal Clients trace

```

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 000B9A0280502FF ...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 5. 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 81).
- 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 6 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 6. 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 /opt/cicscli/bin subdirectory. You can view these using any standard text editor:

CCLMSG.HLP

Help for the end user messages, in the language you chose at installation time.

CCLLOG.HLP

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.

Console messages

Errors generated from within the client process are displayed on the system console. There may be times when you do not want messages (client error and security) to appear on the console; for example, if you leave a CICS Universal Client running unattended overnight. To disable the display of console messages, enter the following command:

```
cicscli -n
```

When the display of console messages is disabled, the messages are still written to the client error log. To re-enable the display of console 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 console messages is enabled by default.

Messages from the client process

Messages from the client process are sent to the system console (/dev/console).

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 xii). 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:

CCL CICS Universal Clients and CICS Transaction Gateway
FAA CICS for OS/2
DFH CICS on System/390
ERZ Transaction Server for Windows NT Version 4.0, and TXSeries
AEG 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.

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”
- “TCP/IP communication problems” on page 90
- “APPC communication problems” on page 91.

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:

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.

CICS Universal Client for Solaris does not start

If you have problems starting or running the client, you may need to change your system specification file, which can be found in the `/etc/system` file. Errors might result because of incorrect values in the following line in this file:

```
set semsys:seminfo_semgni = 4096
```

Troubleshooting

This line sets the maximum number of semaphores that your system can hold at any one time. If this value is too low, the client cannot create enough semaphores to run properly, and may fail to start altogether. The following two lines affect the maximum size of messages queues that the system can support. If these values are too low, there may be a limit on the size of the commarea that you can flow to CICS.

```
set msgsys:msginfo_msgmax = 65535
set msgsys:msginfo_msgmnb = 65535
```

The following two lines are also related to the maximum size of message queues. These values, when multiplied together, must allow for the largest message in the system.

```
set msgsys:msginfo_msgssz = 32
set msgsys:msginfo_msgseg = 16384
```

The following line determines the maximum number of queue entries that can be in the system at anyone time. If this value is too low, there is a danger that the client will freeze while it waits for a queue entry to become available.

```
set msgsys:msginfo_msgtql = 10000
```

Whenever you change the `/etc/system` file, you must restart your machine for the changes to take effect.

Note: These changes will not work if your workstation has less than 32MB of memory, and 64MB is recommended.

TCP/IP communication problems

The following are solutions to problems that can occur when communicating over TCP/IP:

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.
- 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 89 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 8. 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 8.

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
		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	
	01363	01126	MS Windows: Korean SBCS
		01362	MS Windows: Korean DBCS
EBCDIC	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	
	00961	G2: CNS 11643 plane 2	
EBCDIC	00937	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 98
- “CICS Family publications” on page 98
- “Book filenames” on page 99
- “Sample configuration documents” on page 99
- “Other publications” on page 100
- “Viewing the online documentation” on page 100

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 8 show the softcopy filenames of the CICS Transaction Gateway and CICS Universal Client books.

Table 8. 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
Note: 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.

Viewing the online documentation

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 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.

cicsterm. A command that provides 3270 emulation and enables connection to a CICS region.

Change-Number-of-Sessions (CNOS). An internal transaction program that regulates the number of parallel sessions between the partner LUs with specific characteristics.

Glossary

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 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.

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.

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 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.

Glossary

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*.

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

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.

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*.

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.

P

padding. 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 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 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.

Glossary

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.

T

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.

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.

Glossary

Index

A

- advanced program-to-program communication (APPC) 10, 22
- APAR (authorized program analysis report)
 - authorization 79
 - closing 80
 - documentation needed 76
 - process 79
 - submitting 79
- APPC (advanced program-to-program communication) 10, 22
- Applid configuration setting 30

B

- benefits of CICS Universal Clients 2
- binary trace formatter 82
- books 97
 - CICS Transaction Gateway and CICS Universal Clients library 97
 - online 100
 - PDF 101
 - printed 101
- business logic 3

C

- CCLLOG.HLP 87
- CCLMSG.HLP 87
- CICS server problem determination 88
- CICS server PTF requirements 12
- CICS servers 1
- CICS servers, connections 7
- CICS telnet daemon 70
- CICS Universal Client
 - trace analysis 83
- CICS Universal Client for Solaris
 - installing 15
 - national language support 15
- CICSCLI.BIN 80
- cicscli command 50
- CICSCLLOG 33, 87
- CICSCLITRC 80
- CICSCOL environment variable 44
- cicsftrc utility 82
- CICSKEY environment variable 40
- cicsprnt command 66

- cicsteld command 70
- cicsterm command 61
- client control 5
- client control process, restarting 51
- client control process, starting 50
- client control process, stopping 51
- client initialization file
 - specifying which to use 52
- client/server communication, setting up 19
- client/server connections 8
- client trace 52, 81
- client trace file 83
- Client trace file configuration setting 36
- Codepage identifier override configuration setting 32
- color mapping file 44, 61
- commands
 - cicscli 50
 - cicsprnt 66
 - cicsterm 61
- COMMAREA 3
- communication
 - problems 88
 - setting up a client 19
- communication, setting up client/server connection 19
- communication protocols
 - APPC 10
 - NetBIOS 10
 - TCP/IP 10
- configuration
 - configuration file 26
 - configuration tool 26
- configuration conversion tool 37
- configuration file 26
- configuration settings
 - Applid 30
 - Client trace file 36
 - Codepage identifier override 32
 - Connection timeout 35
 - Description 33
 - Hostname or IP address 34
 - Initial transaction 33
 - Local LU name 35
 - Log file 33
 - Maximum buffer size 30
 - Maximum Client wrap size 37

- configuration settings (*continued*)
 - Maximum requests 31
 - Maximum servers 31
 - Mode name 35
 - Model terminal definition 33
 - Partner LU name 35
 - Port 34
 - Print command 31
 - Print file 32
 - Send TCP/IP Keepalive packets 35
 - Server name 33
 - Server retry interval 32
 - SNA 34
 - TCP/IP 34
 - Terminal exit 30
 - Trace 36
 - Trace settings 80
 - Use LU alias names 35
 - Use upper case security 34
- configuration tool 26
- connecting to CICS servers 50, 57
- Connection timeout configuration setting 35
- console messages, disabling the display of 56
- console messages, enabling the display of 56
- Corrective Service Diskette (CSD) 80
- CRSR transaction 22
- CSD (Corrective Service Diskette) 80
- ctgconv, conversion tool 37
- customizing
 - keyboard 40
 - screen colors 44

D

- DBCS (double-byte character set) 12
- defining 3270 printer terminal emulator characteristics 66
- defining 3270 terminal emulator characteristics 61
- Description configuration setting 33
- disabling the display of console messages 56
- disabling the display of messages 56

- DISPLAY environment variable 16
- displaying command syntax 56
- documentation 97
 - HTML 100
 - PDF 101
- documentation of problems 76
- double-byte character set (DBCS) 12
- E**
- ECI (External Call Interface) 3
- enabling the display of console messages 56
- environment variables
 - CICSCOL 44
 - CICSKEY 40
- EPI (External Presentation Interface) 4
- error log
 - CICSCLLLOG 87
 - client error log 87
 - server error log 88
- error processing 87
- ESI (External Security Interface) 4
- Euro support 32
- example list of CICS servers 55
- EXEC CICS RETURN TRANSID IMMEDIATE command 62, 67
- External Call Interface (ECI) 3
- External Presentation Interface (EPI) 4
- External Security Interface (ESI) 4
- F**
- free memory 30
- G**
- glossary of terms and abbreviations 107
- graphical user interface (GUI) 4
- GUI (graphical user interface) 4
- H**
- hardcopy books 101
- hardware requirements 11
- Hostname or IP address
 - configuration setting 34
- HTML (HyperText Markup Language) 100
- HTML documentation, viewing 100
- HyperText Markup Language (HTML) 100
- I**
- initial transaction 61
- Initial transaction configuration setting 33
- initialization files
 - CICSCOL.INI 44
 - CICSKEY.INI 40
- IP address, TCP/IP 19
- K**
- keyboard mapping file 40, 61
- L**
- language support for client applications 12
- listing connected servers 55
- Local LU name configuration setting 35
- Log file configuration setting 33
- logical unit (LU) 22
- LU6.2 10
- M**
- Maximum buffer size configuration setting 30
- Maximum Client wrap size configuration setting 37
- Maximum requests configuration setting 31
- Maximum servers configuration setting 31
- memory requirements 30
- messages, disabling the display of 56
- mode definitions, APPC 22
- Mode name configuration setting 35
- Model terminal definition configuration setting 33
- O**
- object oriented programming 2
- online books, PDF 101
- online documentation, HTML 100
- online help, for end user messages, 87
- online help, for trace and log messages 87
- options
 - cicscli command 57
 - cicsprnt command 68
 - cicsterm command 63
- P**
- Partner LU name configuration setting 35
- password expiry management (PEM) 55
- PDF (Portable Document Format) 101
- PDF books, viewing 101
- PEM (password expiry management) 55
- planning 7
- PMR (Problem Management Record) 74
- Port configuration setting 34
- port number, TCP/IP 19
- Portable Document Format (PDF) 101
- PostScript books 101
- presentation logic 3
- Print command configuration setting 31
- print file, processing 61, 66
- Print file configuration setting 32
- print terminal emulator, starting 66
- printer support 4
- printer terminal emulator
 - characteristics, defining 66
- problem determination 73
- Problem Management Record (PMR) 74
- problem reporting
 - documentation of problems 76
 - information needed 77
 - report sheet 77
 - support organization 74
- problems, common 89
- problems, communication 88
- program support 74
- programming languages 12
- PTF (Program Temporary Fix) 80
- publications, CICS Transaction Gateway and CICS Universal Clients library 97
- R**
- railroad diagrams x
- requirements, hardware 11
- requirements, software 12
- restarting the client 51
- restrictions, using CICS servers 9
- RETAIN database 74
- RETAIN problem management system
 - APARs 79
 - Problem Management Record 74
- S**
- security 34, 53
- Send TCP/IP Keepalive packets
 - configuration setting 35
- Server name configuration setting 33

- Server retry interval configuration
 - setting 32
- servers, listing 55
- SERVICES file, TCP/IP 19
- setting up client/server
 - communication 19
- signon capable terminals 12
- SNA (Systems Network
 - Architecture) 10
- SNA configuration setting 34
- softcopy books, PDF 101
- software requirements 12
- starting a 3270 print terminal
 - emulator 66
- starting a 3270 terminal
 - emulator 61
- starting the client control
 - process 50
- stopping a terminal emulator 62
- stopping the client control
 - process 51
- support organization
 - problem reporting 77
 - program support 74
 - sending documentation to 79
- syntax notation x
- Systems Network Architecture
 - (SNA) 10

T

- TCP/IP (Transmission Control
 - Protocol/Internet Protocol) 10, 19
- TCP/IP configuration setting 34
- telnet support 70
- terminal emulation 3
- terminal emulator, starting 61
- terminal emulator, stopping 62
- terminal emulator characteristics,
 - defining 61
- Terminal exit configuration
 - setting 30
- testing the client/server link 48
- Trace configuration setting 36
- trace file 80
- Trace settings, configuration tool 80
- tracing IBM CICS Universal
 - Clients 80
- transaction programs (TP),
 - APPC 22
- transmission control
 - protocol/internet protocol
 - (TCP/IP) 10
- Transmission Control
 - Protocol/Internet Protocol
 - (TCP/IP) 10, 19
- troubleshooting 89

U

- Use LU alias names configuration
 - setting 35
- Use upper case security
 - configuration setting 34

V

- viewing online documentation 100



Program Number: 5648-B42



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

SC34-5451-01

