CICS® Universal Client for Windows®

Administration

Version 3.1

SC34-5449-01

CICS® Universal Client for Windows®

Administration

Version 3.1

SC34-5449-01

Note!

Before using this information and the product it supports, be sure to read the general information under "Appendix C. Notices" on page 173.

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 SC34-5449-00.

© 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 vii

Tables ix

Summary of changes xi

Changes for IBM CICS Universal Clients Version 3.1 xi

About this book xiii

Who should read this book. xiii

Conventions and terminology used in this book xiv

Prerequisite and related information. . . . xvi

CICS Universal Clients publications . . . xvi

CICS publications xvi

Microsoft® Windows Family publications xvii

NetBIOS-related publications xvii

APPC-related publications. xvii

How to send your comments xviii

Obtaining books from IBM xix

Chapter 1. Overview 1

An introduction to IBM CICS Universal Clients 1

CICS Universal Clients communication . . . 2

CICS Universal Clients and DCE 3

The benefits of using CICS Universal Clients . 3

CICS Universal Clients functions 4

3270 terminal emulation 4

The External Call Interface 4

The External Presentation Interface. 5

The External Security Interface 5

3270 client printer support 5

Client control 6

CICS telnet support 6

Chapter 2. Planning before installation . . 7

CICS Universal Clients and your network . . 7

Connections to CICS servers 7

Accessing the CICS Universal Clients . . . 8

Multiple protocol configurations 9

Mobile communication. 9

Communication between CICS Universal Clients and CICS servers. 11

Restrictions on the use of CICS servers . . 12

Communication protocols 13

Which protocols can you use? 13

Why use a particular protocol?. 14

Hardware requirements 15

Software requirements 17

CICS server PTF requirements 20

Code page support. 20

Chapter 3. Installing CICS Universal Clients 21

Installing CICS Universal Clients 21

Installing CICS Universal Client on Windows platforms 22

Types of installation 22

Installing CICS Universal Client 22

The system PATH and network drives . . 25

Uninstalling CICS Universal Client . . . 25

Problems during installing or uninstalling 26

Updating CICS Universal Client 26

Silent installation 27

Registering COM libraries 29

Registering in-process servers 30

Registering out-of-process servers. . . . 30

Other actions required in setting up a CICS Universal Client. 30

Chapter 4. Setting up client/server communication 31

Client/server definitions. 31

Definition checklists 32

NetBIOS configuration 33

Definition checklists 33

NetBIOS resources 34

TCP/IP configuration 34

Definition checklists 34

Verifying the TCP/IP installation 36

TCP62 configuration 37

On OS/390 38

Firewall implications 39

APPC configuration 40

Configuring APPC for CICS Universal Client for Windows 40

Named Pipes configuration 46

Chapter 5. Customization 47

© Copyright IBM Corp. 1994, 1999

Renaming the client configuration and mapping files 47
Specifying which Java Virtual Machine to use 48
Using the configuration tool 48
The configuration tool interface 49
Configuring Client settings 52
Configuring Server settings 56
Configuring the Workload Manager . . . 64
Trace settings 67
The configuration conversion tool. 69
Using the conversion tool 69
Editing the configuration file 70
CLIENT section 71
SERVER section 71
DRIVER section 71
LOADMANAGER section 72
Configuration file properties 72
Client section 72
Server section 73
Workload Manager section 74
Customizing the keyboard 74
Keyboard mapping file syntax 74
The keyboard mapping file 75
Customizing the screen colors 78
Color mapping syntax 79
The color mapping file 80
Testing the client/server link 82

Chapter 6. Operation 83
An introduction to the CICS Universal Clients commands 83
CICS Universal Client for Windows 98 operation 84
CICS Universal Client for Windows NT operation 84
Running CICS Universal Client for Windows NT as a Windows NT service . . 84
CICS Universal Clients icons 87
Customizing the command icons 87
The CICSCLI command 88
Starting the client control process 88
Starting connections with additional servers 89
Stopping the client control process . . . 89
Restarting the client 90
Specifying which configuration file to use 90
Turning on client trace 90
Specifying the trace components 91
Turning off client trace 91
Setting up security 91
Listing the connected servers 95
Disabling the display of messages. . . . 95
Enabling and disabling the display of pop-up messages 95
Displaying the command parameters. . . 96
CICSCLI and applications 96
CICSCLI command reference 97
The CICSTERM command 101
Using CICSTERM. 101
Stopping a terminal emulator 102
CICSTERM and user exits 102
CICSTERM and RETURN TRANSID IMMEDIATE 103
CICSTERM command reference 103
The CICSPRNT command 106
Using CICSPRNT 107
CICSPRNT and user exits 108
CICSPRNT and RETURN TRANSID IMMEDIATE 108
CICSPRNT command reference 108
Font selection on terminal emulators . . . 111
CICS Client telnet terminal support 112

Chapter 7. CICS Universal Client support for DCE 115
CICS Universal Clients support for DCE . . 115
Functions provided by DCE 116
Enabling CDS support in CICS Universal Clients 116
Servers located via the CDS 116
List systems API calls 117
CICSTERM and CICSPRNT commands 118
Configuring CICS Universal Clients for DCE RPC communication 118
Coordinated security sign-on 118
Unauthenticated RPC 119
Authenticated RPC 119
Data encryption 121
Configuring DCE on the client workstation 121
Configuring IBM DCE 121
Configuring Gradient PC-DCE 122
Configuring the CICS server 122
Configuring DCE RPC communication— a summary 123
Troubleshooting 123

Chapter 8. Workload Manager 125
Workload Manager overview 125
Key concepts 126
What the Workload Manager needs to know 127

Load balancing algorithms 129
The round-robin algorithm. 129
The biasing algorithm 129
Failure recovery 130
Exit implementation 130
Workload Manager installation 131
Configuring the Workload Manager. . . . 132
Tracing the Workload Manager 133
Programming requirements 133

Chapter 9. Problem determination 135
Preliminary checks and information. . . . 135
Program support 136
Reporting problems 136
Documentation of problems 138
Locating and compiling information . . 139
Submitting the documentation 141
APARs and fixes 141
CICS Universal Client problem determination 142
CICS Universal Clients trace 142
CICS Universal Client error processing 149
CICS server problem determination. . . . 150
Communication problem determination . . 150
TCP/IP-providing products 151
APPC-providing products 151
Troubleshooting 152
Starting clients and terminals 152
NetBIOS communication problems . . . 153
TCP/IP communication problems . . . 154
APPC communication problems 155
DCE RPC communication problems. . . . 156

Chapter 10. Migration issues 159
Application compatibility 159
ECI and EPI compatibility 159

Appendix A. CICS Universal Clients data conversion 161
CICS Universal Client for Windows NT data conversion 161
Supported conversions 162

Appendix B. The CICS Transaction Gateway and CICS Universal Clients library 167
CICS Transaction Gateway books 167
CICS Universal Clients books 168
CICS Family publications 168
Book filenames. 169
Sample configuration documents. 169
Other publications 170
Viewing the online documentation 170
Viewing PDF books 171

Appendix C. Notices 173
Trademarks 174

Glossary 177

Index 185

Figures

1. CICS Universal Clients communicating over NetBIOS, TCP/IP, APPC 9
2. Mobile communication to a CICS server 10
3. Example messages returned by ping command 37
4. The configuration tool 49
5. Sample keyboard mapping file for CICS Universal Clients. 77
6. Sample color mapping file 81
7. Sample CICS Universal Clients trace 147
8. CICS Universal Clients trace: using an invalid port number 148

© Copyright IBM Corp. 1994, 1999

Tables

1. Command syntax conventions xiv
2. Communication between CICS Universal Client for Windows NT and CICS servers and functions 11
3. Hardware requirements for CICS Universal Client for Windows 98 15
4. Hardware requirements for CICS Universal Client for Windows NT 15
5. Software requirements for CICS Universal Client for Windows 98 17
6. Software requirements for CICS Universal Client for Windows NT 19
7. Matching definitions: CICS Universal Clients to CICS for OS/2 using NetBIOS . 32
8. Matching definitions: CICS Universal Clients to CICS for OS/2 using NetBIOS . 33
9. Matching definitions: CICS Universal Clients to TXseries for Windows NT . . 36
10. CICS Universal Client keys that can be mapped. 75
11. Sample problem reporting sheet 139
12. CICS Transaction Gateway and CICS Universal Clients books and file names . 169

© Copyright IBM Corp. 1994, 1999

Summary of changes

Changes for IBM CICS Universal Clients Version 3.1

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

- A graphical user interface (GUI) utility that makes configuration of CICS Universal Clients easier.
- A conversion utility, CTGCONV to convert configuration files from existing CICS Universal Clients to the new format of the configuration file.
- Enhanced security functions.
- Improved trace facilities including a utility, CICSFTRC, that converts the binary trace file to ASCII format.
- Improvements to the Workload Manager of CICS Universal Client for Windows.
- CICS Universal Client for Windows NT can now communicate with TXSeries for Windows NT (Version 4.3 or later) servers that are on the same computer as the CICS Client, using the Local Named Pipe protocol.

© Copyright IBM Corp. 1994, 1999

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 the support for Distributed Computing Environment (DCE) in CICS Universal Clients.
- Chapter 8 describes how to use the Workload Manager to enhance the performance of CICS Universal Client for Windows 98 and CICS Universal Client for Windows NT.
- Chapter 9 describes problem determination for CICS Universal Clients. This chapter describes CICS Universal Clients trace and error processing, and discusses communication problems and how to deal with them.
- Chapter 10 discusses some migration issues relevant to using CICS Universal Client applications.
- Appendix A discusses data conversion between CICS Universal Clients and CICS servers.
- Appendix B describes how to view the online information in the CICS Universal Clients library, and how to print and order books.

Who should read this book

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

© Copyright IBM Corp. 1994, 1999

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

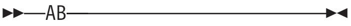
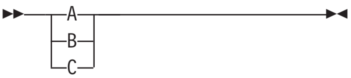
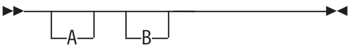
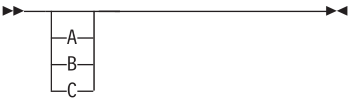
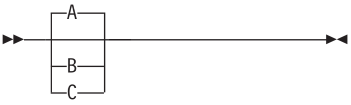
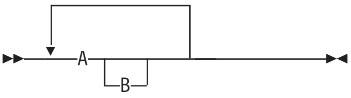
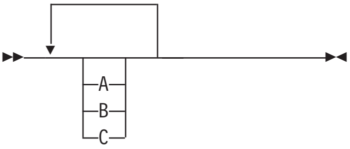
Symbol	**Meaning**
	Required items appear on the main path line.
	If there is more than one required item to choose from, the items are stacked vertically. This is a set of alternatives—one of which you ***must*** specify.
	Optional items appear below the main path line.
	If there is more than one optional 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
	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. **Note:** For CICS, unless otherwise stated, this representation means only that more than one item can be selected.
Punctuation and uppercase characters	Code exactly as shown.
Lowercase characters appearing like *this*	Code your own text, as appropriate. For example, with FILE(*name*) you must specify FILE and () unchanged, but are free to specify any valid text string for the name of your file.

A glossary is provided at the back of this book.

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

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

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

‘CICS Client for Windows Family’ refers to CICS Universal Client for Windows 98, and CICS Universal Client for Windows NT collectively. Similarly, ‘Windows Family’ refers to the Windows 95, Windows 98, and Windows NT® 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 167. 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/`

Microsoft® Windows Family publications

The following books provide introductory material about the Windows Family of operating systems:

- *Windows NT System Guide*
- *Introducing Microsoft Windows 95*

NetBIOS-related publications

The following books are concerned with products providing NetBIOS:

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

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

APPC-related publications

The following books are concerned with products providing APPC support:

IBM products

IBM eNetwork Communications Server for Windows NT:

IBM eNetwork Communications Server for Windows NT, Up and Running!, GC31-8424

IBM eNetwork Personal Communications:

IBM eNetwork Personal Communications Version 4.2 for Windows 95 and Windows NT, Quick Beginnings, GC31-8476

IBM eNetwork Personal Communications Version 4.2 for Windows 95 and Windows NT, Reference, SC31-8477

Novell products

Novell NetWare APPC Technical Reference

Novell NetWare for SAA Version 1.3 Administration Guide

NetWare LU6.2 for Windows Configuration Guide

Novell NetWare for SAA Version 2.0 Administration Guide for Mainframes and Peer Nodes

Novell NetWare for SAA Version 2.0 API Configuration Guide for Microsoft Windows Clients

Novell NetWare for SAA Version 2.2 Getting Started

Novell NetWare for SAA Version 2.2 Installation

Novell NetWare for SAA Version 2.2 Operation and Diagnosis

Novell IntranetWare for SAA Version 3.0 Planning and Installation

Novell IntranetWare for SAA Version 3.0 Operation and Diagnosis

Microsoft products

Microsoft SNA Server Installation Guide

Microsoft SNA Server Administration Guide

Microsoft SNA Server Reference

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.ibmlink.ibm.com/pbl/pbl`

Chapter 1. Overview

This chapter contains the following:

- "An introduction to IBM CICS Universal Clients"
- "CICS Universal Clients communication" on page 2
- "The benefits of using CICS Universal Clients" on page 3
- "CICS Universal Clients functions" on page 4.

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. With CICS Universal Client for Windows 98, you can also run 16-bit applications developed for Windows 3.1.

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 Tranasction 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

© Copyright IBM Corp. 1994, 1999

- CICS Transaction Server for OS/390 Release 1 and higher
- CICS Transaction Server for VSE/ESA Release 1
- CICS for VSE/ESA Version 2 Release 3, and higher

CICS Universal Clients communication

Your CICS Universal Clients can communicate via the following protocols:

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

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

CICS Universal Client for Windows NT can communicate with TXSeries for Windows NT (Version 4.3 or later) servers, which are on the same computer as the CICS Client, using the Local Named Pipe protocol.

Table 2 on page 11 shows the protocols that CICS Universal Clients can use to communicate with each server, and the functions supported. Support for a protocol may be provided by one or more communication software products. You can therefore use the products best suited to your network environment.

A variety of LAN environments are supported, for example:

- You can use Microsoft SNA Server, or IBM eNetwork Communications Server for APPC communication between CICS Client for Windows Family and CICS servers.
- You can use CICS Client for Windows Family in a Novell NetWare LAN, with Novell NetWare for SAA (running in a gateway workstation) providing APPC communication to a CICS server.

  For more information, see “Gateways” on page 8.

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 48).

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* and *CICS Universal Clients COM Automation Programming* books.

CICS Universal Clients and DCE

The Distributed Computing Environment (DCE) provides secure, authenticated client/server interoperability across diverse platforms. Of the CICS Universal Clients, DCE support is provided for CICS Universal Client for Windows NT. The DCE support makes locating servers easier, and provides a more secure environment for client/server communication.

For more information, refer to “Chapter 7. CICS Universal Client support for DCE” on page 115.

The benefits of using CICS Universal Clients

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

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

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

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

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 11) are autoinstalled at most CICS server systems, and do not have to be predefined at the server.

The External Call Interface

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

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

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

The External Presentation Interface

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

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

The External Security Interface

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

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

3270 client printer support

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

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

CICS 3270 client printer support uses CICS 3270 emulation functions. See Table 2 on page 11 for information on which CICS servers currently support CICS 3270 emulation and hence CICS 3270 client printer support.

Client control

CICS Universal Clients provide commands or icons to:

- **Control the client process**

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

  You can:
  - Start and stop the terminal emulator
  - Specify the initial transaction
  - Define the terminal characteristics
  - Specify the name of the keyboard and screen color mapping files
  - Define the command used to process print requests
  - Specify the name of a file used for appending print requests.
- **Control client printer operation**

  You can:
  - 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 112.

Chapter 2. Planning before installation

This chapter describes the following:

- "CICS Universal Clients and your network"
- "Communication protocols" on page 13
- "Hardware requirements" on page 15
- "Software requirements" on page 17

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

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

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

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

Your CICS Universal Clients can communicate with CICS application servers via APPC. This is commonly achieved through a LAN and a workstation acting as a ***SNA gateway***.

© Copyright IBM Corp. 1994, 1999

Gateways

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

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

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

SNA gateway software is required in the gateway workstation to provide gateway server functions to the clients on the LAN:

- ***In an IBM LAN environment,*** eNetwork Communications Server is used.
- ***In a Novell NetWare LAN,*** CICS Universal Client for Windows 98, and CICS Universal Client for Windows NT can communicate using APPC, with Novell NetWare for SAA in the gateway workstation.
- ***In a Microsoft SNA Server or IBM eNetwork Communications Server environment,*** the CICS Client for Windows Family can communicate over APPC. You can install SNA Server on a single Windows NT workstation or use Windows NT, or Windows 95/98 SNA Client programs with the SNA Server program. In this case, SNA Client is installed on each client workstation, and supports the APPC API, while SNA Server is installed on another workstation, and provides the SNA transport layer used to communicate with other systems.

  When used in this manner, a single SNA Server system can support multiple Windows Family SNA Client systems. Similarly, you can use eNetwork Communications Server on a single Windows NT workstation, or use eNetwork Communications Server clients on individual client machines.

Accessing the CICS Universal Clients

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

1. You can have CICS Universal Clients code on each of the workstations from which you want to use CICS.

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

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

Multiple protocol configurations

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

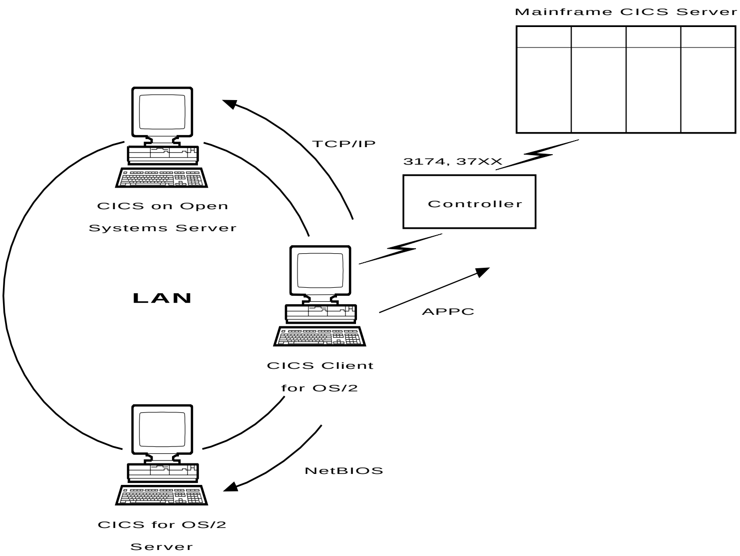


Figure 1. CICS Universal Clients communicating over NetBIOS, TCP/IP, APPC

Mobile communication

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

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

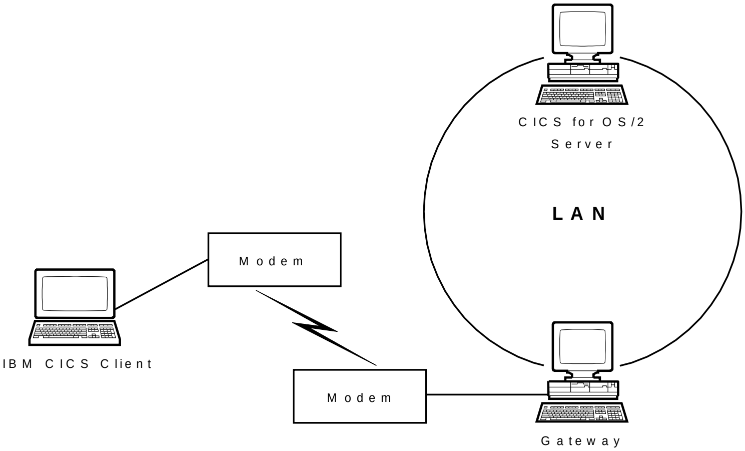


Figure 2. Mobile communication to a CICS server

Communication between CICS Universal Clients and CICS servers

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

Table 2. Communication between CICS Universal Client for Windows NT 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	● note 7	●	●	○	●	●	●
TXSeries Version 4.2 for Windows NT	○	●	●	○	●	●	●
‘CICS on open systems’	○	●	●	○	●	●	●
CICS for OS/400 Version 3.1	○	○	●	○	●	●	●
CICS for MVS/ESA Version 4.1	○	● note 3	●	●	●	● note 4	●
Transaction Server for OS/390 Version 1.x	○	● note 3	●	●	●	●	●
CICS for VSE/ESA Version 2.3	○	○	●	○	●	● note 6	●
CICS Transaction Server for VSE/ESA Version 1.1	○	○	●	○	●	●	●
Note: ● = supported, ○ = not supported.							

Notes on table:

1. EPI always incorporates CICS 3270 terminal emulation and CICS 3270 client printer support.

2. Autoinstall means that the user does not need to predefine the client to the CICS server, that is, control table definitions are automatically created for the client at the CICS server. For CICS for VSE/ESA autoinstall is possible via LU 6.2 single session only, although this restriction does not apply to CICS Transaction Server for VSE/ESA.
3. TCP/IP communication between CICS Universal Client for OS/2, CICS Universal Client for Windows NT, CICS Universal Client for Windows 98 and CICS for MVS/ESA Version 4.1 and later is achieved using TCP62 support.
4. EPI Emulation is supported on CICS for MVS/ESA Version 4.1 with APAR PN68409.
5. APAR PN79262 is required for DBCS support in CICS for MVS/ESA.
6. EPI emulation is supported on CICS for VSE/ESA Version 2.3 with APAR PN87445 (PTF numbers UN90168 and UN90169).
7. NetBios communication is not supported by CICS Universal Client for Windows 98.
8. TXSeries (Version 4.3 or later) for Windows NT can use the Local Named Pipe protocol to communicate with CICS Universal Client on the same computer as the server.

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:

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

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

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

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

Local Named Pipe
CICS Universal Client for Windows NT can communicate with TXSeries (Version 4.3 or later) for Windows NT servers which are on the same computer as the CICS Universal Client using a protocol based on Windows Named Pipes. This provides a fast communications method independent of any network protocol.

In addition to these protocols, CICS Universal Client for Windows NT can use DCE Remote Procedure Calls (RPC) for connection with a CICS server with the equivalent DCE functionality. For more information, see “Chapter 7. CICS Universal Client support for DCE” on page 115.

Communication protocols

The protocols that can be used for the various client/server connections are shown in Table 2 on page 11.

Why use a particular protocol?

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

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

With the TCP62 support provided for CICS Universal Client for OS/2, CICS Universal Client for Windows 98, and CICS Universal Client for Windows NT, the CICS Universal Clients can access CICS for MVS/ESA and Transaction Server for OS/390 using the TCP/IP network. This allows partner LU62 applications to communicate without complex SNA configuration definitions on the client, and without any changes to the LU62 applications on the client or server. Therefore, if you require easy client access to CICS for MVS/ESA over your TCP/IP network you should use the TCP62 support. For information on configuring TCP62, see “TCP62 configuration” on page 37.

If you are already using a particular protocol, you may wish to continue with that protocol alone. Note however, that you can only have one SNA stack on a system. The CICS Host TCP/IP Access Feature contains a SNA stack, so this cannot be used simultaneously with an existing SNA stack on the same system. See “Coexistence of TCP62 with SNA protocol” on page 39.

Hardware requirements

The following table lists the hardware requirements of CICS Universal Client for Windows 98.

Table 3. Hardware requirements for CICS Universal Client for Windows 98

Processor type
Any machine capable of running the appropriate operating system and other prerequisite software.
Network adapters
To connect workstations to the LAN, a suitable adapter such as one of the following is required: • The IBM Token-Ring Network Adapter/A • The IBM Token-Ring Network 16/4 Adapter To connect a workstation to a Novell NetWare LAN, a suitable Token-Ring or Ethernet network adapter is required.
Memory requirements
CICS Universal Client for Windows 98 will run in the minimum 8 MB of memory required by Windows 95 and Windows 98.
Disk space requirements
8MB For the TCP62 support for CICS Universal Client for Windows 98, you will require an additional 5.8 MB

The following table lists the hardware requirements of CICS Universal Client for Windows NT

Table 4. Hardware requirements for CICS Universal Client for Windows NT

Processor type
Any machine capable of running the appropriate operating system and other prerequisite software.
Network adapters
To connect workstations to the LAN, a suitable adapter such as one of the following is required: • The IBM Token-Ring Network Adapter/A • The IBM Token-Ring Network 16/4 Adapter To connect a workstation to a Novell NetWare LAN, a suitable Token-Ring or Ethernet network adapter is required.
Memory requirements

Hardware requirements

Table 4. Hardware requirements for CICS Universal Client for Windows NT (continued)

CICS Universal Client for Windows NT will run in the minimum 16 MB of memory required by Windows NT.
Disk space requirements
8MB For the TCP62 support for CICS Universal Client for Windows NT you will require an additional 5.8 MB.

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 Windows 98.

Table 5. Software requirements for CICS Universal Client for Windows 98

Operating system version
CICS Universal Client for Windows 98 can run under any of the following operating systems: • Windows 95, with Euro Product Update and Year 2000 Update • Windows 98, with Year 2000 Update
NetBIOS support
NetBIOS support is not provided for CICS Universal Client for Windows 98
TCP/IP support
TCP/IP support is provided by Windows 95 and Windows 98, however any vendor TCP/IP implementation that adheres to the Windows Socket API (WINSOCK); for example, Microsoft TCP/IP Winsock can be used.
SNA support

Software requirements

Table 5. Software requirements for CICS Universal Client for Windows 98 (continued)

APPC support is provided by: • Microsoft SNA Server Version 3.0, or later SNA Server Client must be installed on the CICS Universal Client for Windows 98 workstation. • IBM eNetwork Communications Server for Windows NT Version 5.0 or later eNetwork Communications Server Client for Windows 95 must be installed on the CICS Universal Client for Windows 98 workstation. • IBM eNetwork Personal Communications for Windows 95 Version 4.2, or later • Novell NetWare for SAA: At the NetWare for SAA server: – Novell NetWare Version 4.11 – NetWare for SAA Version 2.2 installed with PTF SAA22010 applied. At the NetWare for SAA Client: – Novell NetWare Client Version 4.11. **Note:** You must **not** install the NetWare Client support supplied with Windows NT. – NetWare for SAA Version 2.2 client for Windows 95 shipped in PTF SAA22010, with APAR JR10793 applied. • IntraNetWare for SAA Version 3.0 or later. For TCP62 support, IBM eNetwork Personal Communications Version 4.2 , or later or the Host TCP/IP Access Feature component of CICS Universal Client for Windows 98 is required. See "Coexistence of TCP62 with SNA protocol" on page 39.
Application development language support
The following language compilers are supported: • IBM VisualAge® C++ Version 3.5 or later • Microsoft Visual C++ Version 5.0 or later • IBM VisualAge COBOL Version 2.1 or later • Microsoft Visual Basic Version 5.0 or later • Microsoft Visual Basic Script (VBScript) - part of Scripting Host Version 5.0 for Windows • IBM VisualAge PL/I Version 2.1 • IBM Object REXX Version 1.0.2 • Any other compiler compatible with the above, though only these compilers have been tested by IBM

The following table lists the software requirements of CICS Universal Client for Windows NT.

Table 6. Software requirements for CICS Universal Client for Windows NT

Operating system version
CICS Universal Client for Windows NT can run under any of the following operating systems: • Windows NT Workstation Version 4.0, or later • Windows NT Server Version 4.0, or later • Windows 2000 Service pack 3 is required for Year 2000 support, and Service pack 4 is required for Euro support.
NetBIOS support
NetBIOS support is provided by Windows NT.
TCP/IP support
TCP/IP support is provided by the Windows NT Workstation and Windows NT Server operating systems.
SNA support
APPC support is provided by: • Microsoft SNA Server Version 3.0 or later SNA Server itself may be installed on the workstation, or SNA Server Client may be installed on the CICS Universal Client for Windows NT workstation. • IBM eNetwork Communications Server for Windows NT Version 5.0 or later eNetwork Communications Server itself may be installed on the workstation, or eNetwork Communications Server Client for Windows NT may be installed on the CICS Universal Client for Windows NT workstation. • IBM eNetwork Personal Communications for Windows NT Version 4.2, or later • Novell NetWare for SAA: At the NetWare for SAA server: – Novell NetWare Version 4.11. – NetWare for SAA Version 2.2 installed with PTF SAA22010 applied At the NetWare for SAA Client: – Novell IntranetWare Client for Windows NT **Note:** You must **not** install the NetWare Client support supplied with Windows NT. – NetWare for SAA Version 2.2 client for Windows NT shipped in PTF SAA22010, with APAR JR10793 applied. • IntraNetWare for SAA Version 3.0 For TCP62 support, IBM eNetwork Personal Communications Version 4.2, or later, or the Host TCP/IP Access Feature component of CICS Universal Client for Windows NT is required. See “Coexistence of TCP62 with SNA protocol” on page 39.
DCE support

Table 6. Software requirements for CICS Universal Client for Windows NT (continued)

• IBM Distributed Computing Environment for Windows NT Version 2.0, or later • Gradient PC-DCE for Windows NT/95 Version 2.2
Application development language support
The following language compilers are supported: • IBM VisualAge C++ Version 3.5 or later • Microsoft Visual C++ Version 5.0 or later • IBM VisualAge COBOL Version 2.1 or later • Microsoft Visual Basic Version 5.0 or later • Microsoft Visual Basic Script (VBScript) - part of Scripting Host Version 5.0 for Windows • IBM VisualAge PL/I Version 2.1 • IBM Object Rexx Version 1.0.2 or later (ECI applications only) • IBM VisualAge Interspace® Version 5.2 • Any other compiler compatible with the above, though only these compilers have been tested by IBM

CICS server PTF requirements

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

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

Code page support

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

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

Chapter 3. Installing CICS Universal Clients

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

- Program Files
- Documentation
- Development Kit
- IBM Java Runtime Environment (JRE) Version 1.1.8
- CICS Host TCP/IP Access Feature.

  You install this component or not depending on how you want the TCP62 and SNA protocols to coexist. For more information, see "TCP62 configuration" on page 37.

You can also install the Adobe Acrobat Reader if required.

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

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

Installing CICS Universal Clients

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

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

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

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

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

Follow the instructions for downloading the client software. When you download, files are extracted and the SETUP.EXE program is run automatically. Installation then proceeds as described in the following section.

© Copyright IBM Corp. 1994, 1999

Installing CICS Universal Client on Windows platforms

This section applies to installing CICS Universal Client on Windows 95, Windows 98 and Windows NT.

Before you install CICS Universal Client on Windows NT, you must be logged onto the machine with Administrator privileges. You may have Administrator privileges by being a member of the local Administrators group, or by being a domain administrator.

Types of installation

For CICS Universal Client you can select one of the following types of installation:

Typical
Installs the runtime component of CICS Universal Client, and the documentation, but not the development kit.

Compact
Installs only the runtime components of the CICS Universal Client.

Custom
Allows you to select which components of the product to install and to specify the language for the message files.

Installing CICS Universal Client

It is strongly recommended that you exit all Windows programs before running the Setup program.

CAUTION:
Be very careful if you are installing CICS Universal Client onto a workstation that already has a CICS for Windows NT server installed. Both products have some files of the same name, therefore you must ensure that the installation of the CICS Universal Client does not affect the server. This is particularly important with respect to environment settings.

Remember, if installing onto Windows NT, you must be logged on with Administrator privileges.

1. Insert the CICS Universal Client CD-ROM in the CD-ROM drive. If you are installing via a network drive, connect to that drive.
2. Enter a Run command like the following:

   ```
   d:\SETUP
   ```

   where *d:* is a CD-ROM, or network drive.

   The InstallShield Wizard is prepared and then a Welcome panel is displayed.

3. Select **Next** and the Software License Agreement panel is displayed.
4. To accept the License Agreement, select **Yes**.
5. If you have a Version 2 or Version 3.0 CICS Client on your system, you are asked whether you wish to remove this old version. Select **Next** to upgrade to Version 3.1. This will remove the Version 2 CICS Client and will convert your old customized (*.ini) files to the new format. If you want to back up your Version 2 CICS Client, select **Cancel** at this point.
6. If you are upgrading from Version 2, the Delete panel is displayed. Select **Select All** and then **Delete**.

   Note: If you installed the CICS Gateway for Java in Version 2, this will not be deleted automatically. Before you delete the CICS Gateway for Java you may want to save the Gateway.properties file.
7. If you have backed up a Version 2 CICS Client, start the installation program again, and select **Next** and **Yes** on the introductory panels.
8. The Choose Destination Location panel is displayed.

   If you want to install into a different directory from the default, select **Browse**, and enter the required directory in the Choose Folder panel. Select **Next** to install into the default folder.
9. On the Setup Type panel, select a Typical, Compact, or Custom setup:
   - To install CICS Universal Client with the runtime and documentation components, select **Typical**
   - To install just the runtime component of CICS Universal Client, select **Compact**.
   - To choose individual components of the CICS Universal Client including the development kit and documentation, select **Custom**.

   Note: If you select a Typical or Compact installation, no programming samples are installed. Also, message files are installed in the language to match your locale.
10. Select **Next**.
11. If you selected a Custom setup, a Select Components panel is displayed.

    Select the check box next to each component you want to install. The **Space Required** and **Space Available** fields are updated as you select components.

    To select sub-components of **Development Kit**, select that component, then select **Change**, and the Select Sub-components panel is displayed. Select the check box for each component you want to install, and select **Continue**.

    Select **Next**.
12. If you selected a Custom installation, the Choose Language panel is displayed. On this panel a language is already selected according to your

locale. If you need to install the CICS Universal Client files in another language, select the radio button for that language. Select **Next**.

13. The Select Program Folder panel is displayed. If you want to install into a different folder from the default, enter the name of another folder, or select one from the existing folders list. Select **Next** to proceed.
14. The Select Components panel is displayed. Select check boxes depending on whether you want Setup to:

    Update the PATH
    If you do not select this check box, you will have to set the PATH manually

    Install the client as a Windows NT service
    If you do not select this option, you cannot run CICS Universal Client for Windows NT as a Windows NT service.

    Select **Next**.
15. The COM panel is displayed. Choose which type of COM object you want to register. You can register CICS Universal Client COM API resources as either:

    In process
    This refers to .dll files. This is the preferred type.

    Out of process
    This refers to .exe files. For backward compatibility with older CICS Universal Client applications, you should choose **Out of process**.

    For information about registration of COM objects, see “Registering COM libraries” on page 29.
16. The Install Java panel is displayed. To function properly, the CICS Universal Client requires a Java Virtual Machine (JVM). You can use the one you have already, but it is recommended that you install the supplied IBM JRE to ensure compatibility and the best performance. Select a radio button to install the IBM JRE, or use another JVM.
17. If you do not choose to use the IBM JRE, the Select JVM panel is displayed, and this displays the different JVMs you have on your system. Select the radio button for the JVM you wish to use, or select the **Enter the path to another Java environment** radio button. If you select the latter, the Java Path panel is displayed. Enter the fully-qualified path to your JDK, for example, `C:\JRE1.1.8\BIN\JRE.EXE`
18. If you do not have TCP62 installed, the Install Option panel is displayed, asking you whether you wish to install TCP62. TCP62 allows SNA support over TCP/IP. If you have already installed IBM eNetwork

Personal Communications, you already have TCP62 support. Select **Install TCP 62 support** if required, and select **Next**.

19. CICS Universal Client files are copied to your hard disk.

    If you chose to install the documentation, Adobe Acrobat Reader is installed for you, if you do already have it installed, or have a down-level version.

20. The Setup Complete panel is displayed. You are given the opportunity to view the README file, and to start the configuration tool. Select **Finish** to complete the setup. (If you chose to install TCP62 support, you are only given the option to restart the system.)

The system PATH and network drives

To run CICS Universal Clients, the PATH statement in your Windows environment should be updated with the CICS Universal Clients executable files path. You can do this manually or automatically by selecting this option during Setup. After installation, make sure that in the PATH statements there are no network directories before the CICS Universal Clients executable files path.

This is necessary because if CICS Universal Client is to run as a Windows NT service, its environment is determined when the system is started and is formed from System Environment Variables only.

The environment used by Windows NT services is established before the network drives are attached at Windows NT startup. Consequently, any directory in the system PATH that involves a network drive will not be available to Windows NT services.

Furthermore, Windows NT discards the remainder of a PATH if it finds a non-existent directory, therefore you will have problems running CICS Universal Client if network directories (or non-existent local directories) exist in your system PATH before the CICS Universal Client executable files paths.

A typical symptom of this problem is the appearance of a Service Control Manager dialog box indicating a return code of `0xc0000022` when you try to start CICS Universal Client.

Uninstalling CICS Universal Client

This section describes how to uninstall CICS Universal Client. You can uninstall all of the CICS Universal Client or just individual components of the products.

On Windows NT, you must have Windows NT administrator privileges to uninstall CICS Universal Client.

You can uninstall all of the CICS Universal Client products in two ways:

1. Using **Add/Remove Programs** in the Windows Control Panel. **IBM CICS Universal Client** is shown in the list of programs, and also **TCP 62** if you chose to install it.
2. Running Setup again and selecting the **Uninstall** option.

When you run Setup after you have already installed CICS Universal Client the CICS Universal Client Utility Setup panel is displayed. This gives you the following options.

Add/Remove This allows you to add new components, or remove installed components from the current installation, including the TCP62 component.

Reinstall This allows you to repeat the last installation to restore missing files and settings. This does not include the TCP62 component.

Uninstall This allows you to completely uninstall CICS Universal Client.

If you uninstall a product then when you re-install, the Setup program presents the installation directories of the uninstalled product as the default.

Problems during installing or uninstalling

You may experience problems when installing or unininstalling, for example if there is a system crash. In such cases you can enter the following command to clean up the installation:

```
setup.exe CLEAN
```

This command will delete registry entries and directories of CICS Universal Client, allowing you to perform the install or uninstall cleanly.

Updating CICS Universal Client

After installing CICS Universal Client, you can update your software with updated versions of the product, or corrective service software, which provides fixes for your installed software.

For corrective service software information, go to our Web site at:

```
http://www.ibm.com/software/ts/cics/
```

and follow the **Support** link.

You install updated versions of CICS Universal Client, or corrective service software using SETUP.EXE, in the same way as for the initial installation.

When you update with a newer version of CICS Universal Client, Setup retains any customized (*.ini) files from the older version.

Silent installation

You can use a so-called silent installation to install CICS Universal Client without going through any of the Setup panels. It is only necessary to run the setup program—no user input is required.

In a silent installation, the Setup program uses the information in a response file (setup.iss by default) to determine the installation options. CICS Universal Client can therefore easily be installed with the same installation options on many workstations.

Note: You cannot install the IBM JRE using a silent installation.

You must perform the following to create and verify the response file for a silent installation:

1. Plan the setup
2. Create the response file
3. Play back the silent setup
4. Check for errors.

Planning the setup

Before creating the response file, you should decide on the setup options that are required. Refer to see “Installing CICS Universal Client on Windows platforms” on page 22 for information on the installation dialogs.

Creating the response file

To create a response file, run setup with the record parameter:

```
setup -r
```

All your setup choices are then recorded in the response file, which is called setup.iss by default, and is written to the Windows directory.

However, you can also use the -f1 parameter to specify the name and location of the response file created, for example:

```
setup -r -f1d:\install\cicscli.iss
```

Note that you can specify setup parameters using either a hyphen (-) or a slash (/).

The response file is a plain text file that you can edit if necessary.

Playing back the silent setup

To perform a silent setup, you run setup.exe with the -s parameter. No messages are displayed during silent setup, but setup information is written to a log file named setup.log in the same directory as setup.exe.

You can use the -f1 and -f2 parameters with setup.exe to specify the name and location of the response file and the log file. For example:

```
setup -s -f1d:\install\cicscli.iss -f2d:\install\setup.log
```

You do not need to specify the f1 parameter if the response file is in the same directory as setup.exe.

To verify that a silent setup succeeded, look at the ResultCode value in the [ResponseResult] section of setup.log. The ResultCode value will indicate a successful setup (0) or an error code, as described in the next section.

Checking for errors

The ResultCode value in the [ResponseResult] section of setup.log can contain values as follows:

0	Success
-1	General error
-2	Invalid mode
-3	Required data not found in the response file
-4	Not enough memory available
-5	File does not exist
-6	Cannot write to the response file
-7	Cannot write to the log file
-8	Invalid path to response file
-9	Not a valid list type
-10	Not a valid list type (string or number)
-11	Unknown error during setup
-12	Dialog boxes are out of order
-51	Cannot create the specified folder
-52	Cannot access the specified file or folder
-53	Invalid option selected

Upgrading silently

If you need to install CICS Universal Client over an existing installation, you can also do this silently.

To run an upgrade silent installation, use the following parameters, separated by semicolons, *before* the -s, f1, and f2 parameters:

UPGRADE
Specifies that the response file will be used to upgrade an existing installation

CICSCLI=*path*
Specifies the name and location of the configuration file to be used.

CICSCOL=*path*
Specifies the name and location of the color mapping file to be used.

CICSKEY=*path*
Specifies the name and location of the keyboard mapping file to be used.

The CICSCLI, CICSCOL, and CICSKEY parameters are optional. If they are not specified, the .ini files from the existing installation will be copied to the upgraded installation. If these parameters specify the default names for the .ini files (ctg.ini, cicscol.ini, cicskey.ini) the files in the existing installation are overwritten.

For example:

```
setup UPGRADE;CICSCLI=ctg.ini;CICSCOL=cicscol.ini;CICSKEY=cicskey.ini
      -s -f1d:\install\cicscli.iss -f2d:\install\setup.log
```

Restrictions of silent installation

You cannot use a silent installation for uninstalling or adding or removing components of CICS Universal Client, as can be done using the Utility Setup panel.

Registering COM libraries

COM libraries are provided with CICS Universal Client and these must be registered at installation time. Registration registers the COM classes, associated ProgIDs, and the type libraries for each of the libraries.

Visual Basic will only use the type libraries if you register them to each Visual Basic Project. It is recommended that you do this to make full use of the features and performance enhancements of these type libraries. For more information refer to *CICS Universal Clients COM Automation Programming*.

VBScript does not use type libraries. All the COM libraries support automatic registration and de-registration and you may register in-process or out-of-process libraries, during the installation of CICS Universal Client. If both the in-process and out-of-process libraries are registered, by default Visual Basic and VBScript will use the in-process versions.

Registering in-process servers

To register in-process servers use the Microsoft supplied program REGSVR32. For example to register the ECI COM libraries enter the command:

```
REGSVR32 CCLIECI.DLL
```

to de-register enter the command:

```
REGSVR32 /U CCLIECI.DLL
```

Registering out-of-process servers

To register the out-of-process servers, run the executable and, when a window appears indicating that registration has been done, close it. For example to register the EPI libraries run the program **CCLOEPI**. To unregister the out-of-process servers, run the executable with the /unregister parameter. For example to unregister the EPI libraries run the program **CCLOEPI /unregister**.

Other actions required in setting up a CICS Universal Client

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

If you want the CICS terminal emulation to be started automatically when Windows 95 or Windows 98 starts up, move or copy the CICS Terminal icon to the Startup folder.

If you want the CICS terminal emulation to be started automatically when Windows NT starts up, move or copy the CICS Terminal icon to the Startup group.

Chapter 4. Setting up client/server communication

This chapter describes the following:

- "NetBIOS configuration" on page 33
- "TCP/IP configuration" on page 34
- "APPC configuration" on page 40
- "Named Pipes configuration" on page 46

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

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

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

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

- Enable and set up NetBIOS, TCP/IP, or APPC communication
- Set up the required settings in the configuration file, (see "Using the configuration tool" on page 48)

Client/server definitions

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

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

© Copyright IBM Corp. 1994, 1999

As far as possible, the example values are based on the values in the supplied configuration file, ctg.ini. See “Using the configuration tool” on page 48 for more information.

- The definitions that must match.

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

Definition checklists

CICS server definitions

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

CICS Universal Client definitions

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

Table 7. Matching definitions: CICS Universal Clients to CICS for OS/2 using NetBIOS

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

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

This book does not give details for every possible client/server configuration. Typical examples are given representing the most likely configurations. More example configurations are described in the *Revealed! CICS Transaction Gateway* *with more CICS Clients Unmasked* book, and also in a set of special configuration documents, see "Sample configuration documents" on page 169.

NetBIOS configuration

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

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

Definition checklists

The following shows example definitions for connecting CICS Universal Client for Windows 98 or CICS Universal Client for Windows NT to a CICS for OS/2 server:

CICS server definitions

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

CICS Universal Client definitions

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

Table 8. Matching definitions: CICS Universal Clients to CICS for OS/2 using NetBIOS

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

NetBIOS support is provided with Windows 95, and no further configuration is required after installation.

NetBIOS configuration

NetBIOS support is provided with Windows NT, because NETBEUI (NetBIOS extended user interface) is automatically installed during Windows NT setup. However, via the Network Settings panel, you must configure the NetBIOS Interface with settings like the following example, which is for an IBM Auto 16/4 Token Ring ISA adapter:

Network Route	0: Nbf->IbmTok4->IbmTok41
Lana Number	0. This matches the Adapter setting in the configuration file.

NetBIOS resources

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

NetBIOS Names	1
NetBIOS Commands	Value of Maximum servers + 1
NetBIOS Sessions	Value of Maximum servers.

TCP/IP configuration

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

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

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

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

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

Definition checklists

The following shows example definitions for connecting CICS Universal Client for Windows 98 or CICS Universal Client for Windows NT to a TXSeries for Windows NT server:

CICS server definitions

TCP/IP Definitions (Server)	Ref. Key	Example
Host Name	1	cicstcp
IP Address	2	192.113.36.200
Subnet Mask	3	255.255.255.0
Domain Name Server	4	192.113.36.250
Domain Name	5	acme.ibm.com

TXSeries for Windows NT Listener Definitions	Ref. Key	Example
Listener name		ANYNAME
Protocol		TCP
Note: You can leave other listener definitions as the default values. See *TXSeries Windows NT Administration Reference* for more information.		

CICS Universal Client definitions

TCP/IP Definitions (Client)	Ref. Key	Example
Hostname		myhost
IP Address		192.113.36.78
Subnet Mask	3	255.255.255.0
Domain Name Server	4	192.113.36.250
Domain Name	5	acme.ibm.com

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

Table 9. Matching definitions: CICS Universal Clients to TXseries for Windows NT

Ref. Key	TCP/IP (Server)	TCP/IP (Client)	Client Configuration Settings	Example
1	Host Name		Host name	cicstcp.ibm.com
2	IP Address		IP address	192.113.36.200
3	Subnet Mask	Subnet Mask		255.255.255.0
4	Domain Name Server	Domain Name Server		192.113.36.250
5	Domain Name	Domain Name		acme.ibm.com

TCP/IP support is provided by the Windows 95, Windows 98 and Windows NT operating systems. Refer to the Windows 95/98/NT documentation for information on configuring TCP/IP.

TCP/IP can also be provided by any other product conforming to the WINSOCK API. For information on installing such products, see the associated documentation.

Verifying the TCP/IP installation

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

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

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

```
ping 192.113.36.200
```

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

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

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

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

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

Figure 3. Example messages returned by ping command

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

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

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

TCP62 configuration

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

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

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

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

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

TCP62 configuration

| For detailed information on configuring TCP62, refer to the *Revealed! CICS Transaction Gateway with more CICS Clients Unmasked* book and to any configuration document that is available, see "Sample configuration documents" on page 169.

On OS/390

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

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

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

On OS/390, you must:

1. Install a TCP major node, which defines the AnyNet interface between TCP/IP and VTAM. For further information about how to do this, see the *Guide to SNA over TCP/IP* book, SC31-6527.
2. Install a CDRSC major node, which defines the remote Client device and instructs VTAM to route any session requests through the TCP/IP Physical Unit (ALSLIST).
3. Check that the Physical Unit (PU) for the AnyNet interface is active.

On CICS and VTAM

On CICS, you must:

1. Define an APPC connection to the client workstation. (The connection can be statically defined, or autoinstalled.)
   - On the MODENAME option of the SESSIONS definition, specify the same modename as that specified in the configuration file. (The default modename in the configuration file is TCP62.)
   - On the MAXIMUM option of the SESSIONS definition, specify the second value as one, that is, that CICS for MVS/ESA is to have one

   contention winner. For example, `MAXIMUM(8,1)` means that the modeset is to support eight sessions, and that CICS for MVS/ESA has one contention winner.

2. Add an entry to the VTAM logon mode (LOGMODE) table for the modename specified on the SESSIONS definition. This entry specifies the class of service required for the group of sessions.

On the client workstation

On the client workstation, you must install and configure TCP/IP and ensure that the MVS system name is added to the TCP/IP HOSTS file. You must install TCP62 support as provided by eNetwork Personal Communications (see the following section), and configure the configuration file see “Using the configuration tool” on page 48. The driver is CCLTCP62.DLL

Coexistence of TCP62 with SNA protocol

Depending on whether you wish to use TCP62 alone, or in coexistence with SNA protocols, you must install versions of eNetwork Personal Communications as follows:

- **To use TCP62 but not SNA protocols**:

  Install the Host TCP/IP Access Feature component of the CICS Universal Client. (This is a mini version of IBM Personal Communications supporting only TCP62. If any other Personal Communications support is required, you should install IBM Personal Communications Version 4.2 or later.)

- **To use TCP62 and SNA protocols**:

  Do ***not*** install the CICS Host TCP/IP Access Feature component of the CICS Universal Client. Install IBM Personal Communications Version 4.2 or later to provide both SNA and TCP62 support. No other supported SNA product can be used with TCP62.

Firewall implications

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

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

APPC configuration

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

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

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

Configuring APPC for CICS Universal Client for Windows

The following sections summarize APPC configuration for CICS Universal Client for Windows 98 and CICS Universal Client for Windows NT.

Microsoft SNA Server

APPC support can be provided by Microsoft SNA Server. You will install SNA Server client on a CICS Universal Client for Windows NT or CICS Universal Client for Windows 98 machine, which will communicate with the CICS server through the SNA Server machine.

For information on installing and configuring Microsoft SNA Server, and SNA Server clients, see the *Microsoft SNA Server Installation Guide*, and the *Microsoft SNA Server Administration Guide*.

To allow workstations using the Windows SNA Client to communicate with other systems, the LUs used by the client systems should be defined as **Local APPC LU**s to the SNA Server workstation.

Setting up the SNA Server Windows client: On the SNA Server Windows client, you require the following settings:

SNA client mode	Remote
Primary Server	Name of the SNA Server workstation that the

client systems using SNA Client will use. This computer name of the SNA Server is displayed in both the SNA Server Admin **Servers and Connections** window, and in the **Network Settings** control panel dialog box.

To receive ATI on SNA Server Windows client terminals, you must run the CCLMSATI command to configure the SNA Server Client for the CRSR transaction. The syntax of the command is:

```
CCLMSATI luname
```

where *luname* is the LocalLU you are using.

If CRSR is already configured, the LocalLU setting changes to match the new value.

Before you start the Server Windows 95 client, you must run SNABASE.EXE to start the SNA Server Client SnaBase. If you do not do so, the CICS Client will not be able to communicate with the SNA Server.

Detailed configuration: Detailed information on connecting CICS Universal Client for Windows to CICS Transaction Server for OS/390 using SNA Server is given in the *Revealed! CICS Transaction Gateway with more CICS Clients Unmasked* book and any configuration document that is available, see “Sample configuration documents” on page 169.

IBM eNetwork Communications Server

APPC support can be provided by IBM eNetwork Communications Server. You can install eNetwork Communications Server itself on a CICS Universal Client for Windows NT machine, but it is more likely that you will install eNetwork Communications Server client on a CICS Universal Client for Windows NT or CICS Universal Client for Windows 98 machine, which will communicate with the CICS server through the eNetwork Communications Server machine.

For information on installing and configuring eNetwork Communications Server, and eNetwork Communications Server clients, see the *IBM eNetwork Communications Server Up and Running Guide*.

You must run Configuration to configure Global Data and the LU6.2 Server List.

Setting up the eNetwork Communications Server (server): You must run SNA Node Configuration to configure APPC support, define a connection and the local LU, partner LU and mode specified in the configuration file. You must configure the local LU to be independent.

If using a eNetwork Communications Server SNA Client, ensure that the **SNA API client use** check box, in the Local LU 6.2 definition panel, is selected.

To enable ATI against CICS Client terminals, the transaction program CRSR must be defined on the eNetwork Communications Server, whether or not a eNetwork Communications Server SNA Client is being used, as follows:

TP name	CRSR
Service TP	No
Complete pathname	C:\CICSCLI\BIN\CCLCLNT.EXE
Program parameters	CRSR
Conversation type	Mapped
Synchronization level	Any
Conversation security required	No
Receive_Allocate timeout	0
Incoming allocate timeout	default
TP instance limit	1
PIP allowed	No
For SNA Client use	Yes/No
Dynamically loaded	No
Full duplex support	No

Setting up the eNetwork Communications Server Windows client: You must run Configuration to configure Global Data and the LU6.2 Server List.

To enable ATI against CICS Client terminals, the transaction program CRSR must be defined on the SNA Client, as follows:

TP name	CRSR
LU alias	Alias of the local LU defined for the connection
Complete pathname	C:\CICSCLI\BIN\CCLCLNT.EXE
Program operation	Attach Manager Started
Program parameters	CRSR

Service TP	No
Attach Manager Started	No

The Attach Manager must be started.

Detailed configuration: Detailed information on connecting CICS Universal Client for Windows to CICS Transaction Server for OS/390 using eNetwork Communications Server is given in any configuration document that is available, see “Sample configuration documents” on page 169.

IBM eNetwork Personal Communications

You must run ’SNA Node Configuration’ to configure APPC support. Define a node, a device and a connection. Use an adjacent CP type of ″APPN Node″ or ″Back-Level LEN″ in the connection definition. Define the local LU, partner LU and mode specified in the configuration file. You must configure the local LU to be independent.

The node must be started from ’SNA Node Operations’ before starting a CICS Universal Client connection to a server that uses eNetwork Personal Communications.

To enable ATI against CICS Universal Client terminals, the transaction program CRSR must be defined, as follows:

TP name	CRSR
Service TP	No
Complete pathname	C:\CICSCLI\BIN\CCLCLNT.EXE
Program parameters	CRSR
Conversation type	Mapped
Synchronization level	Any
Conversation security required	No
Receive_Allocate timeout	0
Incoming allocate timeout	default
TP instance limit	1
PIP allowed	No
For SNA Client use	Yes/No
Dynamically loaded	No
Full duplex support	No

Operation of eNetwork Personal Communications: The SNA protocol driver requires that the IBM eNetwork Personal Communications node is started before a CICS Universal Client SNA connection. The TCP62 protocol driver automatically starts the IBM Personal Communications node when a CICS Universal Client TCP62 connection is started, if it is not already running. If the node is started automatically the default configuration file is used, if there is one. If the node is started manually and the TCP62 protocol is required, an AnyNet SNA/IP device should be configured specifying the SNA domain name suffix in the Anynet domain name suffix setting in the configuration file.

Detailed configuration: Detailed information on connecting CICS Universal Client for Windows to CICS Transaction Server for OS/390 using eNetwork Personal Communications is given in any configuration document that is available, see "Sample configuration documents" on page 169.

NetWare for SAA

CICS Universal Client for Windows 98 and CICS Universal Client for Windows NT use the NetWare for SAA APPC API. Bearing this in mind, you should refer to the relevant configuration information in the NetWare for SAA manuals accordingly.

To configure NetWare for SAA:

1. Install the following, on any client machine running NetWare for SAA Version 2.2 and connected to the NetWare for SAA server that you wish to configure.
   - NWSAA Services Manager (SSM)
   - NWSAA Administrator
   - NWADMIN Client Administrator (CICS Universal Client for Windows 98 only)

   See NetWare for SAA Version 2.2 Installation manual, Chapter 2.
2. Start NWSAA Administrator:
   a. Select the server you wish to configure.
   b. Select **Functions**, then **LogIn To Server**.
   c. Select the NetWare for SAA Node and configure the Peer PU profile default setting.
   d. Select and configure **Data Link Adapters**.

   To configure the NetWare for SAA server for CICS Universal Client for Windows NT:

   a. Select **APPC**.

b. Configure a Local LU. You must specify the LU name in the Local LU name setting in ctg.ini.

c. Configure a Partner LU. You must specify the Partner LU name in the Partner LU name setting in ctg.ini, without the Network qualifier. You must also set LUAliasNames = Y.

d. Configure a mode. You must specify the mode name in the Mode name setting in ctg.ini.

e. If ATI processing is required, configure a Transaction Program as follows:

Local TP name:	CRSR
Sync level:	Confirmed
Conversation type:	Mapped
Require conversation security:	unchecked

f. No Host or Peer PU profile is required.

3. Start NetWare for SAA Services Manager as follows:

a. Create a new domain with the required NetWare for SAA server(s) in it.

b. Select the NetWare for SAA server containing the APPC configuration done in the previous step

c. Select **Server**, then **Link/Service Information** and check that there is a link to the required host.

d. Select **Server**, then **Independant LU6.2** and check that there are active sessions with the required partner LU. If there are not, reboot the NetWare for SAA server and check again. Once there are active sessions, you are ready to start the CICS Client.

4. If ATI processing is required: Start NetWare Administrator (NWADMN95.EXE can be found on the NetWare for SAA server in \SYS\PUBLIC\WIN95).

a. **For each user,** select **APPC Configuration** and add an Attach Manager Record as follows:

TP Name:	CRSR
TP Type:	STANDARD
TP Operation:	OPERATOR
TP Filename:	blank
TP Parms:	CRSR
TP Presentation:	blank
TP Icon Filename:	blank

LU Name:	Local LU name

b. **For the NetWare for SAA server,** select **APPC Configuration** and add an Attach Manager Record as follows:

TP Name:	CRSR
TP Type:	STANDARD
TP Operation:	OPERATOR
TP Filename:	blank
TP Parms:	CRSR
TP Presentation:	blank
TP Icon Filename:	blank
LU Name:	blank

Operation: The NetWare for SAA Attach Manager must be started to enable ATI requests to be received.

Detailed configuration: Detailed information on connecting CICS Universal Client for Windows to CICS Transaction Server for OS/390 using NetWare for SAA is given in the *Revealed! CICS Transaction Gateway with more CICS Clients Unmasked* book and any configuration document that is available, see “Sample configuration documents” on page 169.

Named Pipes configuration

CICS Universal Client for Windows NT can communicate with TXSeries (Version 4.3 or later) for Windows NT servers that are on the same computer as the CICS Client, using a protocol based on Windows Named Pipes. This provides a fast communications method independent of any network protocol.

The following definitions are required:

- The TXSeries server must have a Listener Definition for the ’NamedPipe’ protocol.
- The **NamedPipeName** attribute must match the Named Pipe Name setting in the client configuration. All other attributes can use their default values. Named Pipe Name must be 6 alphanumeric characters, for example, `CICSCC`.
- Using the configuration tool you must select the **Named Pipe** protocol, in the Server panel. The Named Pipe Name setting must match the **NamedPipeName** attribute in the TXSeries server Listener Definition. In the Server section you should also select **Use upper case security**. This will create the correct settings in the configuration file including a `LOCALNP` Driver section with DriverName set as `CCLCLPIP`.

Chapter 5. Customization

This chapter describes

- "Renaming the client configuration and mapping files"
- "Using the configuration tool" on page 48
- "Customizing the keyboard" on page 74
- "Customizing the screen colors" on page 78
- "Testing the client/server link" on page 82

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

Renaming the client configuration and mapping files

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

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

For CICS Universal Client for Windows NT, you can set these environment variables via the **System** icon in the Control Panel, for example:

© Copyright IBM Corp. 1994, 1999

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

Specifying which Java Virtual Machine to use

You use the ctgjava command to set the location of the JVM that Java programs such as the configuration tool use. The command has the following options:

```
ctgjava /s=jvmlocation [/v]
```

where *jvmlocation* is a fully qualified path to the JVM. The option /v displays the location of the JVM currently being used.

Using the configuration tool

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

To start the configuration tool, click on its icon, or enter the **ctgcfg** command. The configuration tool will use the JVM specified by the ctgjava command, see "Specifying which Java Virtual Machine to use".

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.

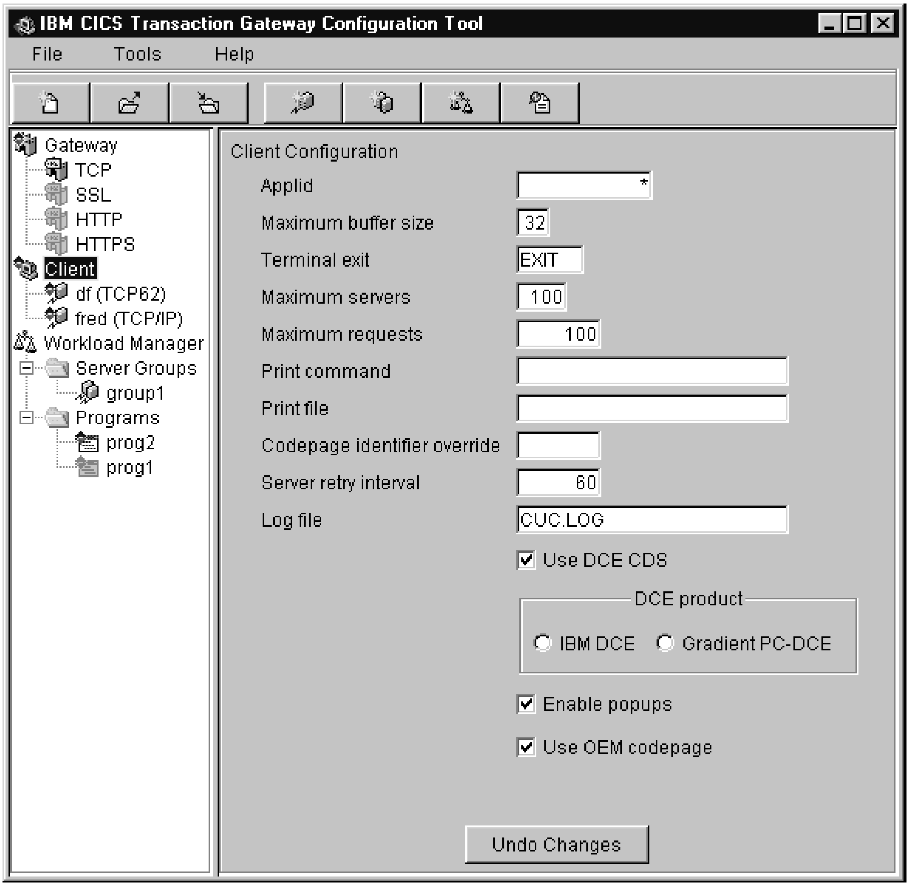
The configuration is stored by default in the ctg.ini file in the bin subdirectory where you installed the CICS Universal Clients. You can edit this file directly, but it is recommended that you use the configuration tool to perform configuration.

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.



Client Configuration

Applid *

Maximum buffer size 32

Terminal exit EXIT

Maximum servers 100

Maximum requests 100

Print command

Print file

Codepage identifier override

Server retry interval 60

Log file CUC.LOG

☑ Use DCE CDS

DCE product

○ IBM DCE ○ Gradient PC-DCE

☑ Enable popups

☑ Use OEM codepage

Figure 4. The configuration tool

Note: On your platform, the configuration tool may not appear exactly as shown in Figure 4.

Tree structure

The tree structure (see Figure 4) 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.

Workload Manager
Contains two subnodes:

Server groups
Server groups are used to group together CICS server definitions with different port numbers. These server instances are used by the Workload Manager to balance workload across CICS servers. Each server group subnode contains the server definitions associated with that server group.

Programs
Contains a number of subnodes, one for each program associated with the Workload Manager. Each program subnode contains the server definitions with which the program is associated.

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.

New Server Group
Creates a new Server group for the Workload Manager.

Delete Server Group
Deletes a Server group subnode from the Workload Manager node.

New Program to Bias
Allows you define a new program for biasing by the Workload Manager.

Delete Biased Program
Deletes a program subnode 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.

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.

For more information on the use of Maximum servers, see “NetBIOS resources” on page 34.

Maximum requests

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

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

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

Print command

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

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

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

A command file may be necessary to act as an interface between the syntax of the invocation command defined here, and more general operating system syntax. For example, if the desired final command was COPY printfile LPT2, a simple command file would be required to reorder the parameters. It is the responsibility of the Print Command to delete the temporary print file after it has finished processing it.

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

Print file

Enter a character string, 1 to 256 characters long.

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

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

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

Note: This setting acts only as a default. The terminal and print emulators provide options to override this value. (See “CICSTERM command reference” on page 103 and “CICSPRNT command reference” on page 108).

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.

On Windows it is not necessary to supply a CCSID, as you can use the **Use OEM codepage** setting to provide support for Euro enabled codepages. If

Codepage identifier override is specified, it overrides the codepage value generated using the **Use OEM codepage** setting. You might do this, for example, to switch back to codepage 850 or 1252 to indicate to the CICS server that the Euro should not be used.

For more information on CCSIDs and data conversion, refer to “Appendix A. CICS Universal Clients data conversion” on page 161.

Server retry interval

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

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

Log file

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

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

Use DCE CDS

Select this check box to use the Cell Directory Server (CDS) of DCE to locate CICS servers. The CDS is used, provided DCE is installed and configured on the client machine. The CICS Universal Client can then communicate with any of the located CICS servers using DCE Remote Procedure Calls (RPC).

If you do not select this check box, you must make appropriate settings for CICS servers as required in the Server settings. In any case, you must select which type of DCE you want to use by selecting the **IBM DCE** or **Gradient PC-DCE** radio button.

This setting is provided because searching the CDS for CICS servers may take some time, particularly with a large DCE cell.

IBM DCE

Select this radio button if you want to use IBM DCE.

Gradient PC-DCE

Select this radio button if you want to use Gradient DCE.

Enable popups

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

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

Use OEM codepage

Select this check box if you want the CICS Universal Client to call the Windows operating system OEM codepage function to get the codepage identifier (for example, 850). If the workstation is Euro enabled, then codepage 850 is mapped to 858 within the client to indicate to the CICS Server that the Euro symbol can be used.

If you do not select this check box, the CICS Universal Client uses the Windows operating system ACP function to get the ASCII identifier (for example 1252). This should be used for Latin-1, Japanese, Korean and Simplified Chinese systems to provide appropriate conversions. If the workstation is Euro enabled, then codepage 1252 is mapped to codepage 5348.

For more information on codepages and data conversion, refer to “Appendix A. CICS Universal Clients data conversion” on page 161.

Configuring Server settings

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

Server name

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

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

Description

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

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

Initial transaction

Enter a transaction identifier of between 1 and 128 characters.

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

Model terminal definition

Enter a string of between 1 and 16 characters.

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

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

Use Windows credentials for security

Select this check box if you want to use the Windows NT password for CICS security.

To use to use this feature, you must do the following:

1. Change to the directory \Program Files\Common Files\IBM\IBM Integrated Sign ON
2. Enter `bthdinst -i CICSCLI`
3. Reboot.

When uninstalling you **must** uninstall in the following sequence:

1. Change to the directory \Program Files\Common Files\IBM\IBM Integrated Sign ON
2. Enter `bthdinst -u CICSCLI`
3. Reboot
4. Uninstall in the usual way.

For more information, refer to “Network Provider Interface (NPI)” on page 94.

Use upper case security

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

This setting is disabled by default.

TCP/IP

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

SNA

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

TCP62

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

NetBIOS

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

DCE

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

Named Pipe

Select this option to specify Local Named Pipe 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.

The \etc subdirectory is located as follows:

OS/2 c:\mptn\etc

Windows NT
c:\winnt\system32\drivers\etc

Windows 95/98
c:\windows\etc

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

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

The \etc subdirectory is located as follows:

OS/2 c:\mptn\etc

Windows NT
c:\winnt\system32\drivers\etc

Windows 95/98
c:\windows\etc

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

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

A timeout occurs if connection establishment takes longer than the specified time. The TCP/IP socket is closed and the return code passed back to the client application is either ECI_ERR_NO_CICS or CICS_EPI_ERR_FAILED.

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

SNA settings

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

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

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

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

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

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

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

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

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

TCP62 settings

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

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

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

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

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

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

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

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

IP address mask for LU name template (optional): Enter an IP address mask in big-endian, that is, most significant byte first order. This must be a hexadecimal number, for example, `FFFFFF80`.

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

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

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

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

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

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

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

Common TCP62 settings

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

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

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

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

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

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

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

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

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

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

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

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

The default is `sna.ibm.com`.

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

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

Advanced TCP62 settings

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

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

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

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

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

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

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

NetBIOS settings

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

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

The name is case-sensitive.

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

DCE settings

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

CICS region: Enter the name of a region on the CICS server. This must be a region name defined in the resource definitions for the CICS server.

IBM DCE: Select this radio button if you want to use IBM DCE.

Gradient PC-DCE: Select this radio button if you want to use Gradient DCE.

Local Named Pipe settings

To display these settings, select the **Named Pipe** option.

Named Pipe Name: Enter a name of up to six alphanumeric characters that match the **NamedPipeName** attribute specified by the TXSeries for Windows NT server in its Listener Definition.

Configuring the Workload Manager

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

For information on how to use the Workload Manager see “Chapter 8. Workload Manager” on page 125.

It is particularly important to understand what is meant by **Server Group**. Server groups are used to group together a number of CICS server definitions with different TCP/IP ports. A server group is equivalent to one physical CICS server (otherwise known as a CICS region). This is needed for CICS on AIX which often has multiple listeners (due to a limit of 256 concurrent connections on one port).

The Programs node is used to show which servers/server groups are set up to use a specific program.

Enable the Workload Manager

Select this check box to enable the Workload Manager. The default is that the Workload Manager is disabled.

Round robin

Select this radio button to select a round-robin algorithm for this server.

The round-robin algorithm assumes that all CICS servers (regions) are equally valid for selection. In the round-robin algorithm, when the client is initially started, it reads from the registry a list of all possible CICS servers to which any ECI request can be sent. An indication of the last server selected is also maintained. When a new ECI request arrives, the next server in the list is selected as the target for that ECI request. The next server indicator is then advanced until the last server has been selected, at which point the first server is selected again as the target for a request.

Biasing

Select this radio button to select a biased algorithm for the Workload Manager.

The biasing algorithm provides a means of balancing workload by specifying that workload should be biased in favor of particular CICS servers (regions). For example, if there are two servers and one has a bias of 75 and the other 25, then program requests are sent in a ratio of 3:1 to the first server. If a server fails, the internal biasing calculation changes. If two servers are available, one with a bias of 100 and the other with a bias of 0, then all requests are sent to the first server. If the first server becomes unavailable, all requests are now directed to the second server. The bias value of 0 means use only if no other server is available.

Another way of looking at biasing is:

```
probability a server will be selected =   bias for server
                                          ------------------
                                          total bias for all servers
```

The biasing algorithm only works for ECI calls and does not work for EPI calls. If you attempt to run an EPI application while the biasing algorithm is selected, the round-robin algorithm is used instead.

Region timeout

Enter a value in seconds between 0 and 3600. The default is 60 seconds.

When a CICS server (region) is selected as the target of a client request, and that server is unavailable, the server is flagged as invalid, and is no longer included in the list of target regions for requests. The Region timeout value specifies the number of seconds after which an invalid region can once more be included in the list of target regions.

There are two options on the **Tools** menu that allow you to configure server groups and programs. These options are only enabled when you select the Workload Manager node.

Configuring a server group

This panel allows you to define which servers belong to a particular server group.

To add a new server group, you select the **New Server Group** option.

To change the members of a server group, select the server group from the tree selection.

In the case of a new server group, enter a name in the **Server group name** field.

To add a server to a group, select a server from the **Servers not in server group** list on the right and then select the arrow button (<<).

To remove a server from a group, select the server from the **Servers in server group** list and then select the arrow button (>>).

Configuring programs

This panel allows you to define which servers are used for biasing in the workload management of a particular program.

To add a new definition, you select the **New Program to Bias** option.

To change a definition, select the program from the tree selection.

In the case of a new definition, enter a name in the **Program name** field.

To disable workload management for the program, deselect the **Manage workload for this program** check box.

To add a server to the list of servers used for biasing, select a server or server group from the **Servers not used for biasing** list on the right and then select the arrow button (<<).

To remove a server from the list, select a server from the **Servers used for biasing** list on the left and then select the arrow button (>>).

In the **Bias value** fields, you can enter a bias value for the server.

In the **Ratios** fields, percentage bias values for each server are shown. This field contains **Standby** if the bias value is 0, which means use only if no other server is available.

Manage workload for this program

Select this check box if you want to enable load balancing for the selected program.

If you disable load balancing for a particular program, the Workload Manager connects to the server specified in the ECI call, rather than randomly selecting a new server.

Trace settings

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

Trace Settings

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

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).
Workload Manager	The Workload Manager.
REXX API	The REXX API.
CICS Client Service	The CICS Universal Client Windows NT service.
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 cicscli.bin file. This is in the Client \BIN subdirectory.

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

For more information about tracing, refer to “CICS Universal Clients trace” on page 142.

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:

CICSCLI.INI	Client initialization file of CICS Clients Version 2 and Version 3.0.
CICSX.INI	Workload Manager configuration file of CICS Universal Clients for Windows 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 70.

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  [/l=file] [/o=file]
```

for each parameter, *file* can be:

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

/c=*file*

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

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

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

/l=*file*

Specifies the Workload Manager configuration file to be converted.

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

If the parameter is not specified, the CICSX_CONFIG_URL environment variable is used to locate the Workload Manager configuration file. If CICSX_CONFIG_URL is not set, the default name in the Client \BIN directory is assumed.

If no Workload Manager configuration file exists, no Workload Manager section is created in ctg.ini.

/o=*file*
Specifies the pathname of the converted file. The default is Client \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
5. LOADMANAGER (CICS Universal Client for Windows NT only)

These sections have the format:

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

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

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

CLIENT section

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

        MaxServers = 1                  # Only allow one server connection
        MaxRequests = 90                # Limit the maximum server interaction
        MaxBufferSize = 32              # Allow for a 32K maximum COMMAREA
#       CCSID=850
#       CPName = ABCD1234.EFGH5678      # The TCP62 client's fully qualified CP name
#       DomainNameSuffix = cicstcp.ibm.com      # Domain name suffix for TCP62 server

ENDSECTION
```

SERVER section

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

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

ENDSECTION
```

DRIVER section

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

        DriverName = CCLWNTSN           #CCLWNTNB, CCLWNTSN, 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.

LOADMANAGER section

```
SECTION LOADMANAGER
    Type=1
    Timeout=100
    BIAS=EC01;REGION1;20;CICSTCP5;20
    NOBALANCE=EC01
ENDSECTION
```

Configuration file properties

If you need to edit the configuration file manually, you need to know how the properties in the configuration file map to the configuration tool names.

The following sections provide a complete list of properties that can be specified. After each property, the corresponding configuration tool setting is given in parentheses.

Client section

SECTION CLIENT = * | *applid* (**Applid**)
CCSID=*codepage* (**Codepage identifier override**)
DCECELLDIRECTORY=Y|N (**Use DCE CDS**)
ENABLEPOPUPS =Y|N (**Enable popups**)
LOGFILE=*filename* (**Log file**)
MAXBUFFERSIZE=*nn* (**Maximum buffer size**)
MAXREQUESTS=*nnnnn* (**Maximum requests**)
MAXSERVERS=*nnn* (**Maximum servers**)
PRINTCOMMAND=*command* (**Print command**)
PRINTFILE=*filename* (**Print file**)
SRVRETRYINTERVAL=*nn* (**Server retry interval**)
TERMINALEXIT=*name* (**Terminal exit**)
USEOEMCP=Y|N (**Use OEM codepage**)
LOADMANAGER=Y|N (**Enable the Workload Manager** - Workload Manager option)
CPIPADDRESSMASK=*mask* (**IP address mask for CP name (optional)** - TCP62 option)
CPNAME=*name* (**Fully qualified CP name or template** - TCP62 option)
DOMAINNAMESUFFIX=*suffix* (**Anynet domain name suffix** - TCP62 option)
REMOTENODEINACTIVITYPOLLINTERVAL=*nnnn* (**Remote node inactivity timeout** - TCP62 option)
MAXWRAPSIZE=*nnnn* (**Maximum Client wrap size** - Trace option)
TRACE=CLI|TRN|DRV|API.1|API.2|CCL|EMU|TLD|CPP|LMG|SER|RXX|ALL
(Trace Settings - see the description of cicscli /d /m for more details)
TRACEFILE=*filename* (**Client trace file** - trace options)
ENDSECTION

Server section

```
SECTION SERVER =servername (Server name)
          DESCRIPTION=description (Description)
          INITIALTRANSID=transid (Initial transaction)
          MODELTERM=name (Model terminal definition)
          UPPERCASESECURITY=Y|N (Use upper case security)
          USENPI=Y|N (Use Windows credentials for security)
          PROTOCOL=protocolname (Protocol)
          protocolname is normally one of the following: TCPIP|SNA|DCE|TCP62|LOCALNP|NETBIOS
          REGION=servergroupname (Configuring a server group - Workload Manager option)

          TCP properties
          NETNAME=ipaddress (Hostname or IP address)
          PORT=nnnnn (Port)
          CONNECTTIMEOUT=nnnnn (Connection timeout)
          TCPKEEPALIVE=Y|N (Send TCP/IP Keepalive packets)

          SNA properties
          NETNAME=luname (Partner LU name)
          LOCALLUNAME=LUName (Local LU name)
          MODENAME=modename (Mode name)
          LUALIASNAMES=Y|N (Use LU alias names)

          DCE properties
          NETNAME=dceregion (CICS region)

          TCP62 properties
          NETNAME=luname (Partner LU name)
          LUNAME=luname (Local LU name or template)
          LUIPADDRESSMASK=mask (IP address mask for LU name template (optional))
          MODENAME=name (Mode name)
          SNASESSIONLIMIT=nnn (Maximum logical SNA sessions)
          SNAMAXRUSIZE=nnn (Maximum SNA RU size)
          SNAPACINGSIZE=nnn (SNA pacing size )

          NetBIOS properties
          NETNAME=applid (Applid)
          ADAPTER=0|1|2|3 (Adapter)

          Named Pipe properties
          NAMEDPIPENAME = CICSCC|name (Named Pipe Name)
ENDSECTION
```

Workload Manager section

```
SECTION LOADMANAGER
        TYPE=1|2 (Round robin or Biasing)
        Timeout=nnnn (Region timeout)
        NOBALANCE=progname (when Manage workload for this program is not selected)
        BIAS=progname (when Manage workload for this program is selected)
ENDSECTION
```

Customizing the keyboard

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

The keyboard mapping file can be identified by:

- The /k option of the CICSTERM command, which identifies a keyboard mapping file with a particular terminal (see page “colorfile parameter” on page 104).
- The CICSKEY environment variable. For example:

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

  You can set this environment variable via the **System** icon in the Control Panel.

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

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

Keyboard mapping file syntax

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

Keyboard mapping file syntax

```
►►──BIND3270function──┬───────────┬──key──┬─────────────────────┬──────────────────►◄
                      └─modifier+─┘       └─┬─;─┬──comment──┘
                                            └─#─┘
```

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

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

The keyboard mapping file

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

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

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

The *Modifier* can be any one of:

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

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

Table 10. CICS Universal Client keys that can be mapped

Group	Keys
Escape key	Escape
Function keys	f1 f2 f3 f4 f5 f6 f7 f8 f9 f10 f11 f12

Table 10. CICS Universal Client keys that can be mapped (continued)

Group	Keys
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

In addition, the following key mappings are allowed:

```
bind Clear        Pause
bind Reset        Scroll Lock
```

Keys specific to particular keyboards

Some keys are specific to particular types of keyboard.

The following additional key is unique to IBM keyboards:

```
rightctrl
```

Key combinations

The following combinations of modifier and key can be mapped:

No modifier
All keys available for mapping.

Alt modifier
Only function keys, numeric keys, movement keys, and alphabetic keys can be mapped.

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.

Note: All modifier and key combinations that are not preempted by Windows can be mapped.

A sample key mapping file is shown in Figure 5.

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

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

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

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

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

Figure 5. Sample keyboard mapping file for CICS Universal Clients (Part 1 of 2)

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

Figure 5. Sample keyboard mapping file for CICS Universal Clients (Part 2 of 2)

Customizing the screen colors

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

The color mapping file can be identified by:

- The /c option of the CICSTERM command, which identifies a color mapping file with a particular server (see page “colorfile parameter” on page 104).
- The CICSCOL environment variable: For example:

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

You can set this environment variable via the **System** icon in the Control Panel.

If neither of these is 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

```
►►──BIND3270attribfg_color──┬───────────┬──┬─────────────────┬──────────────────────►◄
                            └─/bg_color─┘  └─┬─;─┬──comment─┘
                                             └─#─┘
```

The color mapping file

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

```
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 6 on page 81.

```
;***********************************************************************
;* 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 6. Sample color mapping file (Part 1 of 2)

```
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 6. Sample color mapping file (Part 2 of 2)

Testing the client/server link

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

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

If the connection to the server cannot be made, examine any messages and the client error log to determine the cause of the error (see “Chapter 9. Problem determination” on page 135).

Chapter 6. Operation

This chapter describes

- "An introduction to the CICS Universal Clients commands"
- "Running CICS Universal Client for Windows NT as a Windows NT service" on page 84
- "CICS Universal Clients icons" on page 87
- "The CICSCLI command" on page 88
- "The CICSTERM command" on page 101
- "The CICSPRNT command" on page 106
- "Font selection on terminal emulators" on page 111
- "CICS Client telnet terminal support" on page 112

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

An introduction to the CICS Universal Clients commands

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

- The CICSCLI command

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

  This command starts a terminal emulation session with particular options.
- The CICSPRNT command

  This command starts a printer terminal session with particular options.

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

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

© Copyright IBM Corp. 1994, 1999

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 97.

CICS Universal Client for Windows 98 operation

When you click the **Start** button and select **Programs**, you will see that the CICS Universal Client for Windows 98 folder contains icons for controlling client operation (see "CICS Universal Clients icons" on page 87). These icons are based on the CICSCLI, CICSTERM, and CICSPRNT commands, which you enter via the **Run...** command in the **Start** menu.

CICS Universal Client for Windows NT operation

The CICS Universal Client for Windows NT group window contains icons for controlling client operation. These icons are based on the CICSCLI, CICSTERM, and CICSPRNT commands, which you enter via the **Run...** command in the **Start** menu.

Running CICS Universal Client for Windows NT as a Windows NT service

You can run CICS Universal Client for Windows NT as a Windows NT service, if you select to register it as a service during installation. Running CICS Universal Client for Windows NT as a Windows NT service has several advantages:

- You can have the CICS Universal Client for Windows NT service started automatically at Windows NT startup, without having to log on to the computer.
- You can log off from Windows NT and keep CICS Universal Client for Windows NT running, thereby avoiding restarting the Client and its connection to CICS when you log on to Windows NT again.
- Relevant messages associated with running CICS Universal Client for Windows NT as a service are recorded in the Application log and System log of the Windows NT Event Viewer when the service is started and stopped.

CICS Universal Client for Windows NT is registered as a Windows NT service during installation. After installation, the Services dialog box in the Control Panel contains the **IBM CICS Client** service.

Service controls

You can use the Windows NT service API functions to pass service control requests to the **IBM CICS Client** service. Refer to your Windows NT documentation for further details about how to do this.

All the control requests that you can pass to the **IBM CICS Client** service are defined in the file cicsserv.h in the include subdirectory. In addition to these control requests, you can also pass to the service standard NT requests, for example, SERVICE_CONTROL_STOP, SERVICE_CONTROL_PAUSE, SERVICE_CONTROL_CONTINUE, and SERVICE_CONTROL_INTERROGATE.

The following are the control requests that you can pass to the **IBM CICS Client** service.

Control Request	Description
CICS_SRV_STOP	Stops the CICS Universal Client daemon. This also stops the service.
CICS_SRV_STOP_IMMEDIATE	Forces the CICS Universal Client daemon to terminate immediately. This also stops the service.
CICS_SRV_LIST	Lists the servers that the CICS Universal Client is currently connected to.
CICS_SRV_DISK_TRACE_ON	Starts tracing to a disk file. The name of this file is cicscli.bin, and it is written to the Client \BIN directory.
CICS_SRV_DISABLE_POPUP	Suppresses the client error and security popups.
CICS_SRV_ENABLE_POPUP	Activates the client error and security popups.
CICS_SRV_QUIET	Suppresses all CICS Client messages.

Service startup parameters

If you run the CICS Universal Client as a service, you can use the Windows NT service control panel to pass parameters to the service at startup. The following parameters are valid:

/S[=*server***]**
To start the client and optionally connect to a server.

/D[=*size***]**
To enable service tracing and optionally set a size limit.

/U[=*userid***]**
To set the userid to be used with a server.

/P[=*password***]**
To set the password to be used with a server.

/N To suppress the client error and security pop-ups.

/F=*inifile*
To specify a configuration file.

/T[=*size***]**
To enable service tracing to memory and optionally set a size limit.

/R[=*filename***]**
To redirect all client output to the specified file.

Note:

If you run CICS Universal Client as a Windows NT service, NPI will only work if you assign a logon user account to the service, rather than use the system account.

Starting and stopping the service

To start the IBM CICS Client service, select the service in the Services dialog box, and select **Start**. A message is displayed and the service is started.

To stop the IBM CICS Client service, select the service in the Services dialog box, and select **Stop**. A message is displayed and the service is stopped.

Startup parameters for the CICS services

You may choose to start the CICS Universal Client for Windows NT service automatically at Windows NT startup.

To specify automatic startup, select **Startup** in the Services dialog box. The Service panel is then displayed:

Note: You must have Administrator authority to configure service startup.

Select the **Automatic** radio button, and make sure that the **System Account** radio button is selected.

The Service panel also allows you to specify a Manual startup for the service, and to disable the service.

For more information on services and their configuration, refer to the Windows NT documentation.

CICS Universal Clients icons

All CICS Universal Clients are supplied with the following icons:

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

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

Stop Client
Stops the client.

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

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

This is equivalent to the command `CICSTERM`.

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

This is equivalent to the command `CICSPRNT`.

Client Status
Lists connected servers.

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

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

Customizing the command icons

You can customize your CICS Universal Clients installation to ensure that the command functions you use most frequently are associated with icons.

For example, you might want to create an icon for:

```
CICSCLI /s=servername /q /d=nnn /f=mycli.ini
```

to start a connection to a server used for testing applications.

Creating command icons

To create a command icon for CICS Universal Client for Windows 98 or CICS Universal Client for Windows NT, you can copy an existing program item and change its properties:

1. Open the CICS Universal Clients folder.
2. Copy a program item similar to the one you require, and rename it.
3. Select the new program item.
4. Select **Properties** from the **File** menu, and enter the required settings for the new icon in the Shortcut page.

The CICSCLI command

You use the CICSCLI command to:

- Start the client control process, and start communication with CICS servers (/s parameter)
- Stop the client control process (/i and /x parameters)
- Restart the client (/j and /y parameters)
- Specify the configuration file to use (/f parameter)
- Turn client trace on (/d parameter)
- Turn client trace off (/o parameter)
- Specify the client components to be traced (/m parameter)
- Set up security (/c, /u, and /p parameters)
- List connected CICS servers (/l parameters)
- Enable the display of pop-up messages (/e parameter)
- Disable the display of pop-up messages (/n parameter)

The following sections provide examples of using the CICSCLI command. Full details of the command syntax are given in “CICSCLI command reference” on page 97.

Starting the client control process

To start the client control process, enter:

```
CICSCLI /s
```

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

```
CICSCLI /s=servername
```

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

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

Starting connections with additional servers

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

```
CICSCLI /s=servername
```

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

Stopping the client control process

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

Stopping in a controlled manner

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

```
CICSCLI /x
```

To terminate the session with a particular server, enter:

```
CICSCLI /x=servername
```

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

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

Stopping immediately

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

```
CICSCLI /i
```

To terminate the session with a particular server, enter:

```
CICSCLI /i=servername
```

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

Restarting the client

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

Restarting in a controlled manner

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

```
CICSCLI /y
```

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

Restarting immediately

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

```
CICSCLI /j
```

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

Specifying which configuration file to use

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

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

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

Turning on client trace

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

```
CICSCLI /d=nnn
```

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

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

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

Specifying the trace components

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

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

which would cause the transport layer and full API trace information to be produced. For more information on which components you can trace, see "CICSCLI command reference" on page 97.

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 Network Provider Interface (NPI), if this is enabled, see "Network Provider Interface (NPI)" on page 94.

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

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

- In the **CICS_EpiAttributes_t** structure on a **CICS_EpiAddExTerminal** call.
- The EPI function **CICS_EpiSetSecurity**. Note that this function can also be used for terminals added using the **CICS_EpiAddTerminal** call.

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

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

Note: Userids and passwords must not contain DBCS characters.

CICSCLI security commands

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

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

```
CICSCLI /c=servername
```

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

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

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

where *userid* is the userid.

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

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

where *password* is the password.

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

Using the ESI to set security details

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

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

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

Security popups

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

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

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

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

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

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

ECI security

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

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

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

EPI terminal security

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

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

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

Terminal emulator (CICSTERM) security

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

Password expiry management

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

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

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

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

Network Provider Interface (NPI)

Network Provider Interface (NPI) allows you to sign on to a CICS server using the same user ID and password that you use to log onto Windows NT. The CICS server authenticates and authorizes using your domain information, which is stored encrypted in memory.

However, the security information will only be kept synchronized if you sign on to a CICS server that supports NPI, in which case the domain controller is used to authenticate the user ID.

When you install CICS Universal Client for Windows NT, the use of NPI is not enabled. You enable or disable the use of NPI for each CICS server with the **Use Windows credentials for security** configuration setting. To use Windows credentials for security you must also:

1. Change to the directory \Program Files\Common Files\IBM\IBM Integrated Sign ON
2. Enter `bthdinst -i CICSCLI`
3. Reboot.

When uninstalling you **must** uninstall in the following sequence:

1. Change to the directory \Program Files\Common Files\IBM\IBM Integrated Sign ON
2. Enter `bthdinst -u CICSCLI`
3. Reboot
4. Uninstall in the usual way.

If NPI is enabled, but the user ID and password cannot be obtained when signing onto the CICS server, a popup is displayed to prompt for the information.

If you run CICS Universal Client as a Windows NT service, NPI will only work if you assign a logon user account to the service, rather than use the system account.

Restrictions: Windows NT allows the use of longer user IDs and passwords than those allowed by CICS servers. Therefore, if you wish to use NPI, you may need to use a shorter user ID and password to log onto Windows NT.

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 'CICSOS2' (using 'TCP/IP' to 'CICSOS2') is available
CCL8043I Server 'CICSNT' (using 'NETBIOS' to 'CICSNT') is unavailable
```

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

Disabling the display of messages

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

```
CICSCLI /s /q
```

Enabling and disabling the display of pop-up messages

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

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

```
CICSCLI /n
```

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

```
CICSCLI /e
```

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

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

Displaying the command parameters

To display the parameters of the CICSCLI command, enter:

```
CICSCLI /?
```

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

To conform with other popular syntax conventions, this slash character may be replaced with a minus sign (-). The commands and options are not case-sensitive.

All options of the form `/x=variable` may contain spaces in the variable part, if it is enclosed in double quotes. Double quotes within variables must be entered as \″ , that is with a backslash preceding the double quote.

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

CICSCLI Command Syntax

```
►►─CICSCLI──┬────┬─/s─────────────┬───┬──────────────┬───┬─────────┬───┬────────────────┬───┬──────┬──┬──►
            │    └─/s=servername──┘   └─/f=inifile───┘   ├─/d──────┤   ├─/m─────────────┤   └─/n───┘  │
            │                                            └─/d=nnn──┘   └─/m=components──┘             │
            ├─/o──────────────────────────────────────────────────────────────────────────────────────┤
            ├─┬─/d──────┬─────────────────────────────────────────────────────────────────────────────┤
            │ └─/d=nnn──┘                                                                             │
            ├─┬─/m─────────────┬──────────────────────────────────────────────────────────────────────┤
            │ └─/m=components──┘                                                                      │
            ├─┬─/x─────────────┬──────────────────────────────────────────────────────────────────────┤
            │ └─/x=servername──┘                                                                      │
            ├─┬─/i─────────────┬──────────────────────────────────────────────────────────────────────┤
            │ └─/i=servername──┘                                                                      │
            ├─/l──────────────────────────────────────────────────────────────────────────────────────┤
            ├─/j──────────────────────────────────────────────────────────────────────────────────────┤
            ├─/y──────────────────────────────────────────────────────────────────────────────────────┤
            ├─/c=servername──┬─────────────┬──┬───────────────┬───────────────────────────────────────┤
            │                ├─/u=userid───┤  ├─/p=password───┤                                       │
            │                └─/u──────────┘  └─/p────────────┘                                       │
            ├─/e──────────────────────────────────────────────────────────────────────────────────────┤
            ├─/n──────────────────────────────────────────────────────────────────────────────────────┤
            └─/?──────────────────────────────────────────────────────────────────────────────────────┘

►──┬───────┬─────────────────────────────────────────────────────────────────────────────────────────►◄
   ├─/q────┤
   └─/w────┘
```

The options are:

/c=*servername*
: Identifies the name of the server to which security information in the form of a userid and password is to be associated. Some CICS servers require that the user provide security information to the server before interacting with the server. The CICS Universal Client prompts the user at the workstation for a userid and password, unless these have already been provided via CICSCLI (see the descriptions of the /u and /p options).

/d=[*nnn*]
Turns the debug trace on for the client. If tracing is required while the client is starting up, this option may be specified along with the /s option.

nnn is the maximum size of data areas to be traced in bytes. The range is 1 through 32 767 bytes, and the default value is 512 bytes.

Output from the trace option is directed to the cicscli.bin file in the Client \BIN directory, which can then be formatted with the CICSFTRC utility. The resultant trace file, cicscli.trc, is an ASCII file that can be read with a text editor. For more information see "Formatting the binary trace file" on page 144.

/e
Enables the display of client error and security messages in pop-up windows.

/f=*inifile*
Specifies the name of the configuration file. The default is ctg.ini. If the file is not in the Client \BIN directory, you must specify the full path name.

Note: If the file resides on a redirected network drive then problems may occur if the network drive is later detached.

/i
Stops the client immediately. The options /i=*servername* and /i operate as for /x=*servername* and /x respectively *but* the client does not wait for outstanding units of work to complete. Stopping the client in this way may result in a loss of data at connected servers.

/j
Stops the client immediately and then restarts the client.

A client restart involves shutting down the client, waiting for the client to shut down, and then starting up again. CICSCLI /j is equivalent to CICSCLI /i followed by CICSCLI /s. Server connections are not re-established when the client is restarted.

Using /y is the preferred way of restarting the client.

/l
Causes a list of all connected servers to be displayed. For each server, the netname of the server as it is known to the client is also displayed, as well as the state of the connection to the server and the connection protocol. The data is displayed in a window.

/n
Disables the display of client error and security messages in pop-up windows.

Any messages that would have been logged are still logged.

/m=[*components*]

Specifies a comma-separated list of identifiers for components that will be traced when tracing is turned on. You can specify any of the following components:

ALL	All components
API.1	The client API layer (level 1)
API.2	The client API layer (level 1 and 2)
API	Synonymous with API.1
CCL	CICS client daemon
CCP	The C++ class libraries
CLI	The CICSCLI command interface
DEF	The default components, that is the API, CCL, and DRV components
DRV	Protocol drivers (TCP, SNA, and so on)
EMU	CICSTERM and CICSPRNT emulators
LMG	The Workload Manager
RXX	REXX API
SER	CICS Client Windows NT service
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 48). Any component tracing specified using CICSCLI overrides that specified using the configuration tool. If component tracing is not specified either by the CICSCLI command or by using the configuration tool, a default set of components is traced, namely: DRV, CCL, and API. Any component tracing specified using the configuration tool overrides the default set of components.

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

Note that the 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. (Or by the /s parameter.) This password is used if the server requires a password (and userid) before running transactions on the client's behalf.

For ECI applications, any userid and password specified in the ECI parameter block overrides values set via the 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. (Or by the /s parameter.) This userid is used if the server requires a userid (and password) before running transactions on the client's behalf.

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

Specifying /u or /u= (that is, no userid is specified) resets the associated userid to a null value.

/w Prompts the user, before the command completes, to press the Enter key, to confirm that messages output to the screen (both informational and error) have been read.

/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.

The CICSTERM command detects whether the hardware on which the client is running is enabled for double-byte character set (DBCS) display. If it is, the emulator can display DBCS characters.

For CICSTERM emulators, the maximum screen size is 27 rows by 132 columns. This is because CICS Universal Clients support the ASCII-7 subset of the 3270 data stream architecture, which only supports 12 bit addressing. For more information, see *CICS Family: Client/Server Programming*.

Note: Some uses of servers and protocols require a model terminal definition for the emulator that explicitly specifies that the client wants to display DBCS.

Using CICSTERM

You use the CICSTERM command to:

- Start a 3270 terminal emulator (/s or /r parameter)
- Specify the initial transaction (/t parameter)
- Specify the name of the keyboard mapping file (/k parameter)
- Specify the name of the color mapping file (/c parameter)
- Define the 3270 terminal emulator characteristics (/n and /m parameters)

- Specify whether a terminal emulator is signon capable (/a parameter)
- Determine the print file processing (/p parameter)
- Specify a file to which print files are appended (/f parameter)
- Specify a URL for the configuration file used by the Workload Manager.

You issue the CICSTERM command *once* with all the parameters you require.

The following is an example of a CICSTERM command:

```
CICSTERM /s=CICSOS2 /t=CESN /k=mykeys.ini /c=mycols.ini
         /n=cicsv123 /f=clprint.txt /q
```

In this example:

/s=CICSOS2
Specifies that a 3270 terminal emulator is started for the server CICSOS2.

/t=CESN
Specifies that the initial transaction is CESN.

/k=mykeys.ini
Specifies that the keyboard mapping file is named as mykeys.ini.

/c=mycols.ini
Specifies that the color mapping file is named as mycols.ini.

/n=cicsv123
Specifies that the 3270 terminal emulator characteristics are defined by the terminal definition cicsv123.

/f=clprint.txt
Specifies that the print file will be appended to the file clprint.txt.

/q Specifies that the display of messages output by the command is disabled.

All parameters of CICSTERM are optional. That is, you can enter the CICSTERM command without any parameters, and defaults are taken from the configuration file. (This is equivalent to double-clicking on the CICS Terminal icon.) Full details of the parameters are given in “CICSTERM command reference” on page 103.

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 CICSPRNT. (It is however, supported by the EPI).

Note: Issuing these EXEC CICS commands from CICSTERM does not result in the StartTranExit or ReplyExit user exits being driven. The EPI user exits and how CICSTERM can use them are described in the *CICS Family: Client/Server Programming* book.

CICSTERM command reference

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

CICSTERM Command Syntax

```
►►──CICSTERM──┬─────────────────────┬──┬──────────────────┬─────────────────────────────►
              ├─/s──=──servername───┤  ├─/t=initialtransid─┤
              ├─/s──────────────────┤  └─/t────────────────┘
              ├─/r=servername───────┤
              └─/r──────────────────┘

►──┬──────────────┬───┬──────────────┬───┬──────────────┬───┬────┬──────────────────────►
   └─/k=keyfile───┘   ├─/c=colorfile─┤   ├─/m=modelname─┤   └─/a─┘
                      └─/c───────────┘   └─/m───────────┘

►──┬──────────────┬───┬──────────────┬───┬────┬───┬────┬───────────────────────────────►◄
   └─/n=netname───┘   ├─/p=printcmd──┤   ├─/q─┤   └─/?─┘
                      └─/f=printfile─┘   └─/w─┘
```

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 78) to be used with the emulator. If this parameter is omitted, the environment variable CICSCOL is assumed to identify the color mapping file. If CICSCOL is not defined, a filename of cicscol.ini in the Client \BIN directory is assumed.

If the parameter is specified as /c=, (that is, the color mapping filename is omitted), the emulator runs without any color definitions.

/f=*printfile*
Specifies the name of a file to which the output of print requests is appended. If the name of the file contains embedded blanks, it must be surrounded by double quotes (“). Any double quotes within the name of the file must be entered as backslash double quote (\“).

If neither of the /f or /p parameters is provided, the **Print command** or **Print file** configuration settings define the command, file, or default action to take with print requests.

/k=*keyfile*
Identifies the name of a keyboard mapping file (see “Keyboard mapping file syntax” on page 74) to be used with the emulator. If this parameter is omitted, the environment variable CICSKEY is assumed to identify the key mapping file. If CICSKEY is not defined, a filename of cicskey.ini in the Client \BIN directory is assumed.

/m=*modelname*
Specifies the name of a model terminal definition, as known at the server to which the emulator is to connect, to be used to define the terminal characteristics. If neither this parameter nor /n=netname is specified, any Model terminal definition value from the configuration file is used. If no Model terminal definition value has been specified in the configuration file, the server’s default terminal definition is assumed.

If the parameter is specified as /m= (that is, the modelname is omitted), any Model terminal definition value specified in the configuration file is ignored, and the server’s default terminal definition is assumed.

This option is case-sensitive.

/n=*netname*

Specifies the name of a particular terminal definition at the server that this emulator is to be installed as. The precise interpretation of netname varies between servers, for example, on CICS for OS/2 it references a termid defined in the CICS tables, on TXSeries for AIX it is a netname.

This option is case-sensitive.

/p=*printcmd*

Specifies an operating system command used to process the temporary print file generated when print requests are received by the terminal emulator. If the command contains embedded blanks, then the command must be surrounded by double quotes (“). Any double quotes within the command must be entered as backslash double quote (\“).

| If neither of the /f or /p parameters is specified, the **Print command** or **Print file** setting in the configuration file defines the command, file, or default action to take with print requests.

The temporary print file is post-processed by appending the filename to the command, and executing the resultant command. Thus print output may simply be copied to a local printer, copied into a permanent file, processed further for inclusion into a document, and so on. If the temporary file is to be processed by a print command, the command is responsible for deleting the temporary file.

/q Disables the display of all messages output by the command.

/s=*servername* **or /r=***servername*

Specifies the name of the server that the terminal emulator is to be connected to. This servername must correspond to an entry in the configuration file. You can specify /s, or /r, but not both.

If neither parameter is specified, the first server entry in the configuration file is used.

If the parameter is specified as /s or /r (that is, no servername is provided) then, if the configuration file identifies more than one potential server to which the client can connect, the user is prompted to select from a list of available servers. These prompts are generated even if the /q parameter is specified.

If there is only one potential server identified in the configuration file, that server is used and the user is not prompted.

/t=*initialtransid*

Identifies the initial transaction to be invoked for this terminal. If this option is omitted, any initial transaction specified in the configuration file is run. The string may be up to 128 characters long, specifying both a transaction name, and parameters to be passed to the transaction. The transaction name is the first four characters or the characters up to the first blank in the string. The rest of the string is the parameter data.

If the parameter is specified as /t= (that is, the initialtransid is omitted), any initial transaction specified in the configuration file is ignored.

This option is case-sensitive.

Note: Be careful that transactions that you specify either here or in the configuration file do not require terminal input to complete.

/w Prompts the user, before the command completes, to press the Enter key, to confirm that messages output to the screen (both informational and error) have been read.

/? Causes the parameter syntax to be listed; any other options specified are ignored.

The CICSPRNT command

The CICSPRNT command is used to control 3270 printer terminal emulation.

Applications running on the server can direct output to a printer in one of two ways:

1. An application running from a terminal can initiate printing by sending a map or data with the PRINT indicator set
2. The user can start at the client a 3270 Print Terminal Emulator using the CICSPRNT command. A 3270 Print Terminal Emulator must be started for a netname or model terminal definition (or termid in the case of CICS for OS/2) predefined in the server’s terminal tables. Output is directed to such a device by starting a transaction against the printer device.

For CICSPRNT emulators, the maximum screen size is 27 rows by 132 columns. This is because CICS Universal Clients support the ASCII-7 subset of the 3270 data stream architecture, which only supports 12 bit addressing. For more information, see *CICS Family: Client/Server Programming*.

Note: At client workstations you can use the PrintScreen key (as defined by the keyboard mapping file). Note however, that any lines that contain only null characters are not printed. For a ‘blank’ line to be printed, it must contain at least one space character.

Using CICSPRNT

You use the CICSPRNT command to:
- Start a 3270 print terminal emulator (/s or /r parameter)
- Specify the initial transaction (/t parameter)
- Define the 3270 printer terminal emulator characteristics (/n and /m parameters)
- Determine the print file processing (/p parameter)
- Specify a file to which print files are appended (/f parameter)

You issue the CICSPRNT command once with all the parameters you require.

The following is an example of a CICSPRNT command:

```
CICSPRNT /s=CICSOS2 /n=P123 /t=XPRT /f=clprint.txt /q
```

In this example:

/s=CICSOS2
Specifies that a 3270 print terminal emulator is started for the server CICSOS2.

/n=P123
Specifies that the 3270 print terminal emulator characteristics are defined by the terminal definition v123 (in the terminal control table for CICS for OS/2 in this case.)

/t=XPRT
Specifies that the initial transaction is XPRT.

/f=clprint.txt
Specifies that the print file to which print requests are appended is clprint.txt.

/q Specifies that the display of messages output by the command is disabled.

All parameters of CICSPRNT are optional, except that you must specify either /n=*netname* or /m=*modelname*. That is, you can enter the CICSPRNT command with just the /n or the /m parameter, or both, and defaults for other parameters are taken from the configuration file. Full details of the parameters are given in “CICSPRNT command reference” on page 108.

If the system upon which the client is executing supports DBCS, it is assumed that the printer attached to the processor also supports DBCS. Conversely, if the system does not support DBCS, the client will not send DBCS data to the printer.

CICS Printer icon

For CICS Universal Clients, you should edit the settings for the icon to change the parameters according to your usage.

The CICSPRNT process runs as a minimized window, and the window can be enlarged to view the current status of the printer. You can use an action available from a pulldown menu to terminate the print function.

CICSPRNT and user exits

CICSPRNT can be used to drive EPI user exits.

The EPI user exits and how CICSPRNT can use them are described in the *CICS Family: Client/Server Programming* book.

CICSPRNT and RETURN TRANSID IMMEDIATE

Unlike CICSTERM:, CICSPRNT does not support the commands:

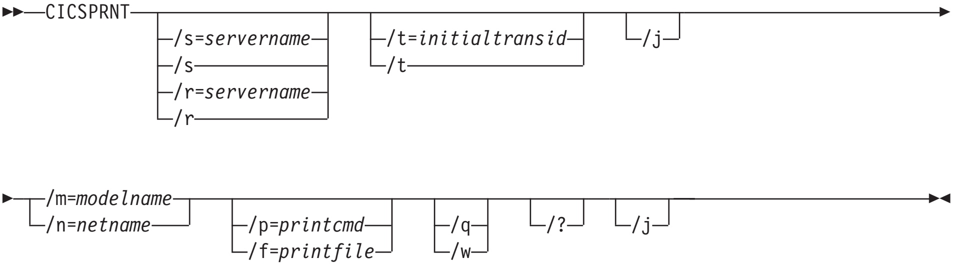
```
EXEC CICS RETURN TRANSID(name) IMMEDIATE
EXEC CICS RETURN TRANSID(name) IMMEDIATE INPUTMESSAGE(data-area)
```

For more information, refer to “CICSTERM and RETURN TRANSID IMMEDIATE” on page 103.

CICSPRNT command reference

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

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 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.

/w Prompts the user, before the command completes, to press the Enter key, to confirm that messages output to the screen (both informational and error) have been read.

/? Causes the parameter syntax to be listed; any other options specified are ignored.

Font selection on terminal emulators

CICS Universal Client for Windows 98, and CICS Universal Client for Windows NT provide a means of selecting and sizing fonts on their terminal emulators.

The terminal emulators provide a menu bar with **File** and **Settings** menus.

The commands on the **File** menu are:

Print Prints the emulator screen. This has the same effect as pressing the Print Screen key.

Exit Stops the terminal emulator.

The commands on the **Settings** menu are:

Font Opens a standard Windows dialog box that allows you to select any of the fixed-pitch fonts installed on your system. Select a font, font style, and size, and then select **OK**. The terminal window is then resized according to your font selection.

Autosize Specifies that the size of TrueType fonts is adjusted according to the size of the terminal window. That is, when you maximize, minimize, or drag the borders to resize the window, the font is resized to match the new window size. This command has no effect if the current font is not a TrueType font.

Save on Exit Specifies that the current font and terminal window position are saved when you close the terminal window. When you select **Save on Exit** the settings are saved in the CICSTERM.INI file, in the following directory:

Client platform	**Directory**
Windows NT	c:\winnt
Windows 95/98	c:\windows

The CICSTERM.INI file is not erased when you delete a CICS Client installation. Therefore, when you re-install the Client, the settings from the old CICSTERM.INI file are used by the emulator.

CICSTERM.INI is initially installed in the bin subdirectory where CICS Universal Client is installed.

CICS Client telnet terminal support

Support is provided for multiple tn3270 terminals to connect to CICS servers via a single CICS Universal Client. This function is available for all CICS servers that support the EPI.

You must first start a CICS telnet daemon by using the following command:

```
CICSTELD [/p=port] [/s=server] [/c=client codepage] [/l=server codepage]
         [/t=tranid] [/m=model]
```

where:

/p=*port*
Specifies the numeric Transmission Control Protocol (TCP) port number that CICSTELD listens on for incoming telnet client requests. If the port number is not valid, CICSTELD issues an explanation message. If you do not specify a port number, CICSTELD assumes the default CICS port number of 1436.

/s=*server*
Specifies the applid of the CICS server.

/c=*client codepage*
Specifies the codepage of the tn3270 client.

The default codepage is 037 (EBCDIC US English).

/l=*server codepage*
Specifies the codepage that is used by the CICS server’s EPI data stream.

The default codepage is 437 (ASCII US English).

/t=*tranid*
Specifies an initial transaction.

/m=*model*
Specifies a model terminal name on the CICS server.

/? Causes the parameter syntax to be listed; any other options specified are ignored.

When you have started the CICS telnet daemon, tn3270 terminals can then connect to the CICS server host.

You can use any Telnet 3270 client to access CICS through the CICS telnet daemon. To connect a tn3270 terminal, you can use tn3270 (or an equivalent program, if tn3270 is not installed by default on your system). For example:

```
tn3270 hostname -p [port] -ext
```

where *hostname* is the hostname of the machine running the telnet Daemon, and *port* is the port number of the telnet daemon, if you are not using the default CICS port number. The `-ext` parameter enables extended datastream support. If extended datastream is not enabled you will get a blank screen. For more information, refer to the tn3270 documentation.

To stop the daemon, press the `q` key.

You may experience problems if the number of Telnet 3270 clients connected to CICSTELD exceeds the MAXTASKS setting of the CICS server. To prevent this situation, use the CICSTELD parameter /n to limit the number of connections to CICSTELD. For example, CICSTELD /n=10 will allow no more than 10 Telnet 3270 connections to CICSTELD. There is no default limit on the number of CICSTELD users.

CICSTELD works only with models that use the standard terminal size of 24x80. If you use a model (specified with the /m parameter), you must ensure that it is of the correct terminal size.

Chapter 7. CICS Universal Client support for DCE

This chapter contains the following sections

- "CICS Universal Clients support for DCE"
- "Functions provided by DCE" on page 116
- "Enabling CDS support in CICS Universal Clients" on page 116
- "Configuring CICS Universal Clients for DCE RPC communication" on page 118
- "Coordinated security sign-on" on page 118
- "Configuring DCE on the client workstation" on page 121
- "Configuring the CICS server" on page 122
- "Configuring DCE RPC communication— a summary" on page 123

This chapter describes the support for Distributed Computing Environment (DCE) provided by CICS Universal Clients, and how to configure CICS Universal Clients to use the DCE Remote Procedure Call (RPC) protocol to communicate with CICS servers.

It is assumed that you are familiar with the principles of DCE. For general information about DCE, refer to the publication: *OSF Distributed Computing Environment: Understanding DCE* by Ward Rosenberry, David Kenny, and Gerry Kisher (published by O'Reilly and Associates, Inc., ISBN 1-56592-005-8).

CICS Universal Clients support for DCE

CICS Universal Clients support the following types of DCE environment:
- Gradient PC-DCE for Windows NT/95/98 Version 2.2
- IBM DCE for Windows NT Version 1.1.1 or Version 2.0, or later

The DCE RPC support provided by CICS Universal Clients can only be used to communicate with CICS servers that support the same DCE functionality, for example, TXSeries Version 4.2 for Windows NT, TXSeries Version 4.2 for AIX.

© Copyright IBM Corp. 1994, 1999

Functions provided by DCE

Use of DCE provides the following functions not otherwise available with CICS Universal Clients:

1. Cell Directory Service (CDS).

   This enables the CICS Universal Client to obtain its list of available CICS servers from the DCE Cell Directory, rather than relying on the configuration file (ctg.ini).
2. Coordinated security sign-on.

   The user can sign on to DCE and be verified on their local machine. This sign-on context (the DCE principal) can then be propagated to the CICS server, avoiding the need to pass an explicit user ID and password, and avoiding the need to sign on again at the server.
3. Data encryption.

   DCE provides the ability to encrypt the data flows between the client and server in a manner transparent to the application.

Enabling CDS support in CICS Universal Clients

The CICS servers available for RPC communication can be located either via the CDS or directly by use of the configuration file.

You enable the use of the CDS by selecting the **Use DCE CDS** check box, in the configuration tool. The CDS services are then used, provided DCE is installed and configured on the client machine.

If you enable the use of the CDS, information the client needs to connect to the server is automatically obtained via the CDS. Otherwise, you must make appropriate settings for each CICS server in your client configuration, see “Configuring CICS Universal Clients for DCE RPC communication” on page 118. In any case, you must select which type of DCE you want to use by selecting the **IBM DCE** or **Gradient PC-DCE** check box in the configuration tool.

The option to enable or disable the use of the CDS is provided because searching the CDS for CICS servers may take some time, particularly with a large DCE cell.

Servers located via the CDS

Using the CDS to locate CICS servers is equivalent to providing the following **Server** settings in your configuration (see “Configuring CICS Universal Clients for DCE RPC communication” on page 118.

Server name An 8-character region name from the cell directory.

Description A description obtained from the cell directory annotation.

Use upper case security
Not selected. Note that on AIX user IDs and passwords (if required) are always case-sensitive.

DCE Selected.

DCE Product All servers located via the CDS use an RPC communications driver dynamic link library (DLL):

CCLGRARP	If you select Gradient PC-DCE.
CCLIBMRP	If you select IBM DCE.

These driver names are stored in the DRIVER section in the configuration file.

The following **Server** settings are not available for servers located via the CDS:

Initial transaction
The default initial transaction for CICSTERM and CICSPRNT is not available. If an initial transaction is required, you must specify it with the `/t=transid` parameter on the CICSTERM or CICSPRNT command.

Model terminal definition
The default model terminal for CICSTERM and CICSPRNT is not available. If a specific model is required, you must specify it with the `/m=model` parameter on the CICSTERM or CICSPRNT command.

For more information on configuration settings, refer to “Using the configuration tool” on page 48.

List systems API calls

CICS Universal Clients provide two application programming interface (API) calls for listing available servers.

When a user application issues a `CICS_EciListSystems` or `CICS_EpiListSystems` call, the list of available CICS servers returned includes all the servers listed in **Server** sections of the configuration file plus any servers that can be located using the CDS in the local DCE cell (assuming the CDS is enabled).

If only CDS-located servers are required, you do not have to configure any **Server** settings using the configuration tool.

Because the servers defined in the CDS can change as new servers are started, or existing servers are shut down, the client must query the CDS each time a `CICS_EciListSystems` or `CICS_EpiListSystems` call is made.

For more information on the API calls, refer to the *CICS Family: Client/Server Programming* book.

CICSTERM and CICSPRNT commands

When you issue the CICSTERM and CICSPRNT commands with the /s or /r parameter, they obtain their server lists via the `CICS_EpiListSystems` call and therefore automatically list the servers in the configuration file, as well as those additional servers located via the CDS.

Configuring CICS Universal Clients for DCE RPC communication

If you do not use the CDS to locate the CICS servers, you must make DCE protocol settings in your configuration as follows:

DCE	You must select this from the **Network Protocol** list.
CICS region	A Region name on the CICS server.
DCE Product	You must select either **IBM DCE** or **Gradient PC-DCE**.

Other settings such as **Model terminal definition** or **Initial transaction** are optional and can be included if required.

Coordinated security sign-on

Using DCE RPC as the communication protocol between CICS Universal Client and CICS server allows “coordinated security sign-on”. This allows the user or client application to perform a single DCE login on the CICS Universal Client system, and for that login to be propagated to the CICS server where it can be used to authorize access to secure CICS resources. A separate CICS user ID and password is not required, nor is there any need to sign on to the CICS system.

DCE RPCs can be defined to run *unauthenticated* or *authenticated*; although the coordinated security sign-on is only available when using authenticated RPCs.

To use authenticated RPCs requires suitable definitions to be made to the CICS server and requires the CICS Universal Client to be correctly configured.

Unauthenticated RPC

If the CICS Universal Client has not been configured for authenticated RPC (see “Configuring the CICS Universal Client for authentication”), or the server definitions specify no security, all RPC communications between the client and server run unauthenticated.

When running unauthenticated, the CICS server may not know and certainly cannot trust the identification of the client; for this reason the coordinated security sign-on is not possible when using unauthenticated RPCs. In this situation access to secure CICS resources requires a separate CICS sign-on in the normal manner, using a valid CICS user ID and password.

This is identical to the security required when connecting the client and server via TCP/IP.

Authenticated RPC

If the server definitions specify a suitable level of RPC security, and the client has been correctly configured (see “Configuring the CICS Universal Client for authentication”), all RPC communications between the client and server will run authenticated.

When running authenticated, the CICS server can obtain the DCE login principal of the originating client user or client application, in addition it can trust this principal and knows it can only have come from a valid DCE login on a valid CICS Universal Client system.

In this situation the server can map the DCE principal to a corresponding CICS user ID and automatically sign on the CICS user. No separate CICS user ID and password are required from the user and no CICS sign-on step (using the CESN transaction, for example) is needed.

Configuring the CICS Universal Client for authentication

Before the CICS Universal Client can participate in authenticated RPC with a server it must be correctly configured. If this configuration is not done, the client can only make unauthenticated RPCs and cannot therefore use the coordinated sign-on features.

To configure the client for authenticated RPC, run the CCLAUTH utility program, provided with the CICS Universal Client.

```
CCLAUTH /U=adminuserid /P=adminpasswd [/Q] [/W]
```

You must supply the CCLAUTH program with the DCE cell administrator user ID and password using the /U= and /P= parameters, because it needs to update the DCE registry with details of the client system.

The CCLAUTH program need only be run once on each CICS client workstation, although it is harmless if run multiple times.

The CCLAUTH command creates a unique DCE principal and key for the client machine that it is run on. A DCE principal is analogous to a userid and is used internally by the CICS Server and the Client for authenticated DCE communication The key is stored on the hard disk of the client machine and is used to generate security tokens.

Obtaining the DCE principal

To enable a client user or client application to change the DCE login if it receives or detects a security error, the DCE principal that is propagated to the CICS server on the authenticated RPC can be obtained at the following times:

- For **CICSTERM** and **CICSPRNT**, when the emulator starts, and before it sends its terminal install request to the server.

  You must perform a dce_login command to log in to DCE before starting the emulator.
- For **external call interface (ECI)** applications, at the start of each new logical unit of work.

  An ECI application can use the DCE sec_login APIs to change the current principal, if it is necessary.
- For **external presentation interface (EPI)** applications, at each call to CICS_EpiAddTerminal, at each call to CICS_EpiStartTran, and before starting any ATI transactions.

  An EPI application can use the DCE sec_login API calls to change the current principal, if it is necessary.

For more information on the ECI and EPI, refer to the *CICS Family: Client/Server Programming* book.

Data encryption

When you use an RPC protocol between the client and server, data encryption is handled entirely by DCE, although the function may not be available for the all levels of DCE on all platforms.

Configuring DCE on the client workstation

This section describes what you need to do to configure DCE on your CICS Universal Clients workstation. Procedures are given for both IBM DCE and Gradient PC-DCE.

Configuring IBM DCE

To configure IBM DCE on the client workstation:

1. From the Start Program menu, select **DCE for Windows NT**, then **DCEsetup**.
2. Select the **Configuration** pull-down, **Create**, then **Full Client** to display the Full Client Configuration Wizard panel.
3. On the first panel:
   a. In **Host name**, enter the host name of your workstation, and ensure that this is not a fully-qualified name. For example, enter `charlie`, rather than `charlie.hursley.ibm.com`.
   b. In **Cell name**, enter the fully-qualified name of the cell in which the CICS Server exists. You can obtain this name by starting a Telnet session to the host name of the DCE cell and entering the `getcellname` command.
   c. In **Security Master Server Host Name**, enter an unqualified name, and enter the Administrator password.
   d. Select **CDS Server is within Broadcast Range**, if applicable. If you do not select this checkbox, enter an unqualified name in **CDS Server Host Name**
   e. Select **Distributed Time Service** and the **Select DTS client** button.
   f. Select **Next** to proceed.
4. On the second panel:
   a. In **Security Master Server Host IP Address**, enter the appropriate IP address
   b. In **CDS Server Host IP Address**, enter the appropriate IP address
   c. Select **Automatically sync local time with cell time**, and enter the fully-qualified host name of the DTS Server with which the client is to synchronize. (Ensure the client workstation and the DTS server are showing the same time/zone).

   d. Select **Integrated Login**.
   e. Select **Automatically start DCE services during system startup**.
   f. Select **Next** to display a summary of your settings.
   g. Select **Finish** and DCEsetup registers the client with the DCE cell directory.
5. Enter the Administrator principal password when prompted and select **OK** to start the configuration.

Configuring Gradient PC-DCE

To configure Gradient PC DCE on the CICS Universal Client for Windows NT workstation:

1. Open the Settings panel and select the **PC-DCE** icon.
2. Select the **configure** option.
3. In the DCE client panel:
   a. Enter the fully-qualified name of the CDS server cell in which the CICS Server exists. (To obtain the fully qualified host name, enter the `getcellname` command on the CDS server.)
   b. Select the **Configure clients daemons** checkbox.
   c. Enter the Administrator principal name and password.
4. In the Options panel:
   a. Select **Start daemons during system boot** (optional)
   b. Select **Distributed Time Service.**
5. In the More servers panel:
   a. Select the **DTS client** checkbox
   b. Enter the Cell Administrator password
6. Return to the DCE client panel and select **OK** to complete the configuration.

Configuring the CICS server

When CICS Universal Clients connect to CICS servers using DCE RPC, the CICS server determines which level of DCE authentication should be used.

You should set the following for authenticated RPC:

- In the CICS Server Resource Definition (RD.stanza) ensure that:
  – AuthenticationService = DCE
  – Region Protection Level > none (that is, whatever DCE protection level you require)
  – Terminal Protection Level > none

For TxSeries for AIX, you can set these via the smitty region definitions panel.

- A DCE userid has to be set up in the DCE cell. This is done automatically via the CICS smitty panel, which deals with creating userids.

Configuring DCE RPC communication— a summary

This section summarizes what you must do to configure CICS Universal Clients and CICS servers for DCE RPC communication, and provides references to further information.

The task that you must perform are:

- "Configuring DCE on the client workstation" on page 121
- Configuring your CICS Universal Client, using CDS or your configuration settings to locate CICS servers.
- "Configuring the CICS server" on page 122.
- "Configuring the CICS Universal Client for authentication" on page 119.

The *Revealed! CICS Transaction Gateway with more CICS Clients Unmasked* book gives a sample configuration and detailed information about the attendant security implications.

There may also be a sample configuration document available, see "Sample configuration documents" on page 169.

Troubleshooting

For information on problems that you might have with DCE RPC communication, and how to deal with them, see "DCE RPC communication problems" on page 156.

Chapter 8. Workload Manager

This chapter discusses how to use the Workload Manager of CICS Universal Client for Windows 98 and CICS Universal Client for Windows NT to improve the performance of the CICS Clients.

The Workload Manager is invoked during ECI and EPI calls from the client, and implements ECI and EPI user exit functions. These ECI and EPI exit functions are accessed during processing at a number of points during the execution of a request. They are described in the *CICS Family: Client/Server Programming* book.

Instead of using the supplied Workload Manager you can write your own user exits that implement the ECI and EPI functions, and these exits can be driven by the CICSTERM and CICSPRNT commands. Information on producing such exits is given in the *CICS Family: Client/Server Programming* book.

When the Workload Manager is activated for a CICSTERM / CICSPRNT command, it takes control of the server to which the client is connected. The Workload Manager server selection overrides any server that you may have specified on the invocation of the CICSTERM / CICSPRNT command.

Workload Manager overview

The Workload Manager is designed to handle various problems when CICS Universal Clients are used to connect with one or more CICS servers (CICS regions):

- The restriction on the number of clients that a single CICS server can accommodate on a particular port.
- How to handle load balancing of work across available, equivalent CICS regions.
- How to handle biasing of work in favor of one region over another.
- How to handle requests that fail to reach a target CICS region.

The Workload Manager resolves these problems and others by dynamically selecting target regions and ports based upon configurable criteria.

© Copyright IBM Corp. 1994, 1999

Key concepts

It is essential to understand the following concepts:

Regions
A region is a physical instance of a CICS server, for example, TXSeries for Windows NT or AIX. This includes the programs and resource definitions for that region.

Programs
CICS programs are executable units that can be started within a CICS region. A particular CICS program may be available on multiple CICS regions. Hence, the Workload Manager has the concept of program affinity, meaning it has a list of CICS regions on which the program is defined.

TCP/IP address
An address that uniquely identifies a machine on a particular network. Each machine may have many threads of execution running simultaneously; a TCP/IP port uniquely identifies a particular thread of execution.

TCP/IP port
A numeric TCP/IP end-point in the range of 0-65535. This port number is defined in the configuration file (ctg.ini) file for each instance of a server definition.

The number of client requests that can be handled on a particular port varies with the CICS server platform. For example, a TXSeries for AIX or TXSeries for Windows NT may handle client requests via a number of ports, but only about 250 client requests can be handled per port. It may therefore be necessary to have multiple server instances defined. On the other hand, CICS for OS/2 regions only use one port per region. but each region may have about 2000 clients connected.

Server instance
An entry in the CICS client configuration that corresponds to an instance of a CICS server. For a particular CICS region there may be multiple instances of CICS server definitions, each with different port numbers. The names provided by the **Server name** configuration setting (Server=*servername* in the configuration file) are used in the actual ECI requests. In the Workload Manager section of the configuration tool, the server instances are gathered together in Server Groups.

Server Group
In the configuration tool, Server Groups are used to group together

server instances with different port definitions. A Server Group can corresponds to one physical CICS server or region, or to several physical servers.

Workload management
The ability of the Workload Manager to select a suitable CICS server as a target for a client request. The load balancing algorithms try to spread the workload over various server instances on a particular region (via random selection) and they allow the user to effectively specify a bias for the selection of a particular region. Known inactive regions are not considered for selection.

Listener
This term was used in CICS Universal Clients Version 3.0, and has now been replaced by **Server instance**.

Note: Apart from TCP/IP, other protocols supported by CICS Universal Clients can be used to communicate with CICS servers. For example, APPC can be used, and in this case the server instances can be identified by Logical Units (LU), rather than by TCP/IP ports.

What the Workload Manager needs to know

This section discusses how CICS server instances relate to the CICS regions, and the information required by the Workload Manager to perform its load management.

The following table illustrates knowledge that the Workload Manager must have. Specifically, the client must be able to correlate server instances against the regions to which they belong.

CICS Region	Server instance
REGION1	CICS1 CICS2 CICS3
REGION2	CICS4 CICS5
REGION3	CICS6

The clients must connect to a specific region via one server instance only, therefore a client in the example can connect to regions using one of the following groupings of servers:

```
(CICS1, CICS2, CICS3),  (CICS4, CICS5),  (CICS6)
```

For example, connections from a specific client to CICS2, CICS4, and CICS6 would be valid but connections to CICS1, CICS3, and CICS5 would not be valid because CICS1 and CICS3 belong to the same region.

As mentioned previously, if a CICS region needs to service a large number of clients, multiple server instances must be predefined, each with their own port numbers. From a client perspective, each server definition represents a uniquely targetable CICS region instance.

For example, if a port can handle 200 concurrent clients and there are 8000 clients, this means 40 ports are needed to support all clients - assuming the clients can distribute their connections across these ports, this would optimally result in 200 clients per port. Using a random distribution model, it is statistically possible for the distribution to be 210 clients on one port and only 190 on another (assuming 200 on the remainder). This would result in a set of client failures. The solution is to increase the number of available ports. For example, increasing the number of ports to 50 would result in a randomly even distribution of 160 clients / ports. The random selection algorithm would have to deviate now by a full 25% to exceed 200 clients on one port.

As a port can only handle a relatively small number of concurrent client connections, balancing of distribution of server instances across all the clients is vitally important. In the example, if all clients selected CICS2 as the connection to REGION1, it is likely that the port for CICS2 would be swamped very quickly and result in failures. If all clients need to access REGION2, then 50% should use CICS4, and 50% should use CICS5.

Not all CICS programs need be available on all possible CICS regions. The exit knows the name of the program that the client request is for, and can select from a subset of all regions, those regions on which a particular program is defined. The exit also has knowledge of a set of regions to which undefined programs may be routed.

For example:

Program	**Region**	**Average Elapsed Time**
PROG1	REGION1 REGION2 REGION3	10 20 15
PROG2	REGION1 REGION3	20 10
PROG3	REGION2	5

This data can be used both to monitor application execution, and also provide input to workload management algorithms.

In conclusion the Workload Manager must know which regions the CICS Universal Clients can connect to, the server instances that belong to each region, and the programs that can be run in each region. You supply this information using the configuration tool which produces the appropriate entries in the configuration file, see “Configuring the Workload Manager” on page 132.

Load balancing algorithms

You can select a ″round-robin″ algorithm or a biasing algorithm for load balancing. You select an algorithm using the configuration tool see “Configuring the Workload Manager” on page 132.

The round-robin algorithm

The round-robin algorithm assumes that all CICS regions are equally valid for selection. In the round-robin algorithm, when the client is initially started, it reads from the configuration file a list of all possible CICS regions to which any ECI request can be sent. An indication of the last region selected is also maintained. When a new ECI or EPI request arrives at the exit, the next region in the list is selected as the target for that ECI or EPI request. The next region indicator is then advanced until the last region has been selected, at which point the first region is selected again as the target for a request.

Note: The round-robin algorithm is also used by the EPI/CICSTERM/CICSPRNT **CICS_EpiAddTerminal** or **CICS_EpiAddExTerminal** requests (see the CICS Family: Client/Server Programming book).

The biasing algorithm

The biasing algorithm provides a means of balancing workload by specifying that workload should be biased in favor of particular regions. For example, if there are two regions and one has a bias of 75 and the other 25, then program requests will be sent in a ratio of 3:1 to the first region. If a region fails, the internal biasing calculation changes. If two regions are available, one with a bias of 100 and the other with a bias of 0, then all requests are sent to the first region. If the first region becomes unavailable, all requests are now directed to the second region. This is a special case of a bias value of 0, which means use only if no other region is available.

Another way of looking at biasing is:

```
probability a region will be selected =   bias for region
                                          ------------------
                                          total bias for all regions
```

The biasing algorithm **only** works for ECI calls and does not work for EPI calls. If you attempt to run an EPI application whilst the biasing algorithm is selected, the round-robin algorithm is used instead.

Failure recovery

If a request to route an ECI call to a particular region fails, indicating that the region is no longer available, the region is removed from the list of available regions.

The exit can be re-invoked for a particular request if that request fails to reach the target CICS region. The exit can then select an alternate, available CICS region for re-submission. Requests are sent adaptively to known, functional CICS regions. Regions that have previously failed can be ″pinged″ periodically to determine if the region's availability has been restored

A region that has been lost due to having previously failed as a target of a routing request is re-added to the list of available regions after a configurable elapsed time. If after re-adding, the region again fails as the target of a request, it is again removed from the active list and the algorithm repeated.

Note: An EPI/CICSTERM/CICSPRNT **CICS_EpiAddTerminal** or **CICS_EpiAddExTerminal** request that fails will also result in a region being marked out of service. See the CICS Family: Client/Server Programming book for more information about **CICS_EpiAddTerminal** and **CICS_EpiAddExTerminal** .

Exit implementation

The exit is implemented as a re-entrant multi-threaded Dynamic Load Library (DLL). This exit (code) is dynamically loaded by each client application that executes. The DLL executes in the process address space of the ECI-calling application rather than the CICS Client daemon.

ECI implementation

The Workload Manager is implemented as a Windows DLL called **cclmeci.dll.** This DLL contains implementations for the following functions:

CICS_EciInitializeExit
Initialize the exit. Called once per process at the first ECI call. This function builds the list of available regions and other attributes.

CICS_EciExternalCallExit1
Called at the start of an ECI call. This function is passed a reference to the ECI parameter block and may change the value of the eci_system_name field to select an alternate CICS server. The value of

this field may be a Server name. This function selects the correct target CICS server for each ECI client request. This function involves workload management algorithms.

CICS_EciSystemIdExit
If on return from the **CICS_EciExternalCallExit1,** the ECI request fails to reach the target CICS server, either because of communication failure or a CICS server failure, **CICS_EciSystemIdExit** is called to select an alternate CICS server target. The function flags that a CICS server is no longer available and should be removed from the list of selectable CICS servers. The workload management algorithm is then re-executed to select the next CICS server to be targeted by the client.

EPI implementation

The Workload Manager is implemented as a Windows DLL called **cclmepi.dll.** This DLL contains implementations for the following functions:

CICS_EpiInitializeExit
Initialize the exit. Called once per process at the first EPI call. This function builds the list of available regions and other attributes.

CICS_AddTerminalExit
Selects a CICS server.

CICS_EpiSystemIdExit
CICS_EpiSystemIdExit is called to select an alternate CICS server target. The function flags that a CICS server is no longer available and should be removed from the list of selectable CICS servers. The workload management algorithm is then re-executed to select the next CICS server to be targeted by the client.

Workload Manager installation

When you install CICS Universal Client for Windows NT, or CICS Universal Client for Windows 98, the ECI Exit DLL (cclmeci.dll) and the EPI Exit DLL (cclmepi.dll) are installed into the CICS Universal Clients\BIN directory:

Once installed, the exit is used by succeeding ECI client calls without modification to any currently existing ECI program.

The supplied load management DLLs can coexist with any user-written DLLs. The user-written DLLs are called after the load management DLLs. (There is one exception however; **CICS_EciSetProgramAliasExit**.)

Configuring the Workload Manager

You must configure the Workload Manager before you can use it. This is done by creating the appropriate entries in the configuration file, preferably by using the configuration tool. The configuration file is used by the Workload Manager to control its operation, and determine the workload balancing. In the Workload Manager configuration, you define which type of balancing algorithm to use.

The configuration parameters are as follows:

Enable the Workload Manager
Select this check box to enable the Workload Manager. The default is that the Workload Manager is disabled.

Round Robin
Select this radio button if a round-robin algorithm is to be used. For more information, refer to “The round-robin algorithm” on page 129.

Biasing
Select this radio button if a biasing algorithm is to be used. For more information, refer to “The biasing algorithm” on page 129.

Region Timeout
Enter a value in seconds. When a CICS region is selected as the target of a client request by the exit, and that region is unavailable, the exit flags the region as invalid and no longer includes that region in the list of target regions for requests. The region timeout value specifies the number of seconds after which an invalid region can once more be included in the list of target regions. For more information, refer to “Failure recovery” on page 130.

See “Configuring the Workload Manager” on page 64 for information on how to define Server Groups, and how to define biasing information for programs.

The following excerpt from the configuration file shows the parameters in the LOADMANAGER section:

```
SECTION LOADMANAGER
        Type = 1
        Timeout= 100
        BIAS=EC01;REGION1;25;CICSTCP5;75
        NOBALANCE=EC02

ENDSECTION
```

The Type parameter determines whether a Round Robin (0) or Biasing (1) algorithm is used.

The BIAS parameter lets you set the bias for a certain ECI application over several regions. In the above example, EC01 is a program name, while REGION1 and CICSTCP5 are regions. The EC01 program is biased over these regions with values of 25 and 75 respectively. The region name has to match a region specified in one of the Server Sections of the configuration file.

The NOBALANCE parameter specifies that no load balancing is performed for the EC02 program.

Tracing the Workload Manager

You can enable tracing for the Workload Manager in two ways:

1. By selecting Workload Manager in the **Trace** section of the configuration tool.
2. By specifying **LMG** on the CICSCLI /m command.

   This command overrides the configuration setting.

By default, tracing of the Workload Manager is not enabled.

Programming requirements

To invoke the Workload Manager correctly, the eci_version field in the ECI_PARMS control block passed to the CICS_ExternalCall function must be ECI_VERSION_1A or later.

The Workload Manager will not route extended Logical Unit of Work (LUW) calls to more than one region at a time. It is meaningless to send an extended request started on one region to a second region.

Chapter 9. Problem determination

This chapter discusses problem determination

- "Preliminary checks and information"
- "Program support" on page 136
- "CICS Universal Client problem determination" on page 142
- "CICS server problem determination" on page 150
- "Communication problem determination" on page 150
- "Troubleshooting" on page 152

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.

© Copyright IBM Corp. 1994, 1999

Program support

If you need to contact your support organization, you should first refer to the Service and Support card supplied with the product for details of the support available to you, or visit our Web site at:

```
http://www.ibm.com/software/ts/cics/
```

and follow the **Support** link.

This section provides guidance on reporting problems, detailing the information you should collect to assist your support organization. It also explains how the problem is handled through to resolution and the provision of a fix.

Reporting problems

Before reporting your problem to the support organization try to ensure that the error is actually occurring within your CICS Universal Clients system, even if you are unsure whether the problem is due to CICS Universal Clients itself.

In practice, many errors reported to support organizations turn out to be user errors, or they cannot be reproduced, indicating just how difficult it can be to determine the precise cause of a problem. User errors are mainly caused by faults in application programs or by errors in setting up systems.

The ability of the support organization to resolve your problem quickly is dependent on the quality of the information you provide to them in relation to the problem. For this reason it is good practice to collect and organize problem data before making the initial contact.

Summarize your problem and the documentation/information you collect on a problem reporting sheet, ideally using the suggested structure and headings shown in Table 11 on page 139. 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 138 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 142. 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 138 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 141. 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 142.

Documentation of problems

In a communication environment, where many clients may be connected to several servers, you must obtain information from both the client and the server sides.

To facilitate problem determination, try to reduce your environment to only one client workstation and one server to narrow down the range of possibilities that may cause the error.

Listed below are the sources of information likely to be needed by the support organization when tackling problem determination for CICS Universal Clients. The list is also summarized on the example problem reporting sheet (see Table 11 on page 139), 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 or Event logs
- A dump file for the CICS server
- A trace file for both CICS Universal Client (binary) and the server
- A trace for any communication product you were using
- Listings of relevant application programs
- A sample short program that demonstrates the problem
- CICS Universal Clients and Server message logs wherever possible. These contain information that is often overlooked.
- A list of APARs applied to the system
- Installation and customization parameters in effect at time of failure (for example, the configuration file or the .NDF file of Communications Server).
- Details of any user environment variables and hardware configurations.
- Operating Systems, compiler and other application release levels.
- A list of all the information you are providing, and the form in which it is being provided.

Table 11. Sample problem reporting sheet

PROBLEM REPORTING SHEET		
Date Company name Address Telephone	Severity	Problem Number Contact name
System configuration—CICS Client		
CICS Client Rel Client service level Fixes applied?	COBOL compiler level C compiler level PL/I compiler level	Operating System Communications prod.
System configuration—CICS Server		
CICS Server Rel Server service level Host CICS Rel Fixes applied?	COBOL compiler level C compiler level PL/I compiler level JDK/JRE level	Operating System Communications Prod. Database Server
Problem/Enquiry		
Abend/Trap Wait Loop Performance	Incorrout Message Other	
Documentation available—CICS Client		
ctg.ini Message Logs CICS Client Trace Symptom string Comms product trace	Program output Application source Translator output Compiler output	Operating system dump CICS Client dump Testcases User Env. variables
Documentation available—CICS Server		
Abend code Message Logs CICS Server Trace Symptom string Comms product trace	Program output Application source Translator output Compiler output	Operating system dump CICS Server dump Testcases User Env. variables
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.

- 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: UN*nnnnn* (where *nnnnn* is a valid PTF number provided with the service fix).
- Obtain Windows NT operating system details by using the WINVER.EXE command at a command prompt.

  If you are using SNA Server, you can find its version number by entering the SNAVER.EXE command.
- 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.

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 149

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 97), or by specifying a list of components using the configuration tool.

The output from the trace function is a binary trace file called, by default, cicscli.bin in the Client \BIN directory. Using the configuration tool, you can specify the filename for this file, but not the extension, which is always .BIN. Using the **Maximum Client wrap size** configuration setting, you can specify that the binary trace file should wrap into a second trace file, and you can also specify the maximum size of these files.

To read the trace, you must run the CICSFTRC utility to convert the binary file(s) into a text file, cicscli.trc by default. The following summarizes the trace files (default names are shown):

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 144 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
```

To stop client trace, enter the CICSCLI command with the /o option (or click on the equivalent icon), for example:

```
CICSCLI /o
```

The trace is also automatically stopped if you stop the client by using the `CICSCLI /x` command.

Wrapping of client trace

You can control the size of the binary trace file by specifying that it wraps into a second trace file. You turn on wrapping of trace using the **Maximum Client wrap size** configuration setting, where you specify the maximum size of the wrapping trace (in kilobytes). If this value is 0 (the default), wrapping trace is not turned on.

When wrapping trace is turned on, two files (called cicscli.bin and cicscli.wrp) are used. Each file can be up to half the size of the **Maximum Client wrap size** value.

Formatting the binary trace file

You use the Binary Trace Formatter utility CICSFTRC to convert the binary trace file cicscli.trc to ASCII text. The utility has the following parameters:

/m=*list of components*
: Specifies that only trace points from the listed components are written to the text file. The components you can specify are the same as for CICSCLI /m. If /m is not specified, all trace points in the binary trace are written to the text file.

/w[*=filename***]**
: Indicates that there are two binary trace files to format and then concatenate (that is, the binary files were created with a wrapping trace). If no filename is specified with the /w parameter, CICSFTRC assumes that the name of the second trace file is cicscli.wrp.

/n Indents entry and exit points in the test trace file to make it more readable. By default, indentation is turned off.

/p Specifies that the TIDs and PIDs are to be shown in the trace. By default, this parameter is turned off.

/i=*filename*
Specifies the name of the input (binary) trace file, which is cicscli.bin by default.

/o=*filename*
Specifies the name of the output (text) trace file. If no /o parameter is specified, the name of the text trace file is assumed to be cicscli.trc.

/d=*data size*
Specifies the maximum amount of data to write to disk for each dump. If this option is not specified, the maximum data size is 512 bytes.

If wrapping of trace is turned on, *data size* can be a maximum of half of the wrapping size.

CICS Universal Clients trace analysis

The CICS Universal Clients trace file records detailed information on all actions taken during the execution of a particular activity. You can use this information in your problem determination activities, and to help understand how the CICS Universal Clients perform a particular function, for example, establishing a connection to a CICS server.

If you cannot interpret the trace yourself, you should contact your support organization and forward the unformatted binary trace file.

Some sample traces are shown in “Sample CICS Universal Clients trace” on page 146.

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.

Sample CICS Universal Clients trace: Figure 7 on page 147 shows the trace information recorded during the successful connection of a CICS Universal Client to a CICS server using the TCP/IP protocol. The trace was generated using the commands:

```
CICSCLI -s=cicstcp -d
CICSCLI -o
```

The trace was generated for CICS Universal Client for AIX but is applicable for other CICS Universal Clients

```
11:01:47.129 [1007] TRC: CCL1042  *** CICS Client for AIX v3.1 Service Level 00 - service trace started
11:01:47.130 [2183] CCL: CCL2048  Maximum trace data size set to 512
11:01:47.133 [2114] CCL:CCL2142 GetNextTimeout timeout is 0 seconds
11:01:47.135 [2030] CCL:CCL2106 Comms Event : LINK-UP
11:01:47.135 [2019] CCL:CCL2055 Connection with server established (LinkId=1)
11:01:47.135 [2035] CCL:CCL2109 Send server TCS data
11:01:47.136 [3214] CCL: CCL3251  Comms Allocate request (LinkId=1, Tran='CCIN')
11:01:47.136 [3217] CCL: CCL3238  Comms Allocate completed (LinkId=1, ConvId=1, Rc=0)
11:01:47.136 [2127] CCL:CCL2143 CommsBegin - OK
11:01:47.138 [3100] CCL: CCL3113  CCIN install request: ApplId='*       ', Codepage=819
11:01:47.138 [3225] CCL:CCL3242 Comms Send request: Length=38 (ConvId=1)
11:01:47.139 [3229] CCL:CCL3241 Comms Send completed (ConvId=1, Rc=0)
11:01:47.139 [2131] CCL:CCL2143 CommsSendData - OK
11:01:47.139 [3230] CCL:CCL3254 Comms Wait request (ConvId=1)
11:01:47.140 [4410] DRV:CCL4411 TCP/IP (to CICSTCP) send data: Length=90
50ef0  0000005A00000001 000B90A0280502FF  ...Z........(...  ...]............
50f00  0003D1004004C3C3 C9D5180B02404040  ....@........@@@  ..J. .CCIN...
50f10  404040404040400B 01BF000000000000  @@@@@@@.........         .........
50f20  00000083002612FF 0000000C01010000  .....&..........  ...c............
50f30  0000000300000008 03F8F1F900000008  ................  .........819....
50f40  0401000000000006 0C80              ..........        ..........
11:01:47.140 [3234] CCL:CCL3243 Comms Wait completed (ConvId=1, Rc=0)
11:01:47.140 [2114] CCL:CCL2142 GetNextTimeout timeout is 0 seconds
11:01:47.163 [2029] CCL:CCL2107 Comms Private Event : COMMS
11:01:47.163 [4411] DRV:CCL4412 TCP/IP (to CICSTCP) receive data: Length=12
40220  0000003D00000001 00039001          ...=........      ............
11:01:47.164 [4409] DRV:CCL4412 TCP/IP (to CICSTCP) receive data: Length=49
50efc  003112FF0000000C 0102000000000003  .1..............  ................
50f0c  0000000D017CF0E9 F8C1C1C1C1000000  .....|..........  .....@0Z8AAAA...
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  .....|..........  .....@0Z8AAAA...
50f1c  0B03F8F8F5F960F1 0000000909010100  ......′.........  ..8859-1........
50f2c  00                                 .                 .
11:01:47.167 [3102] CCL: CCL3114  CCIN install response: ApplId='@0Z8AAAA', 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 7. Sample CICS Universal Clients trace

Message ID	**Explanation**

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 143).

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 8 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 8. CICS Universal Clients trace: using an invalid port number

Message ID **Explanation**

CCL4413 Shows the port number used for this connection request.

You must check your definitions in the SIT on the server, the configuration file on the workstation, and the services file for the port number specified.

You must provide a valid port number or use the default value.

CICS Universal Client error processing

There are two categories of messages in IBM CICS Universal Clients:

1. Messages displayed to the user
2. Errors written to the CICS Universal Clients error log and trace file.

The *CICS Universal Clients Messages* book explains all of these messages.

Error log messages

Any errors on the client workstation that are not caused by incorrect use of the API are written to the client error log.

The error log (cicscli.log) is an ASCII text file that you can browse using a standard text editor.

The error messages are explained in *CICS Universal Clients Messages* and also the help text for all messages is provided in two ASCII text files in the Client \BIN subdirectory. You can view these using any standard text editor:

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.

Pop-up messages

Errors generated from within the client process are displayed in a pop-up window. You must click on **OK** in the pop-up window before the client can continue processing. There may be times when you do not want messages (client error and security) to appear in pop-up windows; for example, if you leave a CICS Universal Client running unattended overnight. To disable the display of pop-up messages, enter the following command:

```
CICSCLI  /n
```

When the display of pop-up messages is disabled, the messages are still written to the client error log. To re-enable the display of pop-up messages, enter the following command.

```
CICSCLI  /e
```

CICS Universal Client error processing

You can specify the /n parameter together with the /s parameter. The display of pop-up messages is enabled by default.

CICS server problem determination

The most important facilities for problem determination on CICS servers are:

- Traces
  - auxiliary
  - internal
- Dumps
- CICS message logs
- Statistics information
- Monitoring information
- Execution diagnostic facility (EDF)
- CICS-supplied transactions, CEBR and CECI
- Independent software vendor (ISV) tools

Information about these facilities is given in the *Problem Determination* books for the individual products (see “CICS publications” on page xvi). You may need to contact the server system administrator for more information, for example, about a CICS server error log.

The following shows the error message prefixes for CICS products:

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.

TCP/IP-providing products

TCP/IP provides the following diagnostic tools:

ARP
Displays and modifies the IP-to-Ethernet or token ring physical address translation tables used by address resolution protocol (ARP).

HOSTNAME
Displays the host name of your workstation.

IPCONFIG
Displays all current TCP/IP network configuration values. This is helpful if you have to find out if an IP interface is active.

NETSTAT
Displays protocol statistics and current TCP/IP network connections. This helps you obtain information about your own IP interfaces, for example, to list IP addresses and TCP/IP routing tables in use at your workstation.

NSLOOKUP
Displays information from Domain Name System (DNS) name servers.

PING
Verifies connections to a remote computer or computers.

TRACERT
Traces the TCP/IP path to a requested destination. It is useful for determining whether there is a problem with one of the intermediate nodes.

For more information on these utilities, refer to the online help for TCP/IP.

APPC-providing products

This section describes problem determination in products involved in APPC communication.

VTAM Buffer Trace

You can use VTAM buffer tracing to record the flow of data between logical units in the CICS environment. The trace entries include the netname of the terminal (logical unit) to which they relate. For details on VTAM buffer tracing and other VTAM problem determination functions, see the appropriate book in the VTAM library.

NetWare for SAA products

For information see the *Novell Netware APPC Technical Reference*.

NetWare error log events are written to the SYSTEM.LOG file in the \SYS directory.

SNA Server

See the *Microsoft SNA Server Administration Guide* for information on diagnostic tools that you can use with SNA Server.

The APING utility

In an APPC environment, you can use the APING utility to test a connection. APING exchanges data packets with a partner computer over APPC and times how long the data transfer takes. It can be used to get a first estimate of the session setup time between two computers and the throughput and turnaround time on that APPC session. You can use APING to determine whether a session can be set up between two computers and to display extensive error information if session allocation fails. APING consists of two transaction programs: APING, which runs on the client side, and APINGD, which runs on the server side.

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"
- "NetBIOS communication problems" on page 153
- "TCP/IP communication problems" on page 154
- "APPC communication problems" on page 155.

For information on error messages, refer to *CICS Universal Clients Messages*.

Starting clients and terminals

The following are solutions to problems that can occur when starting clients and terminals:

CICS for OS/2 server reports “transaction CCIN failed”

CCIN and CTIN are two transactions required on the CICS for OS/2 server to enable a CICS Universal Client (or another CICS for OS/2 server) to attach using TCP/IP. For TCP/IP support, either CICS for OS/2 Version 2.0.1, or CICS for OS/2 Version 2.0 with CSD4 are required. Although the two Versions are essentially identical, CCIN and CTIN must be defined in the program control table for Version 2.0. Both transactions call the same program—FAATPPCI.

A CICSTERM request has gone to the wrong server

If you do not specify the /s=*servername* option on the CICSTERM command, a CICSTERM request is issued against the first server listed in the configuration file, even if it is not yet activated. The *servername* should be as specified in the configuration file.

NetBIOS communication problems

The following are solutions to problems that can occur when communicating over NetBIOS:

There are only a few NetBIOS resources when running CICS for OS/2 server and CICS Universal Client for OS/2 on the same workstation

The CICS for OS/2 server is probably using all of the NetBIOS resources for itself and leaving none for CICS Universal Client for OS/2. To avoid restricting the number of commands and sessions, reduce the values of *n1*, *n2*, and *n3* in the \IBMLAN\IBMLAN.INI file, or PROTOCOL.INI file.

```
[networks]
    net1 = netbeui$, 0, LM10, n1,n2,n3
```

However, if you are using the LAN services program, leave sufficient resources for the LAN server/requester. Also in IBMLAN.INI, the `wrkheuristics` entry should contain 0 at position #18.

The NetBIOS connection fails. The trace reports: CCL4217 NetBIOS command failed for conversation 0/0: Cmd=0090, Rc=0014

This is a general return code returned by NetBIOS. Check that:

- the **Maximum NetBIOS Systems** value is greater than 0 in the System Initialization Table for CICS for OS/2
- If using CICS Universal Client for Windows NT you have configured your NetBIOS for the correct LAN type, for example, Novell or IBM. (The default is Novell.)

Message FAA5573W received from CICS for OS/2: NetBIOS Listener Name exists elsewhere on adapter 0

If the application ID specified in the CICS for OS/2 System Initialization Table is not unique in the network, message FAA5573W is displayed during startup of CICS for OS/2. This usually happens if you use the default value (CICSOS2) for the application ID.

To solve this problem, select **Continue** in the CICS for OS/2 console window and change the application ID according to your standards. You must shut down and restart CICS for OS/2 for the changes to take effect.

TCP/IP communication problems

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

CICS Client for Windows Family cannot load the TCP/IP protocol driver

You have probably received message CCL3229E in a pop-up window and message CCL3247 in the cicscli.log error log file. These messages indicate you have probably specified the wrong device driver in the configuration file. For all TCP/IP communication, CICS Universal Client for Windows 98 and CICS Universal Client for Windows NT should use the CCLWNTIP driver. Any vendor offering a WINSOCK interface can use this TCP/IP driver.

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 152 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 11. 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 11.

Automatic transaction initiation against a CICS Universal Clients terminal does not work

Different products provide different APPC implementations for handling inbound attaches. Therefore for CICS Universal Clients using IBM eNetwork Communications Server, Microsoft SNA Server, and IBM eNetwork Personal Communications, you must predefine the CRSR attach program to relay the information to the CICS Universal Clients. This is necessary to enable the CICS server to perform automatic transaction initiation (ATI) against the CICS Universal Clients terminals. (As long as the CICS server supports CICS 3270 terminal emulation, see Table 2 on page 11.)

For CICS Universal Clients therefore, you must define the inbound attach transaction CRSR as follows:

CICS Universal Client for Windows 98 using IBM eNetwork Communications Server

Define CRSR to use program CCLCLNT.EXE. See “IBM eNetwork Communications Server” on page 41.

CICS Universal Client for Windows NT using IBM eNetwork Communications Server

Define CRSR to use program CCLCLNT.EXE. See “IBM eNetwork Communications Server” on page 41.

CICS Universal Client for Windows 98 using IBM eNetwork Personal Communications
Define CRSR to use program CCLCLNT.EXE. See “IBM eNetwork Personal Communications” on page 43.

CICS Universal Client for Windows NT using IBM eNetwork Personal Communications
Define CRSR to use program CCLCLNT.EXE. See “IBM eNetwork Personal Communications” on page 43.

CICS Client for Windows Family using NetWare for SAA
You do not need to define CRSR, because an exit is automatically driven when an inbound attach is received.

DCE RPC communication problems

This section discusses problems that can occur when communicating over DCE RPC.

To help with solving problems, it is useful to know what userid the CICS server gives the client. You can use the CEMT I TA transaction to list all the running tasks on the server together with the facility (terminal) they are running against, and the userid of the user running it. Enter `CEMT I TA` from a Client terminal emulator, then look for a CEMT transaction with a facility value that matches the terminal id in the status bar of the terminal emulator.

The DCE Client can connect to the server but only uses unauthenticated RPC communication.

Possible causes of this are:

- You have not run the CCLAUTH utility.

  If you see the following tracepoints:

  ```
  [5601] CCL5612 RPC authentication for all servers is disabled
  [5625] CCL5608 RPC DCE reports: Requested key is unavailable ( dce/sec )
  [5602] CCL5612 DCE authentication for the client is disabled
  ```

  it is likely that CCLAUTH has not been run on the client machine.

  If you do run CCLAUTH, and the client is correctly configured, you should see:

  ```
  [5601] CCL5612 RPC authentication for all servers is enabled
  [5602] CCL5612 DCE authentication for the client is enabled
  ```

  For more information on CCLAUTH, see “Configuring the CICS Universal Client for authentication” on page 119.

- The user application program/client terminal is not logged in to DCE.

  If the following message does not appear in the client trace, it means that the application program or client terminal is not logged in to DCE.

  ```
  CCL2083 DCE Context supplied
  ```

  This message precedes the client sending information to the server. The following are some examples tracepoints that it directly precedes:

  ```
  CCL2045 Request directed to server name
  CCL3251 Comms allocate request
  CCL3242 Comms send request
  ```
- The runtime protection level in the server's region definitions is set to none.

  If the trace contains `CCL2083 DCE context supplied` messages but does not contain the message:

  ```
  CCL5614 RPC ( to servername ) caller principal: principal name
  ```

  then the CICS region's runtime protection level is probably set to `none`.
- The userid that the client user logged into DCE with does not match the DCE login userid specified in the server UD.stanza entry.

  If the CEMT I TA transaction shows that the client is still not given the authenticated userid, then check that the principal given in message CCL5614 is the same as the principal specified when the desired userid was created. If these are different, this means that the user logged in with the wrong principal.

Chapter 10. Migration issues

This chapter discusses issues concerning the migration of applications between different versions of CICS Clients.

Application compatibility

CICS Universal Clients do not support the *development* of 16-bit applications, but old 16-bit applications can still be run.

ECI and EPI compatibility

With CICS Client for Windows 95 and CICS Client for Windows NT Version 2.0.3 and later it is possible to port 16-bit applications to 32-bit applications. If such applications are to be changed, or if they need to be ported to other client platforms, the ECI call requires very minor changes before recompiling.

Notes:

1. The header files are now called cics_eci.h (C), cicseci.cbl (COBOL), and cics_epi.h (C), cicsepi.cbl (COBOL), References to faaecih.h and faaepih.h, or faaeciw.cbl and faaepiw.cbl must be changed.
2. The CICS ECI entry point is now called CICS_ExternalCall() for C and CICSEXTERNALCALL for COBOL.
3. The CICS EPI entry point is now called CICSEXTERNALCALL for 32-bit COBOL. Any calls to FaaExternalCall() must be changed.
4. The correct link library is now called CCLWIN.LIB Attempts to link with FAACLIB.LIB or FAACIC32.LIB will fail.

ECI and EPI client applications are portable between members of the CICS Universal Clients family, provided the application does not contain operating-system-specific calls.

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

© Copyright IBM Corp. 1994, 1999

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.

CICS Universal Client for Windows NT data conversion

The ASCII CCSID is determined dynamically. By default the GetOEMCP function is used to obtain the current OEM code page identifier, for example CCSID 850, for the system. If the **Use OEM codepage** configuration setting is specified, the GetACP function is used to obtain the current ANSI code page identifier, for example CCSID 1252, for the system.

The identifier may not be unique. For example, there are differences between IBM CCSID 932 and Microsoft CCSID 932 - the CICS Client maps the Microsoft one to CCSID 943 which is the equivalent one defined in the CRDA Registry. The Microsoft CCSID 1252 may be the pre-euro or post-euro version - the post-euro is mapped to CCSID 5348.

© Copyright IBM Corp. 1994, 1999

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:

	CCSID	CPGID	
ASCII	00921	00921	PC data: Latvia, Lithuania
	01257	01257	MS Windows: Baltic Rim
EBCDIC	01112	01112	Host: Latvia, Lithuania

Cyrillic:

	CCSID	CPGID	
ASCII	00866	00866	PC data: Cyrillic
	00915	00915	ISO 8859-5: Cyrillic
	01251	01251	MS Windows: Cyrillic
EBCDIC	01025	01025	Host: Cyrillic multilingual

Estonia:

	CCSID	CPGID	
ASCII	00922	00922	PC data: Estonia
	01257	01257	MS Windows: Baltic Rim
EBCDIC	01122	01122	Host: Estonia

Greek:

	CCSID	CPGID	
ASCII	00869	00869	PC data: Greece
	00813	00813	ISO 8859-7: Grek
	01253	01253	MS Windows: Greek
EBCDIC	00875	00875	Host: Grek

Hebrew:

	CCSID	CPGID	
ASCII	00856	00856	PC data: Hebrew
	00916	00916	ISO 8859-8: Hebrew
	01255	01255	MS Windows: Hebrew
EBCDIC	00424	00424	Host: Hebrew

Latin-1:

	CCSID	CPGID	
ASCII	00437	00437	PC data: USA, many other countries
	00819	00819	ISO 8859-1: Latin-1 countries
	00850	00850	PC data: Latin-1 countries
	01252	01252	MS Windows: Latin-1 countries
EBCDIC	00037	00037	Host: USA, Canada, etc
	00273	00273	Host: Austria, Germany
	00277	00277	Host: Denmark, Norway
	00278	00278	Host: Finland, Sweden
	00280	00280	Host: Italy
	00284	00284	Host: Spain, Latin America (Spanish)
	00285	00285	Host: United Kingdom
	00297	00297	Host: France
	00500	00500	Host: International Latin-1
	00871	00871	Host: Iceland
	01047	01047	Host: Latin-1 Open Systems

Latin-1 including euro support:

	CCSID	CPGID	
ASCII	00858	00858	PC data: Latin-1 countries
	05348	01252	MS Windows: Latin-1 countries, version 2
EBCDIC	01140	01140	Host: USA, Canada, etc
	01141	01141	Host: Austria, Germany
	01142	01142	Host: Denmark, Norway
	01143	01143	Host: Finland, Sweden
	01144	01144	Host: Italy
	01145	01145	Host: Spain, Latin America (Spanish)
	01146	01146	Host: United Kingdom
	01147	01147	Host: France
	01148	01148	Host: International Latin-1
	01149	01149	Host: Iceland

Latin-2:

	CCSID	CPGID	
ASCII	00852	00852	PC data: Latin-2 multilingual
	00912	00912	ISO 8859-2: Latin-2 multilingual
	01250	01250	MS windows: Latin-2
EBCDIC	00870	00870	Host: Latin-2 multilingual

Latin-5:

	CCSID	CPGID	
ASCII	00857	00857	PC data: Latin-5 (Turkey)
	00920	00920	ISO 8859-9: Latin-5 (Turkey)
	01254	01254	MS Windows: Turkey
EBCDIC	01026	01026	Host: Latin-5 (Turkey)

Latin-9:

	CCSID	CPGID	
ASCII	00923	00923	ISO 8859-15: Latin-9
EBCDIC	00924	00924	Host: Latin-9

Japanese:

	CCSID	CPGID	
ASCII	00932	00897	PC data: SBCS
		00301	PC data: DBCS
	00942	01041	PC data: extended SBCS
		00301	PC data: DBCS
	00943	00897	PC data: SBCS
		00941	PC data: DBCS for Open environment
	00954	00895	G0: JIS X201 Roman
		00952	G1: JIS X208-1990
		00896	G2: JIS X201 Katakana
		00953	G3: JIS X212
EBCDIC	00930	00290	Host: Katakana, extended SBCS
		00300	Host: DBCS
	00931	00037	Host: Latin-1, SBCS
		00300	Host:DBCS
	00939	01027	Host: Latin-1, extended SBCS
		00300	Host: DBCS

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 168
- "CICS Family publications" on page 168
- "Book filenames" on page 169
- "Sample configuration documents" on page 169
- "Other publications" on page 170
- "Viewing the online documentation" on page 170

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.

© Copyright IBM Corp. 1994, 1999

- *CICS Transaction Gateway Programming*, SC34-5594

  This book provides an introduction to Java programming with the CICS Transaction Gateway.

  There are also additional HTML pages that contain programming reference information.

CICS Universal Clients books

- *CICS Universal Client for OS/2 Administration, SC34-5450*

  This book describes the administration of the CICS Universal Client for OS/2.
- *CICS Universal Client for Windows Administration*, SC34-5449

  This book describes the administration of the CICS Universal Client for Windows 98 and CICS Universal Client for Windows NT.
- *CICS Universal Client for AIX Administration, SC34-5348*

  This book describes the administration of the CICS Universal Client for AIX.
- *CICS Universal Client for Solaris Administration, SC34-5451*

  This book describes the administration of the CICS Universal Client for Solaris.
- *CICS Universal Clients Messages*

  This online book lists and explains the error and trace messages that can be generated by CICS Universal Clients.

  You cannot order this book.
- *CICS Universal Clients C++ Programming*, SC33-1923

  This book describes how to write object oriented programs for the ECI and EPI in the C++ language.
- *CICS Universal Clients COM Automation Programming*, SC33-1924

  This book describes how to write object oriented programs for the ECI and EPI according to the Component Object Model (COM) standard.

CICS Family publications

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

  This book describes the programming interfaces associated with CICS client/server Programming— the External Call Interface (ECI), the External Presentation Interface (EPI), and the External Security Interface (ESI). It is intended for application designers and programmers who wish to develop client applications to communicate with CICS server systems.

Book filenames

Table 12 show the softcopy filenames of the CICS Transaction Gateway and CICS Universal Client books.

Table 12. CICS Transaction Gateway and CICS Universal Clients books and file names

Book title	File name
CICS Universal Clients Messages	CCLHAB
CICS Universal Client for AIX Administration	CCLHAD
CICS Universal Client for OS/2 Administration	CCLHAE
CICS Universal Client for Windows Administration	CCLHAF
CICS Universal Client for Solaris Administration	CCLHAG
CICS Transaction Gateway for OS/390 Administration	CCLHAI
CICS Transaction Gateway Messages	CCLHAJ
CICS Transaction Gateway Programming	CCLHAK
CICS Transaction Gateway for Windows Administration	CCLHAL
CICS Transaction Gateway for OS/2 Administration	CCLHAM
CICS Transaction Gateway for AIX Administration	CCLHAN
CICS Transaction Gateway for Solaris Administration	CCLHAO
CICS Universal Clients C++ Programming	CCLHAP
CICS Universal Clients COM Automation Programming	CCLHAQ
CICS Family: Client/Server Programming	DFHZAD
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/`

and follow the **Library** link.

Other publications

The following International Technical Support Organization (ITSO) Redbook publication contains many examples of client/server configurations:

- *Revealed! CICS Transaction Gateway with more CICS Clients Unmasked, SG24-5277*

  This book supersedes the following book:
- *CICS Clients Unmasked, GG24-2534*

You can obtain ITSO Redbooks from a number of sources. For the latest information, see:

`http://www.ibm.com/redbooks/`

You can find information on CICS products at:

`http://www.ibm.com/software/ts/cics/`

Viewing the online documentation

You can access all of the documentation provided with CICS Transaction Gateway and CICS Universal Client in our online library. You need Adobe Acrobat Reader and a suitable Web browser to use the online library (and you may need to configure these).

To get to the online library:

- On Windows and OS/2, select the **Documentation** icon.
- On AIX and Solaris, run the **ctgdoc** script.

and the library home page is displayed.

The online library allows you to link to:

- CICS Transaction Gateway and CICS Universal Clients books in PDF format.
- Programming reference documentation in HyperText Markup Language (HTML) files (provided for CICS Transaction Gateway only).
- README files.
- Sample configuration documents in PDF format.
- Translated books in PDF format. (You may find that not all books are translated for your language.)
- The CICS Web site.

Guidance information on using Acrobat Reader is also provided.

Updated versions of the books may be provided from time to time, check our Web site at:

```
http://www.ibm.com/software/ts/cics/
```

and follow the **Library** link.

Viewing PDF books

The PDF information provides powerful functions for:

- Navigating through the information. There are hypertext links within PDF documents, and to other PDF documents and Web pages.
- Searching for specific information.
- Printing all or part of PDF documents on a PostScript printer.

You can find out more about Acrobat Reader at the Adobe Web site:

```
http://www.adobe.com/acrobat/
```

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.

© Copyright IBM Corp. 1994, 1999

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

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM United Kingdom Laboratories, MP151, Hursley Park, Winchester, Hampshire, England, SO21 2JN. Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

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

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

AIX
Anynet
CICS
CICS/MVS
DB/2
eNetwork
IBM
Interspace
LAN Distance
MVS
MVS/ESA
OS/2
OS/390
OS/400
RETAIN
System/390
TXSeries
VisualAge

VSE/ESA

VTAM

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

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

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Other company, product, and service names may be trademarks or service marks of others.

Glossary

This glossary defines special terms used in this book.

If you cannot find the word you are looking for, try the *IBM Dictionary of Computing*.

A

APING. A command used to verify an APPC link.

APPC. Advanced program-to-program communication. An implementation of the SNA/SDLC LU 6.2 protocol that allows interconnected systems to communicate and share the processing of programs. The APPC protocol is used by IBM CICS Universal Clients to communicate with CICS server systems.

application definition file. In software preparation, a file that contains information about how to install a particular CID-enabled software product, and how to gather configuration parameters to be used in the response file and software object.

application server. A CICS run-time process that executes CICS tasks.

ATI. See *automatic transaction initiation*.

Attach. In SNA, the request unit that flows on a session to initiate a conversation.

Attach Manager. The component of APPC that matches Attaches received from remote computers to accepts issued by local programs.

autoinstall. A facility that enables terminal definitions to be created automatically in a CICS server on request. The definition is deleted when the client terminal is logged off.

automatic transaction initiation (ATI). The process whereby a transaction request made internally within a CICS system or network leads to the scheduling of the transaction. An ATI request from a CICS Universal Client can cause the initiation of a transaction in a CICS server system.

B

BIND. In SNA, a request to activate a session between two logical units (LUs).

business logic. The part of a distributed application that is concerned with the application logic rather than the user interface of the application. See *presentation logic*.

C

CICS Gateway for Java. In CICS Universal Clients Version 2, an interface between a Java program and CICS that allows a Java-enabled browser to dynamically download a Java Applet and transparently access CICS data. This has been replaced by the CICS Transaction Gateway.

CICS on open systems. A term used to refer generically to the products: CICS for HP-UX, CICS for Sun Solaris, TXSeries for AIX, TXSeries for HP-UX, and TXSeries for Solaris.

CICS on System/390. A term used to refer generically to the products: CICS for MVS/ESA, CICS/MVS, and CICS for VSE/ESA.

cicsprnt. A command that allows you to define a printer terminal on the CICS Universal Client workstation. The command enables CICS applications running on the server to direct output to the client-attached printer.

cicsteld. A command that starts a CICS telnet daemon. The telnet daemon allows a telnet 3270 client to connect to a CICS server.

© Copyright IBM Corp. 1994, 1999

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

CID. Configuration, installation, and distribution.

CID methodology. An IBM-specified way to install and configure products on, or remove products from, remote workstations and hosts. CID-enabled products use response files and redirected installation and configuration to eliminate or reduce user interaction with the CID-enabled product.

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

client/server. A distributed application design model, in which the client sends a request to its partner, the server, which executes the request and returns the results to the client. This design model is associated typically with cooperative processing.

client configuration file. A file containing configuration information used to inform the CICS Universal Client of the CICS servers it can connect to, and the communication protocols to be used.

CNOS. See *Change-Number-of-Sessions*.

color mapping file. A file used by IBM CICS Universal Clients to customize the 3270 screen color attributes on client workstations.

COMMAREA. A communication area that is used for passing data both between programs within a transaction and between transactions.

connection. In data communication, an association established between functional units for conveying information. In Open Systems Interconnection architecture, an association established by a given layer between two or more entities of the next higher layer for the purpose of data transfer. In TCP/IP, the path between two protocol application that provides reliable data stream delivery service. In Internet, a connection extends from a TCP application on one system to a TCP application on another system.

control table. In CICS, a storage area used to describe or define the configuration or operation of the system.

conversation. The communication between a CICS Universal Client and a CICS server. In APPC, a connection between two transaction programs over an LU-LU session that allows them to communicate with each other while processing a transaction. See also *session*.

conversation security. In APPC, a process that allows validation of a user ID or group ID and password before establishing a connection.

cooperative processing. A subset of *distributed processing,* in which at least the user interface (presentation) aspect of the application runs on a programmable workstation.

D

data link control (DLC). A set of rules used by nodes on a data link (such as an SDLC link or a token ring) to accomplish an orderly exchange of information.

DBCS. See *double-byte character set*.

DCE. See *distributed computing environment*.

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 computing environment (DCE). The Distributed Computing Environment (DCE) provides secure, authenticated, client/server interoperability across diverse platforms.

Distributed Feature client. The CICS clients supplied with CICS for OS/2 Version 2.0. These are functionally inferior to IBM CICS Universal Clients.

distributed processing. The processing of different parts of the same application in different systems, on one or more processors.

domain. In the Internet, a part of a naming hierarchy in which the domain name consists of a sequence of names (labels) separated by periods (dots).

domain name. In TCP/IP, a name of a host system in a network.

Domain Name Server. In TCP/IP, a server program that supplies name-to-address translation by mapping domain names to internet addresses. Synonymous with name server.

dotted decimal notation. The syntactical representation for a 32-bit integer that consists of four 8-bit numbers written in base 10 with periods (dots) separating them. It is used to represent IP addresses.

double-byte character set (DBCS). A set of characters in which each character is represented by 2 bytes. Languages such as Japanese, Chinese and Korean, which contain more symbols than can be represented by 256 code points, require double-byte character sets. Because each character requires 2 bytes, the typing, display, and printing of DBCS characters requires hardware and programs that support DBCS. Contrast with *single-byte character set*.

E

ECI. See *external call interface*.

emulator, emulation program. A program that allows a host system to communicate with a workstation in the same way as it would with the emulated terminal. In IBM CICS Universal Clients the terminal emulation function allows client workstations to run CICS transactions that use 3270 data flows.

EPI. See *external presentation interface*.

Ethernet. A 10-megabit baseband local area network that allows multiple stations to access the transmission medium at will without prior coordination, avoids contention by using carrier sense and deference, and resolves contention by using collision detection and transmission. Ethernet uses carrier sense multiple access with collision detection (CSMA/CD).

external call interface (ECI). A facility that allows a non-CICS program to run a CICS program. Data is exchanged in a COMMAREA as for normal CICS interprogram communication.

external presentation interface (EPI). A facility that allows a non-CICS program to appear to CICS as one or more standard 3270 terminals. 3270 data can be presented to the user by emulating a 3270 terminal or by using a graphical user interface.

external security interface (ESI). A facility that enables client applications to verify and change passwords for user IDs on CICS servers.

G

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

H

host. In the Internet suite of protocols, an end system. The end system can be any system; it does not have to be a mainframe.

host address. An IP address that is used to identify a host in an internet.

host ID. In TCP/IP, that part of the Internet address that defines the host on the network. The length of the host ID depends on the type of network or network class (A, B, or C).

host name. In the Internet suite of protocols, the domain name of a host.

I

independent LU. A logical unit (LU) that can both send and receive a BIND, and which supports single, parallel, and multiple sessions. See *BIND*.

internet. A collection of networks interconnected by a set of routers that allow them to function as a single, large network. Contrast with *Internet*.

Internet. The internet administered by the Internet Architecture Board (IAB), consisting of large national backbone networks and many regional and campus networks all over the world. The Internet uses the Internet suite of protocols.

Internet address. See *IP address*.

Internet Architecture Board. The technical body that oversees the development of the internet suite of protocols known as TCP/IP.

Internet Packet Exchange (IPX). In Novell NetWare LANs, a communication protocol that sends data packets to requested destinations.

Internet Protocol (IP). In TCP/IP, a protocol that routes data from its source to its destination in an Internet environment.

Interoperability. The capability to communicate, execute programs, or transfer data among various functional units in a way that requires the user to have little or no knowledge of the unique characteristics of those units.

IP. Internet Protocol.

IP address. The 32-bit address defined by the internet protocol, standard 5, Request for Comment (RFC) 791. It is usually represented in dotted decimal notation.

IPX. Internet Packet Exchange.

K

keyboard mapping file. A file used by IBM CICS Universal Clients to customize the keyboard mapping of client workstations.

L

LAN. See *local area network.*

link support layer (LSL). The part of Novell's Open Data-Link Interface (ODI) that allows the LAN driver for a network adapter to service more than one protocol. See *Open Data-Link Interface (ODI).*

local area network (LAN). A network of workstations, or terminals, or both, where all the connected systems are relatively near to each other. See *wide area network.*

logical unit (LU). In SNA, a port through which an end user accesses the SNA network in order to communicate with another end user and through which the end user accesses the functions provided by system services control points (SSCP). An LU can support at least two sessions, one with an SSCP and one with another LU, and may be capable of supporting many sessions with other logical units. See *network addressable unit, primary logical unit, secondary logical unit*.

LU-LU session. In SNA, a session between two logical units (LUs) in an SNA network. It provides communication between two end users, or between an end user and an LU services component.

LU-LU session type 6.2. In SNA, a type of session for communication between peer systems. Synonymous with APPC protocol.

LU 6.2. See *APPC.*

M

MAC. Medium access control.

medium access control (MAC) sublayer. One of two sublayers of the ISO Open Systems Interconnection data link layer proposed for local area networks by the IEEE Project 802 Committee on Local Area Networks and the European Computer Manufacturers Association (ECMA). It provides functions that depend on the topology of the network and uses services of the physical layer to provide services to the logical link control (LLC) sublayer. The OSI data link layer corresponds to the SNA data link control layer.

mode. In SNA, a set of parameters that defines the characteristics of a session between two LUs.

N

name server. In TCP/IP, synonym for Domain Name Server In Internet communications, the host that translates host names into their respective internet addresses when requested by the hosts on the network A physical device, and its associated software, that enables a processor or controller to be connected to a network.

NDIS. Network Driver Interface Specification.

NetBIOS. Network Basic Input/Output System. The standard interface to networks, IBM personal computers (PCs), and compatible PCs, that is used on LANs to provide message, print server, and file-server functions. The IBM NetBIOS application program interface (API) provides a programming interface to the LAN so that an application program can have LAN communication without knowledge and responsibility of the data link control (DLC) interface.

network address. In SNA, an address, consisting of subarea and element fields, that identifies a link, link station, or network addressable unit (NAU). Subarea nodes use network addresses; peripheral nodes use local addresses. The boundary function in the subarea node to which a peripheral node is attached transforms local addresses to network addresses and vice versa. See also *network name.*

network addressable unit (NAU). In SNA, a logical unit, a physical unit, or a system services control point. The NAU is the origin or the destination of information transmitted by the path control network. See also *logical unit, network address, network name.*

network name. In SNA, the symbolic identifier by which end users refer to a network addressable unit (NAU), link station, or link. See also *network address.*

Network Driver Interface Specification (NDIS). A specification developed jointly by Microsoft and 3Com that lets an OS/2 system support one or more network adapters and protocol drivers. The device driver functions at the medium access control (MAC) layer. This kind of driver is also called an NDIS MAC driver.

node type. In SNA, a designation of a node according to the protocols it supports and the network addressable units (NAUs) it can contain. Four types are defined: 1, 2, 4, and 5. Type 1 and type 2 nodes are peripheral nodes; type 4 and type 5 nodes are subarea nodes.

O

ODI. Open Data-Link Interface.

Open Data-Link Interface (ODI). Novell's architecture that allows multiple LAN drivers and protocols to coexist in network systems. The ODI specification defines the set of interface and software modules used to decouple device drivers from protocol stacks and to enable multiple protocol stacks to share transparently the network hardware.

P

pacing. A technique by which a receiving station controls the rate of transmission of a sending station to prevent overrun.

packet internet groper (PING). In Internet communications, a program used in TCP/IP networks to test the ability to reach destinations 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

remote procedure call (RPC). Remote procedure calls are analogous to the standard procedure calls used in modern programming, providing modularity and reducing code size. The main difference between standard procedure calls and remote procedure calls is that in a remote procedure call, the called procedure is being executed by a different process than the caller. To the calling procedure, a remote procedure call looks the same as a call to a local procedure. When a program makes a remote procedure call to a remote application, the procedure’s parameters are automatically bundled into a request message, which is then sent to the remote program or system.

request unit (RU). In SNA, a message unit that contains control information such as a request code, or function management (FM) headers, end-user data, or both.

request/response unit. A generic term for a request unit or a response unit. See also *request unit* and *response unit*.

response file. A file that contains a set of predefined answers to questions asked by a program and that is used in place of a user dialog. See *CID methodology*.

response unit (RU). A message unit that acknowledges a request unit; it may contain prefix information received in a request unit.

RU. Request unit. Response unit.

RPC. See *remote procedure call*.

S

SAA. See *Systems Application Architecture*.

SBCS. See *single-byte character set*.

SDLC. See *Synchronous Data Link Control*.

secondary logical unit (SLU). In SNA, the logical unit (LU) that contains the secondary half-session for a particular LU-LU session. Contrast with primary logical unit. See also *logical unit*.

session. In SNA, a logical connection between two network addressable units (NAUs) that can be activated, tailored to provide various protocols, and deactivated as requested. The session activation request and response can determine options for such things as rate and concurrency of data exchange, control of contention and error recovery, and characteristics of the data stream. Sessions compete for network resources, such as links within the path control network. For routing purposes, each session is identified by the network or local addresses of the session partners. See *LU-LU session, LU-LU session type*.

session limit. In SNA, the maximum number of concurrently active LU-LU sessions that a particular logical unit (LU) can support.

side information. System-defined values required by CPI-C to initialize a conversation. These values include the partner LU name, mode name, and transaction program name. See *Common Programming Interface for Communication (CPI-C)*.

single-byte character set (SBCS). A character set in which each character is represented by 1 byte. Contrast with double-byte character set.

SNA gateway. A gateway that handles multiple APPC communication sessions simultaneously. See *gateway*.

SNA sense data. An SNA-defined encoding of error information In SNA, the data sent with a negative response, indicating the reason for the response.

SNASVCMG mode name. The SNA service manager mode name. This is the architecturally-defined mode name identifying sessions on which CNOS is exchanged. Most APPC-providing products predefine SNASVCMG sessions.

subnet. In TCP/IP, a part of a network that is identified by a portion of the Internet address. Synonym for subnetwork.

subnet address. In Internet communications, an extension to the basic IP addressing scheme where a portion of the host address is interpreted as the local network address.

Synchronous Data Link Control (SDLC). A communications protocol for managing synchronous, code-transparent serial-by-bit information transfer over a link connection.

Systems Application Architecture (SAA). A set of IBM software interfaces, conventions, and protocols that provide a framework for designing and developing applications across multiple computing environments.

Systems Network Architecture (SNA). The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through, and controlling the configuration and operation of, networks.

T

TCP62. A protocol mapper provided by AnyNet that allows partner LU62 applications to communicate over a TCP/IP network.

TCP/IP. Transmission Control Protocol/Internet Protocol.

terminal emulation. The capability of a microcomputer or personal computer to operate as if it were a particular type of terminal linked to a processing unit and to access data.

token ring. A network with a ring topology that passes tokens from one attaching device to another; for example, the IBM Token-Ring network.

trace. A record of data that provides a history of events that occurred in a system. The process of recording the sequence in which the statements in a program are executed and, optionally, are the values of the program variables used in the statements.

transaction program. A program that uses the APPC application program interface (API) to communicate with a partner application program in the same node or at a partner node.

Transmission Control Protocol/Internet Protocol (TCP/IP). A set of communications protocols that support peer-to-peer connectivity functions for both local and wide area networks. TCP/IP can be used for client/server links between IBM CICS Universal Clients and, for example, CICS for OS/2, CICS for Windows NT.

type 2.0 node. An SNA node that attaches to a subarea network as a peripheral node and provides full end-user services but no intermediate routing services.

type 2.1 node. An SNA node that can be configured as an endpoint or intermediate routing node in a network, or as a peripheral node attached to a subarea network.

U

user session. Any APPC session other than a SNASVCMG session.

V

verb. A reserved word that expresses an action to be taken by an application programming interface (API), a compiler, or an object program. In SNA, the general name for a transaction program's request for communication services.

W

wide area network (WAN). A network that provides communication services to a geographic area larger than that served by a local area network or a metropolitan area network, and that may use or provide public communication facilities.

Index

Numerics

- 3270 data streams 101, 106
- 3745 cluster controller 8

A

- Adapter configuration setting 64
- adapters 15
- advanced program-to-program communication (APPC) 13, 40
- Anynet domain name suffix configuration setting 62
- APAR (authorized program analysis report)
  - authorization 141
  - closing 142
  - documentation needed 138
  - process 141
  - submitting 141
- APPC (advanced program-to-program communication) 13, 40
- application compatibility 159
- Applid configuration setting 52
- Applid configuration setting (NetBIOS) 63
- ATI 155
- authenticated RPC 119

B

- benefits of CICS Universal Clients 3
- Biasing configuration setting 65
- binary trace formatter 144
- books 167
  - CICS Transaction Gateway and CICS Universal Clients library 167
  - online 170
  - PDF 171
  - printed 171
- bthdinst 57, 94
- business logic 4

C

- CCIN 152
- CCLAUTH utility 120, 156
- CCLCLNT.EXE 155
- CCLLOG.HLP 149
- cclmeci.dll 130
- cclmepi.dll 131
- CCLMSG.HLP 149
- CCLOEPI program 30
- CCLSNWTP.EXE 156
- Cell Directory Service (CDS) 116
- CEMT transaction 156
- CICS printer icon 108
- CICS region configuration setting 64
- CICS server problem determination 150
- CICS server PTF requirements 20
- CICS servers 1
- CICS servers, connections 7
- CICS telnet daemon 112
- CICS Universal Client
  - trace analysis 145
- CICS Universal Client for Windows 98
  - operation 84
- CICS Universal Client for Windows NT
  - operation 84
  - running as a Windows NT service 84
- CICS Universal Clients icons 87
- cicscli.bin 142
- CICSCLI command 88
- cicscli.log 55, 149
- cicscli.trc 142
- CICSCOL environment variable 78
- CICSFTRC utility 144
- CICSKEY environment variable 74
- CICSPRNT command 106
- CICSTELD command 112
- CICSTERM command 101
- CICSTERM.INI 112
- client control 6
- client control process, restarting 90
- client control process, starting 88
- client control process, stopping 89
- client initialization file
  - specifying which to use 90
- client/server communication, setting up 31
- client/server connections 11
- client trace 90, 143
- client trace file 145
- Client trace file configuration setting 68
- coaxial connections 8
- code pages 20
- Codepage identifier override configuration setting 54
- color mapping file 78, 101
- command icons, creating 87
- commands
  - CICSCLI 88
  - CICSPRNT 106
  - CICSTERM 101
- COMMAREA 4
- communication
  - problems 150
  - setting up a client 31
- communication, mobile 9
- communication, setting up client/server connection 31
- communication protocols
  - APPC 13
  - NetBIOS 13
  - TCP/IP 13
- compatibility, ECI and EPI 159
- configuration
  - configuration file 48
  - configuration tool 48
- configuration conversion tool 69
- configuration examples 31
- configuration file 48
- configuration settings
  - Adapter 64
  - Anynet domain name suffix 62
  - Applid 52, 63
  - Biasing 65
  - CICS region 64
  - Client trace file 68
  - Codepage identifier override 54
  - Connection timeout 59
  - DCE 58
  - Description 56
  - Enable popups 56
  - Enable the Workload Manager 64
  - Fully qualified CP name or template 62
  - Gradient PC-DCE 56, 64
  - Hostname or IP address 59
  - IBM DCE 55, 64
  - Initial transaction 57

© Copyright IBM Corp. 1994, 1999

configuration settings *(continued)*
- IP address mask for CP name (optional) 62
- IP address mask for LU name template (optional) 61
- Local LU name 60
- Local LU name or template 60
- Log file 55
- Manage workload for this program 67
- Maximum buffer size 52
- Maximum Client wrap size 68
- Maximum logical SNA sessions 63
- Maximum requests 53
- Maximum servers 34, 53
- Maximum SNA RU size 63
- Mode name 60, 61
- Model terminal definition 57
- Named Pipe 58
- Named Pipe Name 64
- NetBIOS 58
- Partner LU name 60
- Port 59
- Print command 53
- Print file 54
- Region timeout 65
- Remote node inactivity timeout 63
- Round robin 65
- Send TCP/IP Keepalive packets 59
- Server name 56
- Server retry interval 55
- SNA 58
- SNA pacing size 63
- TCP/IP 58
- TCP62 58
- Terminal exit 52
- Trace 67
- Trace settings 143
- Use DCE CDS 55
- Use LU alias names 60
- Use OEM codepage 56
- Use upper case security 58
- Use Windows credentials for security 57

configuration tool 48, 132
connecting to CICS servers 89, 97
Connection timeout configuration setting 59
controller, 3745 8
coordinated security sign-on 118
Corrective Service Diskette (CSD) 142
corrective service software 26
creating icons for clients 30
CRSR transaction 40
CSD (Corrective Service Diskette) 142
ctgcfg command 48
CTGCONV, conversion tool 69
ctgjava command 48
CTIN 152
customizing
- command icons 87
- keyboard 74
- screen colors 78

D

data encryption 116
DBCS (double-byte character set) 17
DCE (Distributed Computing Environment) 115
DCE configuration setting 58
DCE principal 120
DCE support
- authenticated RPC 119
- CCLAUTH utility 120, 156
- CDS 116
- Configuration settings 116
- coordinated security sign-on 116, 118
- data encryption 116, 121
- functions provided 116
- locating servers 116
- obtaining the principal 120
- remote procedure calls 115
- RPC 118
- unauthenticated RPC 119

defining 3270 printer terminal emulator characteristics 107
defining 3270 terminal emulator characteristics 101
definition checklists 31
Description configuration setting 56
disabling the display of messages 95
disabling the display of pop-up messages 95
disk space required 15, 16
displaying command syntax 96
Distributed Computing Environment (DCE) 115
documentation 167
- HTML 170
- PDF 171

documentation of problems 138
double-byte character set (DBCS) 17
dynamic name generation (TCP62) 62

E

ECI (External Call Interface) 4
ECI and EPI compatibility 159
Enable popups configuration setting 56
Enable the Workload Manager configuration setting 64
enabling the display of pop-up messages 95
eNetwork Communications Server 41
environment variables
- CICSCOL 78
- CICSKEY 74
- PATH 25

EPI (External Presentation Interface) 5
error log
- cicscli.log 149
- client error log 149
- server error log 150

error processing 149
ESI (External Security Interface) 5
Euro support 54, 56
example list of CICS servers 95
EXEC CICS RETURN TRANSID IMMEDIATE command 103, 108
External Call Interface (ECI) 4
External Presentation Interface (EPI) 5
External Security Interface (ESI) 5

F

FAATPPCI 153
font selection on terminal emulators 111
fonts, selecting on terminal emulators 111
free memory 52
Fully qualified CP name or template configuration setting 62

G

gateway 8
glossary of terms and abbreviations 177
Gradient PC-DCE configuration setting 56, 64
graphical user interface (GUI) 5
GUI (graphical user interface) 5

H

hardcopy books 171
hardware requirements 15
Hostname or IP address configuration setting 59
HTML (HyperText Markup Language) 170
HTML documentation, viewing 170
HyperText Markup Language (HTML) 170

I

IBM DCE configuration setting 55, 64
icons, client 30
icons, for CICSCLI commands 87
initial transaction 101
Initial transaction configuration setting 57
initialization files
- cicscol.ini 78, 79
- cicskey.ini 74
- CICSTERM.INI 112

installation
- CICS Universal Client on Windows 22
- compact installation 22
- custom installation 22
- response file 27
- setup.log 28
- silent installation 27
- types of installation 22
- typical installation 22
- uprading silently 28

IP address, TCP/IP 34
IP address mask for CP name (optional) configuration setting 62
IP address mask for LU name template (optional) configuration setting 61

J

Java virtual machine (JVM) 48
JVM (Java virtual machine) 48

K

KeepAlive packets 39
keyboard mapping file 74, 101
keys specific to particular keyboards 76

L

LAN Distance 9
language support for client applications 18, 20
listing connected servers 95
Local LU name configuration setting 60
Local LU name or template configuration setting 60
Local Named Pipe protocol 13
locating servers 116
Log file configuration setting 55
logical unit (LU) 40
LU6.2 13

M

Manage workload for this program configuration setting 67
Maximum buffer size configuration setting 52
Maximum Client wrap size configuration setting 68
Maximum logical SNA sessions configuration setting 63
Maximum requests configuration setting 53
Maximum servers configuration setting 53
Maximum SNA RU size configuration setting 63
memory requirements 52
messages, disabling the display of 95
Microsoft SNA Server 40
migration 159
mobile communication 9
mode definitions, APPC 40
Mode name configuration setting 60, 61
Model terminal definition configuration setting 57
multiple protocol configurations 9

N

Named Pipe configuration setting 58
Named Pipe Name configuration setting 64
Named Pipes protocol 46
NetBIOS (Network Basic Input/Output System) 13, 33
NetBIOS configuration setting 58
NetBIOS listener 154
NetBIOS resources 34
NetWare for SAA 44
network adapters 15
Network Basic Input/Output System (NetBIOS) 13
network drives, problems with 25
Network Provider Interface (NPI) 94
non-existent directories 25
NPI (Network Provider Interface) 94

O

object oriented programming 3
online books, PDF 171
online documenatation, HTML 170
online help, for end user messages, 149
onlne help, for trace and log messages 149
operation
- CICS Universal Client for Windows 98 84
- CICS Universal Client for Windows NT 84

options
- CICSCLI command 97
- CICSPRNT command 108
- CICSTERM command 104

P

Partner LU name configuration setting 60
password expiry management (PEM) 94
PATH environment variable 25
PDF (Portable Document Format) 171
PDF books, viewing 171
PEM (password expiry management) 94
planning 7
PMR (Problem Management Record) 136
pop-up messages, disabling the display of 95
pop-up messages, enabling the display of 95
Port configuration setting 59
port number, TCP/IP 34
Portable Document Format (PDF) 171
PostScript books 171
presentation logic 4
Print command configuration setting 53
print file, processing 102, 107
Print file configuration setting 54
print terminal emulator, starting 107
printer support 5
printer terminal emulator characteristics, defining 107
problem determination 135

Problem Management Record (PMR) 136
problem reporting
  documentation of problems 138
  information needed 139
  report sheet 139
  support organization 136
problems, common 152
problems, communication 150
program support 136
programming languages 18, 20
protocol configurations, multiple 9
PTF (Program Temporary Fix) 142
publications, CICS Transaction Gateway and CICS Universal Clients library 167

R

railroad diagrams xiv
Region timeout configuration setting 65
registering COM libraries 29
REGSVR32 program 30
Remote node inactivity timeout configuration setting 63
remote procedure calls 115
requirements, hardware 15
requirements, software 17
response file, setup.iss 27
restarting the client 90
restrictions, using CICS servers 12
RETAIN database 136
RETAIN problem management system
  APARs 141
  Problem Management Record 136
Round robin configuration setting 65
RPC, authenticated 119
RPC, unauthenticated 119

S

SDLC connections 8
security 58, 91
  Network Provider Interface (NPI) 94
Send TCP/IP Keepalive packets configuration setting 59
Server name configuration setting 56
Server retry interval configuration setting 55
servers, listing 95
services, Windows NT 84
SERVICES file, TCP/IP 34
setting up client/server communication 31
signon capable terminals 20
silent installation 27
SNA (Systems Network Architecture) 13
SNA configuration setting 58
SNA gateway 7
SNA pacing size configuration setting 63
SNA Server 40
softcopy books, PDF 171
software requirements 17
starting a 3270 print terminal emulator 107
starting a 3270 terminal emulator 101
starting the client control process 88
Startup parameters for the CICS services 86
stopping a terminal emulator 102
stopping the client control process 89
support organization
  problem reporting 139
  program support 136
  sending documentation to 141
syntax notation xiv
Systems Network Architecture (SNA) 13

T

TCP/IP (Transmission Control Protocol/Internet Protocol) 13, 34
TCP/IP configuration setting 58
TCP62 configuration setting 58
TCP62 protocol 37
TCP62 support 14
telnet support 112
terminal emulation 4
terminal emulator, starting 101
terminal emulator, stopping 102
terminal emulator characteristics, defining 101
terminal emulator settings 112
Terminal exit configuration setting 52
testing the client/server link 82
token ring connections 8
Trace configuration setting 67
trace file 142
Trace settings, configuration tool 143
tracing IBM CICS Universal Clients 142
transaction programs (TP), APPC 40
transmission control protocol/internet protocol (TCP/IP) 13
Transmission Control Protocol/Internet Protocol (TCP/IP) 13, 34
troubleshooting 152

U

unauthenticated RPC 119
unregistering COM libraries 29
updating
  installed software 26
Use DCE CDS configuration setting 55
Use LU alias names configuration setting 60
Use OEM codepage configuration setting 56
Use upper case security configuration setting 58
Use Windows credentials for security configuration setting 57

V

viewing online documentation 170
VTAM
  buffer trace 151

W

Windows NT service 84
WINSOCK 154
Workload Manager 50, 64
  biasing algorithm 129
  configuring 132
  installation 131
  operational characteristics 132
  overview 125
  port 126
  programs 126
  regions 126
  round-robin algorithm 129
  server instance 126
  TCP/IP port 126
  workload management 126

Program Number: 5648-B42

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

SC34-5449-01