CICS® Universal Clients

# Messages

*Version 3.1*

GC34-5452-01

CICS® Universal Clients

# Messages

*Version 3.1*

GC34-5452-01

**Note!**

Before using this information and the product it supports, be sure to read the general information under “Appendix. Notices” on page 59.

**Second edition (September 1999)**

This edition applies to Version 3.1 of IBM CICS Universal Client program number 5648-B42.

This edition replaces GC34-5452-00. It will also apply to all subsequent versions, releases, and modifications until otherwise indicated in new editions.

# Contents

# About this book

This book lists the user messages, error log messages, and trace log messages for CICS Universal Client Version 3.1.

It is intended for use as a quick reference, with messages in each chapter organized in alphanumeric sequence. Each message entry gives the message identifier, message text, and further diagnostic and explanatory information.

## Conventions and terminology used in this book

The terminology in this book should be familiar to anyone who has used CICS and the supported operating systems.

## Prerequisite and related information

The messages refer to several non-IBM books; these mainly relate to the platform on which the client is running and it is assumed that these books are available to you.

## 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. User messages

**CCL0510E Command option *option* is invalid**

**Explanation:** An invalid command option was specified on the command line.

**System Action:** The command line application terminates.

**User Response:** Retype the command using the correct options.

**CCL0511E Command option *option* must specify a value**

**Explanation:** A command line application was run with no value supplied for the named required option.

**System Action:** The command line application terminates.

**User Response:** Retype the command using the correct options.

**CCL0512E Command option *option* cannot be used more than once**

**Explanation:** The named option was used more than once on the command line. This is invalid.

**System Action:** The command line application terminates.

**User Response:** Retype the command using the correct options.

**CCL0513E Command option *option1* cannot be used with option *option2***

**Explanation:** The two named options, used on the command line, were incompatible.

**System Action:** The command line application terminates.

**User Response:** Retype the command using the correct options.

**CCL0514E Command option *option1* or *option2* must be provided**

**Explanation:** The command line did not include one of the named required options.

**System Action:** The command line application terminates.

**User Response:** Retype the command using the correct options.

**CCL0515E Command option *option1* can only be used with *option2***

**Explanation:** *option1* was specified on the command line, but *option2* was not also specified. This is invalid.

**System Action:** The command line application terminates.

**User Response:** Retype the command using the correct options.

**CCL1021E Unable to write to text file *filename***

**Explanation:** The binary trace formatter is unable to write any further information to the CICS client trace file during conversion of the binary trace.

**System Action:** The binary trace formatter terminates.

**User Response:** Determine the cause of the error. It may be that the disk which the binary trace formatter is writing to is full, or the disk might be write-protected.

**CCL1022E Unable to read from binary file *filename***

**Explanation:** The binary trace formatter is unable to read from the binary trace file during conversion of the binary trace.

**System Action:** The binary trace formatter terminates.

**User Response:** Determine the cause of the error. It may be that the binary trace file has been removed during conversion, or the binary trace file might be read-protected.

**CCL1023E Unable to open file *filename***

**Explanation:** The binary trace formatter is unable to open the specified file for reading or writing as required.

**System Action:** The binary trace formatter terminates.

**User Response:** Determine the cause of the error. If the file that is being opened is the binary trace file, check that the file exists and that it is not read-protected. If the file that is being opened is the output text file, make sure that the disk on to which the file is being written is not write-protected.

**CCL1024E Unable to correctly interpret all bytes of file *filename***

**Explanation:** The binary trace formatter is unable to process the specified binary trace file.

**System Action:** The binary trace formatter terminates.

**User Response:** Determine the cause of the error. Make sure that the file being read is a valid binary trace file, and that it was created with the version of the CICS

client currently installed on your system. If after checking this you still receive the same error, you may need to remove the binary trace file before running the client trace again.

---

**CCL1025E Not enough memory to convert binary trace file**

**Explanation:** The binary trace formatter has run out of free memory whilst processing the binary trace.

**System Action:** The binary trace formatter terminates.

**User Response:** Increase the amount of available memory in the system by shutting down applications that are not needed, and then re-run the binary trace formatter. You can reduce the amount of memory that the binary trace formatter needs by specifying on the command line the maximum amount of data to convert at any one time (refer to the documentation for further information on how to do this).

---

**CCL1027E Bad version number detected in** *filename*

**Explanation:** The binary trace formatter detected an invalid version number in the binary trace file.

**System Action:** The binary trace formatter terminates without performing any further processing.

**User Response:** Make sure that the file you are trying to convert is a valid binary trace file and that you are using the most recent version of the binary trace formatter that is available on your system.

---

**CCL1036I Warning: component** *component* **was not found in the binary trace**

**Explanation:** The binary trace formatter was told to process trace points from a named component, but no trace points from that component were found in the binary trace file.

**System Action:** This message is for information only. The binary trace formatter will continue to process the binary trace file.

**User Response:** In order to stop this message being displayed, make sure that all the components you specify on the command line of the binary trace formatter exist in the binary trace.

---

**CCL1037I [Bytes Remaining -** *bytes* **(***percentage***)]**

**Explanation:** The binary trace formatter has been started and is in the process of converting the binary trace. The message shows the progress that the binary trace formatter has made so far.

**System Action:** This message is for information only. The binary trace formatter will continue to process the binary trace file.

**User Response:** No further action is required

---

**CCL1038I** *filename* **already exists. Do you want to overwrite it?**

**Explanation:** The binary trace formatter has detected that the trace file you have told it to generate already exists. You are given the opportunity to erase it or terminate the formatter.

**System Action:** If you enter *Y* at the command prompt, the binary trace formatter will continue and the old trace file will be overwritten. If you enter *N* at the command prompt, the binary trace formatter will terminate and the old trace file will remain unchanged.

**User Response:** Select the appropriate option according to whether you want to overwrite the old trace file or terminate the binary trace formatter.

---

**CCL1100I The following list shows what components can be traced**

**Explanation:** A request to list all client components has been given.

**System Action:** A list of all client components is displayed on the screen. An *X* is placed beside each component that is currently being traced.

**User Response:** No further action is required

---

**CCL1120I Trace has not been started**

**Explanation:** A command that required tracing to be turned on was given, but at the moment tracing is disabled.

**System Action:** Some client commands that require tracing to be turned on will fail, but most requests to the client will be unaffected.

**User Response:** If you want to issue requests to the client that rely on tracing, tracing must be turned on.

---

**CCL1121I Component trace started to:** *component list*

**Explanation:** A request to trace only those components specified on the command line was given.

**System Action:** The client runs as normal, writing all trace points from the specified components to the binary trace file.

**User Response:** No further action is required.

---

**CCL1122E Invalid list of components. Please re-type.**

**Explanation:** You have specified a list of components on the command line, but one or more of the components in the list is invalid.

**System Action:** The request that required a list of

components to be specified is ignored.

**User Response:** Refer to the CICS client documentation for a list of valid components and retype the request.

---

**CCL1123I Reading binary trace file** *filename*

**Explanation:** The binary trace formatter is about to process the specified binary trace file.

**System Action:** This message is for information only. The binary trace formatter will continue.

**User Response:** No further action is required

---

**CCL1124I Reading binary trace file** *filename1* **and** *filename2*

**Explanation:** The binary trace formatter is about to process the specified binary trace files. The binary trace formatter is reading from two trace files because it is formatting a wrapping trace. Refer to the documentation for further information about wrapping traces.

**System Action:** This message is for information only. The binary trace formatter will continue.

**User Response:** No further action is required

---

**CCL1125I Generating text trace file** *filename*

**Explanation:** The binary trace formatter will write all text formatted output to the specified file.

**System Action:** This message is for information only. The binary trace formatter will continue.

**User Response:** No further action is required

---

**CCL2001E Server** *server* **is undefined**

**Explanation:** The client received a request for a server, but the server is not defined in the client initialization file that was specified when the client first started up.

**System Action:** The request fails but the client continues.

**User Response:** Ensure the server name specified by the ECI or EPI application, or by the 3270 Emulator is correct. If required, add the server name to the client initialization file, then stop and restart the client to activate the changes.

---

**CCL2003I Server** *server* **requires a security identification**

**Explanation:** The client received a request destined for a secure server. The server required a valid userid and password before it could accept the request, and was supplied either with an invalid one or none at all.

**System Action:** The client displays a pop-up panel allowing a userid and password to be entered or the request to be cancelled.

**User Response:** Enter a valid userid and password to allow the request to be processed by the server. This message can be avoided either by ensuring that an ECI application supplies a valid userid and password or by using the CICSCLI /C command to set the server security.

---

**CCL2004E The client cannot continue**

**Explanation:** An error was detected.

**System Action:** The client terminates.

**User Response:** Examine any other messages and the client error log to determine the cause of the error.

---

**CCL2005I The client will continue**

**Explanation:** An error was detected.

**System Action:** Client processing continues.

**User Response:** Examine any other messages and the client error log to determine the cause of the error.

---

**CCL2006I Refer to the client error log for more details**

**Explanation:** An error has occurred for which further useful diagnostic information has been written to the client error log.

**System Action:** Error information is written to the client error log.

**User Response:** Examine the client error log to obtain the information.

---

**CCL2007E Free memory is exhausted**

**Explanation:** The client ran out of memory while processing a request.

**System Action:** The request fails and the client attempts to continue processing.

**User Response:** Try altering the values in the client initialization file to provide the client with a larger free memory pool or to reduce the overall memory requirements.

---

**CCL2008E An internal state error occurred**

**Explanation:** The client detected an unexpected internal error.

**System Action:** Error information is written to the client error log.

**User Response:** Obtain the client error log - and, if possible, trace information - and refer the problem to your support organization.

**CCL2009E An unexpected error has occurred**

**Explanation:** The client detected an unexpected error.

**System Action:** Error information is written to the client error log.

**User Response:** Obtain the client error log - and, if possible, trace information - and refer the problem to your support organization.

**CCL2052E Cannot open error log file** *file name*

**Explanation:** The client could not open or write to the specified error log file.

**System Action:** Log messages are lost.

**User Response:** Check that the name specified in the LogFile parameter in the client initialization file is correct and accessible.

**CCL2053I Cannot open trace file** *file name* **- tracing has been disabled**

**Explanation:** The client could not open or write to the specified trace file.

**System Action:** Trace information is lost.

**User Response:** Check that the name specified in the TraceFile parameter in the client initialization file is correct and accessible.

**CCL2054I A CICS server error has occurred**

**Explanation:** An error occurred on a CICS server with which the client was trying to communicate.

**System Action:** Error information is written to the client error log; client processing continues.

**User Response:** Examine any other messages and the client error log to determine the cause of the error.

**CCL2087I Server** *server***, User ID invalid, please re-enter**

**Explanation:** The userid entered in the previous security pop-up panel is not known to the receiving security system.

**System Action:** The client displays a pop-up panel allowing a userid and password to be re-entered or the request to be cancelled.

**User Response:** Enter a valid userid and password to allow the request to be processed by the server.

**CCL2088I Server** *server***, Password invalid, please re-enter**

**Explanation:** The password entered in the previous security pop-up panel is incorrect.

**System Action:** The client displays a pop-up panel allowing a userid and password to be re-entered or the request to be cancelled.

**User Response:** Enter a valid userid and password to allow the request to be processed by the server.

**CCL2089I Server** *server***, Password expired, please change**

**Explanation:** The password entered in the previous security pop-up panel has expired.

**System Action:** The client displays a pop-up panel allowing the password to be changed or the request to be cancelled.

**User Response:** Enter the expired password, a new password and confirm the new password to allow the request to be processed by the server.

**CCL2090I Server** *server***, New Password unacceptable, please re-enter**

**Explanation:** The new password entered in the previous security pop-up panel is unacceptable to the receiving security system.

**System Action:** The client displays a pop-up panel allowing a new password to be supplied or the request to be cancelled.

**User Response:** Enter the expired password, a new password and confirm the new password to allow the request to be processed by the server.

**CCL2091I New and Confirm Passwords do not match, please re-enter**

**Explanation:** The new password and confirm passwords entered in the previous security pop-up panel do not match.

**System Action:** The client displays a pop-up panel allowing a new password to be supplied or the request to be cancelled.

**User Response:** Enter the expired password, a new password and confirm the new password to allow the request to be processed by the server.

**CCL3001E Errors detected at line** *number* **in file** *file name*

**Explanation:** Errors were detected at the specified line in the client initialization file.

**System Action:** The client terminates.

**User Response:** Correct the initialization file and restart the client.

**CCL3002E Cannot find client initialization file** *file name*

**Explanation:** The specified client initialization file could not be found.

**System Action:** The client terminates.

**User Response:** Ensure that the required initialization file exists and is correctly specified when starting the client.

**CCL3003E Missing = sign after parameter** *parameter*

**Explanation:** An = sign was expected after an initialization file parameter in the client initialization file.

**System Action:** The client terminates.

**User Response:** Correct the initialization file error and restart the client.

**CCL3004E Parameter** *text* **not recognized**

**Explanation:** An unknown parameter starting with specified text was encountered in the client initialization file.

**System Action:** The client terminates.

**User Response:** Correct the initialization file error and restart the client.

**CCL3005E Parameter** *parameter* **repeated or misplaced**

**Explanation:** A valid parameter was encountered in the wrong position in the client initialization file.

**System Action:** The client terminates.

**User Response:** Correct the initialization file error and restart the client.

**CCL3006E Parameter** *parameter* **has invalid value** *value*

**Explanation:** A valid parameter had an incorrect value specified in the client initialization file.

**System Action:** The client terminates.

**User Response:** Correct the initialization file error and restart the client.

**CCL3007E Inconsistent values for parameters** *parameter 1* **and** *parameter 2*

**Explanation:** The values of certain parameters specified in the client initialization file have dependencies on each other. The values for the two specified parameters are inconsistent.

**System Action:** The client terminates.

**User Response:** Correct the initialization file error and restart the client.

**CCL3008E Required parameter** *parameter* **missing**

**Explanation:** The specified parameter is missing from the client initialization file and must be provided.

**System Action:** The client terminates.

**User Response:** Correct the initialization file error and restart the client.

**CCL3009E Not enough memory for client initialization**

**Explanation:** There was not enough free memory to process the client initialization file.

**System Action:** The client terminates.

**User Response:** Try altering the values in the client initialization file to provide the client with a larger free memory pool or to reduce the overall memory requirements.

**CCL3010E Missing driver definition for protocol** *protocol*

**Explanation:** The value of the protocol parameter in a Server section of the client initialization file must match with the name of a Driver section. No matching driver definition was found for the named protocol.

**System Action:** The client terminates.

**User Response:** Correct the initialization file error and restart the client.

**CCL3011E Errors detected while reading file** *file name*

**Explanation:** Errors were encountered while reading the client initialization file.

**System Action:** The client terminates.

**User Response:** Correct the initialization file error and restart the client.

**CCL3012E Duplicate** *section* **section definition**

**Explanation:** A Client, Server, or Driver section in the client initialization file had the same name as a previously specified section.

**System Action:** The client terminates.

**User Response:** Correct the initialization file error and restart the client.

**CCL3013E Unable to access file** *file name*, **rc=***rc*

**Explanation:** Errors were encountered while accessing file *file name*.

**System Action:** The client terminates.

**User Response:** Change the file permissions and restart the client.

**CCL3109E Cannot connect to server** *server* **- client is already installed**

**Explanation:** The client name specified in the Client section of the initialization file was already installed on the specified server.

**System Action:** The server rejects the client's installation request.

**User Response:** Ensure all client names are unique within the network or specify the name as ″*″ to enable a suitable unique name to be generated automatically.

**CCL3110E Cannot connect to server** *server* **- server is busy**

**Explanation:** The specified server was busy and could not process the client installation request.

**System Action:** The server rejects the client's installation request.

**User Response:** Retry the request sometime later when the server is not so busy. If the problem persists contact your server administrator.

**CCL3111E Cannot connect to server** *server* **- server rejected install request**

**Explanation:** The specified server could not process the client installation request.

**System Action:** The server rejects the client's installation request, server continues processing.

**User Response:** Examine the client error log and, if possible, trace information on both the client and server to determine the cause of the error.

**CCL3112E Cannot connect to server** *server* **- request rejected by server exit**

**Explanation:** A user exit running on the specified server rejected the request to install the client definition.

**System Action:** The server rejects the client's installation request, the client continues processing.

**User Response:** Check with the server system administrator to determine why the client installation request was rejected.

**CCL3121E Cannot connect to server** *server* **- invalid codepage specified**

**Explanation:** The codepage value sent to the server was rejected.

**System Action:** The server rejects the client's installation request, the client continues processing.

**User Response:** If the CCSID parameter has been specified in CTG.INI, check that the value specified is valid. Otherwise, check with the server system administrator to determine why the codepage value was rejected.

**CCL3229E Cannot load protocol driver** *driver*

**Explanation:** The specified communications protocol driver module could not be found or loaded.

**System Action:** No communication can take place using the protocol, the client continues processing.

**User Response:** Examine the client error log and, if possible, trace information to determine the cause of the error.

**CCL3280E Errors occurred while initializing protocol driver** *driver*

**Explanation:** The specified communications protocol driver module reported errors during startup.

**System Action:** No communications can take place using the protocol, the client continues processing.

**User Response:** Examine the client error log and, if possible, trace information to determine the cause of the error.

**CCL3281E Errors occurred while terminating protocol driver** *driver*

**Explanation:** The specified communications protocol driver module reported errors while it was closing down.

**System Action:** The error is ignored and the client continues processing.

**User Response:** Examine the client error log and, if possible, trace information to determine the cause of the error.

**CCL3282E Errors occurred while connecting to a server using protocol driver** *driver*

**Explanation:** The specified communications protocol driver module reported errors while it was trying to connect to a server.

**System Action:** The client cannot communicate with the server, the client continues processing.

**User Response:** Examine the client error log and, if

possible, trace information to determine the cause of the error.

**CCL3283E Errors occurred while disconnecting from a server using protocol driver** *driver*

**Explanation:** The specified communications protocol driver module reported errors while it was closing a connection with a server.

**System Action:** The error is ignored and client continues processing.

**User Response:** Examine the client error log and, if possible, trace information to determine the cause of the error.

**CCL3284E Errors occurred while starting a conversation with a server using protocol driver** *driver*

**Explanation:** The specified communications protocol driver module reported errors while it was starting a new conversation with a connected server.

**System Action:** The request to start the conversation fails and the client continues processing.

**User Response:** Examine the client error log and, if possible, trace information to determine the cause of the error.

**CCL3285E Errors occurred while closing a conversation with a server using protocol driver** *driver*

**Explanation:** The specified communications protocol driver module reported errors while it was closing a conversation with a connected server.

**System Action:** The request to close the conversation fails and the client continues processing.

**User Response:** Examine the client error log and, if possible, trace information to determine the cause of the error.

**CCL3286E Errors occurred while sending data to a server using protocol driver** *driver*

**Explanation:** The specified communications protocol driver module reported errors while it was sending data over a conversation with a connected server.

**System Action:** The request that sent the data fails and the client continues processing.

**User Response:** Examine the client error log and, if possible, trace information to determine the cause of the error.

**CCL3287E Errors occurred while receiving data from a server using protocol driver** *driver*

**Explanation:** The specified communications protocol driver module reported errors while it was receiving data from a conversation with a connected server.

**System Action:** The request that required the data fails and the client continues processing.

**User Response:** Examine the client error log and, if possible, trace information to determine the cause of the error.

**CCL3288E Errors occurred while communicating with a server using protocol driver** *driver*

**Explanation:** The specified communications protocol driver module reported errors while it was communicating or trying to communicate with a server.

**System Action:** The request that initiated the communications fails and the client continues processing.

**User Response:** Examine the client error log and, if possible, trace information to determine the cause of the error.

**CCL6109I Port:** *port* **Users:** *number of users*

**Explanation:** This status message indicates the number of active users of the CICS Client telnet service.

**System Action:** The telnet demon continues processing.

**User Response:** No further action is required.

**CCL6116I The terminal with netname** *netname* **was deleted.**

**Explanation:** The terminal with the given netname was deleted because the telnet session ended.

**System Action:** The telnet demon continues processing.

**User Response:** No further action is required.

**CCL6118I Terminal with netname** *netname* **installed for user** *address***.**

**Explanation:** A telnet session has been established. The users TCP/IP address and the netname of the CICS terminal are shown.

**System Action:** The telnet demon continues processing.

**User Response:** No further action is required.

**CCL6119E Failed to install a terminal for user *address*.**

**Explanation:** The telnet demon failed to connect to the CICS server.

**System Action:** The telnet demon continues processing.

**User Response:** Check that the CICS Client can connect to the server.

**CCL6120E Only tn3270 telnet sessions are supported at present.**

**Explanation:** A telnet session has been established but the terminal is not of a tn3270 type.

**System Action:** The telnet demon continues processing. The inbound session is disconnected.

**User Response:** Use tn3270 type telnet sessions to connect to cicsteld.

**CCL7023E Command option *option* is invalid**

**Explanation:** An invalid command option was specified when starting the CICSTERM or CICSPRNT 3270 emulator.

**System Action:** The emulator terminates.

**User Response:** Restart the emulator with the correct option.

**CCL7024E Command option *option* must specify a value**

**Explanation:** A CICSTERM or CICSPRNT command was issued with no value supplied for the named required option.

**System Action:** The emulator terminates.

**User Response:** Restart the emulator using a valid value for the option.

**CCL7025E Command option *option* cannot be used more than once**

**Explanation:** On a CICSTERM or CICSPRNT command, the named option was used more than once. This is invalid.

**System Action:** The emulator terminates.

**User Response:** Restart the emulator using the correct options.

**CCL7026E Command option *option1* cannot be used with option *option2***

**Explanation:** The two named options, used on a CICSTERM or CICSPRNT command, were incompatible.

**System Action:** The emulator terminates.

**User Response:** Restart the emulator using the correct options.

**CCL7027E Command option *option1* or *option2* must be provided**

**Explanation:** A CICSPRNT command did not include one of the named required options.

**System Action:** The emulator terminates.

**User Response:** Restart the emulator using the correct options.

**CCL7028E Cannot open binding file *file name***

**Explanation:** CICSTERM could not find or open the specified color, or keyboard, binding file.

**System Action:** The emulator terminates.

**User Response:** Ensure the required binding files exist and are correctly specified, then restart the emulator.

**CCL7029E Out of memory processing binding file *file name***

**Explanation:** CICSTERM ran out of memory while processing a color, or keyboard, binding file.

**System Action:** The emulator terminates.

**User Response:** Reduce the number of bindings specified in the file or try to provide more system free memory; then restart the emulator.

**CCL7030E Error detected at line *number* in binding file *file name* - duplicate bind for item *item***

**Explanation:** CICSTERM detected an error in a color, or keyboard, binding file. Only one binding is allowed for the specified item.

**System Action:** The emulator terminates.

**User Response:** Correct the binding file and restart the emulator.

**CCL7031E Error detected at line *number* in binding file *file name* - unknown command *command***

**Explanation:** CICSTERM detected an error in a color, or keyboard, binding file. The specified command was not a valid binding command.

**System Action:** The emulator terminates.

**User Response:** Correct the binding file and restart the emulator.

**CCL7032E Error detected at line** *number* **in binding file** *file name* **- invalid function** *function*

**Explanation:** CICSTERM detected an error in a keyboard binding file. The specified item was not a valid 3270 keyboard function.

**System Action:** The emulator terminates.

**User Response:** Correct the binding file and restart the emulator.

**CCL7033E Error detected at line** *number* **in binding file** *file name* **- invalid modifier** *modifier*

**Explanation:** CICSTERM detected an error in a keyboard binding file. The specified item was not a valid workstation keyboard modifier value.

**System Action:** The emulator terminates.

**User Response:** Correct the binding file and restart the emulator.

**CCL7034E Error detected at line** *number* **in binding file** *file name* **- invalid key** *key*

**Explanation:** CICSTERM detected an error in a keyboard binding file. The specified item was not a valid workstation key name.

**System Action:** The emulator terminates.

**User Response:** Correct the binding file and restart the emulator.

**CCL7035E Error detected at line** *number* **in binding file** *file name* **- invalid attribute** *attribute*

**Explanation:** CICSTERM detected an error in a color binding file. The specified item was not a valid 3270 screen field attribute.

**System Action:** The emulator terminates.

**User Response:** Correct the binding file and restart the emulator.

**CCL7036E Error detected at line** *number* **in binding file** *file name* **- invalid color** *color*

**Explanation:** CICSTERM detected an error in a color binding file. The specified item was not a valid workstation display color.

**System Action:** The emulator terminates.

**User Response:** Correct the binding file and restart the emulator.

**CCL7037E Unable to communicate with the CICS Client**

**Explanation:** CICSTERM or CICSPRNT could not communicate with the client process. This probably means the client was unable to start.

**System Action:** The emulator terminates.

**User Response:** Try to start the client using the CICSCLI /S command.

**CCL7038E The CICS Client has not been started**

**Explanation:** CICSTERM or CICSPRNT did not start because the client was not, or could not be, started.

**System Action:** The emulator terminates.

**User Response:** Try to start the client using the CICSCLI /S command.

**CCL7039I EMULATOR - CICS server selection**

**Explanation:** The emulator was started with the /R or /S option and a server name was not specified to connect CICSTERM or CICSPRNT to.

**System Action:** A list of available server names is provided.

**User Response:** Choose the required server from the list or CANCEL to terminate the emulator.

**CCL7043E Terminal screen width is invalid - reset to 80 columns**

**Explanation:** The server returned a terminal definition that contained an invalid screen width for the terminal emulator. (Screen widths are normally 40, 80, or 132 columns.)

**System Action:** The emulator resets the screen width to a standard 80 columns. Output may appear corrupted because the client and server terminal definitions may not match.

**User Response:** Check with the server system administrator to determine why the terminal definition is incorrect.

**CCL7044E Terminal screen size is too large - reset to maximum**

**Explanation:** The server returned a terminal definition that contained a screen size too large for the terminal emulator. The product of screen width and screen depth cannot exceed 4096.

**System Action:** The emulator resets the screen depth to the maximum number of rows. Output may appear corrupted because the client and server terminal definitions may not match.

**User Response:** Check with the server system

administrator to determine why the terminal definition is incorrect.

**CCL7045E Connection lost with server** *server*

**Explanation:** The connection with the specified server has been lost. This would normally indicate that the server is no longer available.

**System Action:** The emulator waits until the server is available again and tries to reconnect.

**User Response:** Wait for the connection to be reestablished or clear the error message and terminate the emulator.

**CCL7046E The CICS Client reported a resource shortage**

**Explanation:** The client returned a resource shortage error. This would normally indicate that the client ran out of free memory. If SNA communications are being used, it may indicate that there are insufficient LU6.2 sessions available.

**System Action:** The emulator attempts to continue, although further requests to the client may receive the same error response.

**User Response:** If the client is very busy the resource shortage may disappear when the client becomes less loaded. Otherwise, examine the client error log and, if possible, trace information to determine the cause of the error. Then, stop the client and reconfigure it to provide more of the particular resource causing the problem.

**CCL7047E The CICS Client is closing down**

**Explanation:** CICSTERM or CICSPRNT could not be started because the client was closing down.

**System Action:** The emulator terminates.

**User Response:** Wait for the client to close down, then restart the emulator.

**CCL7048E Server** *server* **is undefined**

**Explanation:** The client indicated that the specified server name is unknown.

**System Action:** The emulator terminates.

**User Response:** Ensure the required server name is one of those listed in the client initialization file, or use the /S option to select from a list of valid names.

**CCL7049E Unable to connect to server** *server*

**Explanation:** The client was unable to connect to the specified server.

**System Action:** The emulator terminates.

**User Response:** Check for other messages and examine the client error log to determine the cause of the error.

**CCL7050E The MaxBufferSize setting is too small for this terminal**

**Explanation:** The emulator tried to send more data than the client could accept.

**System Action:** The data is ignored and the emulator continues.

**User Response:** Increase the MaxBufferSize value in the Client section of the client initialization file and stop, then restart, the client.

**CCL7051E Server** *server* **does not support client terminals**

**Explanation:** Client 3270 emulator terminals are not supported by all CICS servers. The specified server is one of those that does not provide this support. If this is a CICS/400 server this message can also be generated if CICS is unavailable.

**System Action:** The emulator terminates.

**User Response:** Select a server that does support client terminals or check with the server system administrator to determine why client terminals are not supported.

**CCL7052E Server** *server* **rejected this terminal with a security error**

**Explanation:** The specified server is secure and required a valid userid and password before it could install a client terminal.

**System Action:** The emulator terminates.

**User Response:** Enter a valid userid and password when prompted by the client. Alternatively, to avoid the prompt use the CICSCLI /C command to set the userid and password.

**CCL7053E Errors found while communicating with server** *server*

**Explanation:** The client encountered errors while communicating with the specified server.

**System Action:** The emulator ignores the errors and tries to continue.

**User Response:** Check for other messages and examine the client error log to determine the cause of the error.

**CCL7054I Server *server* is currently unavailable**

**Explanation:** The client found that the specified server was currently unavailable.

**System Action:** The emulator waits until the server becomes available and then tries to connect.

**User Response:** Wait for the connection to be established or clear the error message and terminate the emulator.

**CCL7055E Terminal install failed - NetName *name* is invalid**

**Explanation:** The server rejected the terminal because the specified NetName was invalid.

**System Action:** The emulator terminates.

**User Response:** Ensure the requested terminal NetName is correct and defined at the server. If the error persists, check with the server system administrator for the cause of the error.

**CCL7056E Terminal install failed - Model *model* is invalid**

**Explanation:** The server rejected the terminal because the specified Model was invalid.

**System Action:** The emulator terminates.

**User Response:** Ensure the requested terminal Model is correct and defined at the server. If the error persists, check with the server system administrator for the cause of the error.

**CCL7057E Terminal install failed - NetName *name* is already installed**

**Explanation:** The server rejected the request because another terminal was already installed with the same NetName.

**System Action:** The emulator terminates.

**User Response:** Ensure the requested terminal NetName is correct. If the error persists, check with the server system administrator for the cause of the error.

**CCL7058E Terminal install failed - rejected by the server**

**Explanation:** The server was unable to accept the terminal and rejected the installation request.

**System Action:** The emulator terminates.

**User Response:** Check with the server system administrator to determine why the terminal was rejected.

**CCL7059E Terminal install failed - reason unknown**

**Explanation:** The server rejected the terminal.

**System Action:** The emulator terminates.

**User Response:** Check with the server system administrator to determine why the terminal was rejected.

**CCL7060I DBCS datastream errors detected**

**Explanation:** The terminal emulator detected errors in the DBCS datastream.

**System Action:** The terminal emulator ignores the errors and tries to continue.

**User Response:** Check with the server system administrator to determine why incorrect data is being generated. Client trace information may be helpful when trying to find these errors.

**CCL7061E Unable to open or create print output file**

**Explanation:** The terminal emulator was unable to open or create the local print file specified by the PrintFile option in the client initialization file.

**System Action:** The print request is ignored and the terminal emulator continues.

**User Response:** Ensure the name of the print file you specified in the client initialization file is correct and accessible.

**CCL7062E Unable to write to the print output file**

**Explanation:** The terminal was unable to write to the local print file specified by the PrintFile option in the client initialization file.

**System Action:** The print request is ignored and the emulator continues.

**User Response:** Ensure the name of the print file you specified in the client initialization file is correct and accessible.

**CCL7063E Unable to execute print command**

**Explanation:** The terminal emulator is unable to execute the local print command specified by the PrintCommand option in the client initialization file.

**System Action:** The print request is ignored and the emulator continues.

**User Response:** Ensure the print command you specified in the client initialization file is correct.

**CCL7069E Server *server* has reported a transaction ABEND**

**Explanation:** The specified server reported an ABEND while trying to run the current transaction. This may have been caused by a communications failure between the client and server.

**System Action:** The terminal emulator ignores the errors and tries to continue.

**User Response:** Examine any other messages and the client error log and server error log to determine the cause of the error.

**CCL7070E Terminal screen depth is invalid - reset to 24 rows**

**Explanation:** The server returned a terminal definition that contained an invalid screen depth for the terminal emulator. (Screen depths are normally between 10 and 60 rows).

**System Action:** The terminal emulator has reset the screen depth to a standard 24 rows. Output may appear corrupted because the terminal emulator and server terminal definitions may not match.

**User Response:** Check with the server system administrator to determine why the terminal definition is incorrect.

**CCL7071E Screen data exceeds the display space available**

**Explanation:** The server returned more data than the terminal emulator can accept.

**System Action:** The excess data is ignored and the terminal emulator tries to continue.

**User Response:** Check with the server system administrator to determine why such a large amount of data is being generated. Client trace information may be helpful when trying to find these errors.

**CCL7072E The CICS Client has reached its MaxServers limit of *number***

**Explanation:** The client can only communicate simultaneously with the number of servers specified by the MaxServers option in the Client section of the client initialization file. This limit has been reached.

**System Action:** The emulator terminates.

**User Response:** Use the CICSCLI /X command to close connections to servers that are no longer required, or increase the ″MaxServers″ setting in the client initialization file; then stop and restart the client.

**CCL7073E Terminal install failed - requested terminal is not a 3270 device**

**Explanation:** The server rejected the terminal because the server terminal definition does not represent a 3270 device.

**System Action:** The emulator terminates.

**User Response:** Ensure the requested terminal Model or NetName is correct and correctly defined at the server. If the error persists, check with the server system administrator for the cause of the error.

**CCL7074E Terminal install failed - server is busy**

**Explanation:** The server rejected the terminal emulator because the server was too busy.

**System Action:** The emulator terminates.

**User Response:** Retry the request sometime later when the server is not so busy.

**CCL7098E An internal processing error has occurred**

**Explanation:** The terminal emulator detected an unexpected internal error.

**System Action:** Error information is written to the client error log.

**User Response:** Obtain the client error log - and, if possible, trace information - and refer the problem to your support organization.

**CCL7099I Refer to the error log for more details**

**Explanation:** An error has occurred for which further useful diagnostic information has been written to the client error log.

**System Action:** Client processing continues.

**User Response:** Examine the client error log to obtain the information.

**CCL7108E No CICS servers are available**

**Explanation:** There no CICS servers available for the emulator to use.

**System Action:** The emulator terminates.

**User Response:** Check that there are one or more valid CICS servers in either the CICS Client initialization file or the local DCE Cell (if the client is configured to use DCE CDS services). If load management exits are active then check that at least one of the servers the emulator tried to connect to is available. Examine the Client trace to see which servers the load management attempted to connect to.

**CCL7109E CICS_EpiAddTerminal rejected by the EPI user exit**

**Explanation:** The EPI user exit (CICSEPIX) returned CICS_EXIT_DONT_ADD_TERMINAL and rejected the CICS_EpiAddTerminal request.

**System Action:** The EPI terminal is not added.

**User Response:** Check with the server system administrator to determine why the terminal was rejected.

**CCL7136E A minimum screen size of *number* rows by *number* columns is required**

**Explanation:** The screen size available is too small for cicsterm. The minimum size is 25 rows by 80 columns.

**System Action:** The terminal emulator has not been started.

**User Response:** Change the screen size of your terminal before starting cicsterm and retry.

**CCL8024E Unable to communicate with the CICS Client**

**Explanation:** CICSCLI could not communicate with the client process. This probably means the client was unable to start.

**System Action:** CICSCLI performs no action.

**User Response:** Try starting the client using the CICSCLI /S command.

**CCL8025E The CICS Client has not been started**

**Explanation:** CICSCLI did not run because the client was not, or could not be, started.

**System Action:** CICSCLI performs no action.

**User Response:** Try starting the client using the CICSCLI /S command.

**CCL8026I Client trace is enabled**

**Explanation:** The CICSCLI /D command was issued to start client tracing.

**System Action:** Trace information is appended to the client trace file specified by the TraceFile parameter in the client initialization file.

**User Response:** You can examine the trace file by using a standard ASCII text editor.

**CCL8027I Client trace is disabled**

**Explanation:** The CICSCLI /O command was issued to stop client tracing.

**System Action:** No further trace information is written to the client trace file or to the trace memory buffer.

**User Response:** You can examine the trace and/or dump files by using a standard ASCII text editor.

**CCL8030E Unable to change client trace state**

**Explanation:** The client was unable to set the requested trace state.

**System Action:** Service tracing remains unchanged.

**User Response:** Examine any other messages and the client error log to determine the cause of the error.

**CCL8031E Server *server* is unknown**

**Explanation:** The specified server name was not defined in the client initialization file that was specified when the client first started up.

**System Action:** CICSCLI performs no action.

**User Response:** Correct the server name and reissue the command. If required add the server to the client initialization file, then stop and restart the client to activate the changes.

**CCL8032E Unable to complete stop request**

**Explanation:** The client was unable to disconnect from the server.

**System Action:** CICSCLI performs no action.

**User Response:** Try using CICSCLI /I; if this fails examine any other messages and the client error log to determine the cause of the error.

**CCL8033I Connection to server *server* is stopping**

**Explanation:** The client accepted the request to close a connection with a server and is processing that request.

**System Action:** It may take some time for the close request to complete because all active conversations with the server must complete first.

**User Response:** Wait for the connection to close; if the connection does not close you may have to force it to close using the CICSCLI /I command. This forces any open conversations to end.

**CCL8034I The CICS Client is stopping**

**Explanation:** The client accepted a request to close all server connections and to terminate.

**System Action:** It may take some time for the close request to complete because all active conversations with the server must complete first.

**User Response:** Wait for the client to stop; if the client does not stop you may have to force it to close using the CICSCLI /I command. This forces any open conversations to end.

**CCL8035I The CICS Client is already started**

**Explanation:** A request to start the client was ignored because it was already started.

**System Action:** CICSCLI performs no action.

**User Response:** No further action is required.

**CCL8036E Unable to perform start request, return code** *value*

**Explanation:** The client was unable to complete the request to start the connection to a server. The value of the internal transport error return code is given.

**System Action:** CICSCLI performs no action.

**User Response:** Examine any other messages and the client error log to determine the cause of the error.

**CCL8037I Connection to server** *server* **is being started**

**Explanation:** The client accepted a request to start a connection with a server and is processing that request.

**System Action:** It may take some time for the start request to complete because communications with the server must be established.

**User Response:** Wait for the connection to complete; you can use the CICSCLI /L command to check connection status.

**CCL8038I The CICS Client is starting**

**Explanation:** The client has accepted a request to start.

**System Action:** It may take a few moments for the start request to complete.

**User Response:** Wait for the client to start; you can use the CICSCLI /L command to check client status.

**CCL8039I The CICS Client is stopping**

**Explanation:** The client was requested to perform a function but first had to wait to terminate following an earlier CICSCLI /X command.

**System Action:** The termination processing continues.

**User Response:** Wait for the client to terminate. If the client does not close you may need to force it close using the CICSCLI /I command. This forces any open conversations to end.

**CCL8040I No servers are being used by the CICS Client**

**Explanation:** The client is active but is not connected to any servers.

**System Action:** No further action is performed.

**User Response:** No further action is required.

**CCL8042I Server** *server* **(using** *protocol* **to** *netname***) is available**

**Explanation:** The client is connected to the server and the server is available. The client is connected using the specified protocol to the system known on the network by the specified netname.

**System Action:** Processing continues.

**User Response:** No further action is required.

**CCL8043I Server** *server* **(using** *protocol* **to** *netname***) is unavailable**

**Explanation:** The client is connected to the server but the server is unavailable. The client is connected using the specified protocol to the system known on the network by the specified netname.

**System Action:** No further action is performed.

**User Response:** Wait for the server to become available, or close the connection using the CICSCLI /X command if the server is no longer required.

**CCL8044I Server** *server* **(using** *protocol* **to** *netname***) is connecting**

**Explanation:** The client is waiting to connect to the server, the state of the server is not yet known. The client is connecting using the specified protocol to the system known on the network by the specified netname.

**System Action:** The connection processing continues.

**User Response:** Wait for the connection to complete. After completion the status of the server - either available or unavailable - is known.

**CCL8045I Server** *server* **(using** *protocol* **to** *netname***) is stopping**

**Explanation:** The connection with the server is currently waiting to be closed.

**System Action:** The termination processing continues.

**User Response:** Wait for the connection to close. If the connection does not close you may need to force it to close using the CICSCLI /I command. This forces any open conversations to end.

**CCL8046E Unable to perform list request**

**Explanation:** The client was unable to perform a request to list the current server connection details.

**System Action:** CICSCLI performs no action.

**User Response:** Examine any other messages and the client error log to determine the cause of the error.

**CCL8050I Requested security information has been updated**

**Explanation:** The client completed a request to set the userid and password information for a secure server.

**System Action:** Secure server information is updated.

**User Response:** No further action is required.

**CCL8051E Invalid request to change security information**

**Explanation:** A CICSCLI command was issued with the /U or /P options but without the /C option specifying a server name.

**System Action:** CICSCLI performs no action.

**User Response:** Reissue the command with the correct options.

**CCL8052E Unable to make the requested security information changes**

**Explanation:** The client was unable to perform the request to change the userid and password information for a secure server.

**System Action:** CICSCLI performs no action.

**User Response:** Examine any other messages and the client error log to determine the cause of the error.

**CCL8053E Connection to server *server* is already started**

**Explanation:** The client was unable to perform a request to connect to the named server, because a connection with the server already exists.

**System Action:** CICSCLI performs no action.

**User Response:** If the connection is no longer required, use the CICSCLI /X command to close it.

**CCL8054E Connection to server *server* has not been started**

**Explanation:** The client was unable to perform the request to start a connection with the specified server.

**System Action:** CICSCLI performs no action.

**User Response:** Examine any other messages and the client error log to determine the cause of the error.

**CCL8055E The CICS Client has reached its MaxServers limit**

**Explanation:** The client can only communicate simultaneously with the number of servers specified by the MaxServers option in the Client section of the client initialization file. This limit has been reached.

**System Action:** CICSCLI performs no action.

**User Response:** Use the CICSCLI /X command to close connections to servers that are no longer required. Alternatively, increase the MaxServers setting in the client initialization file; then stop and restart the client.

**CCL8056E *Size* is an incorrect trace size limit value**

**Explanation:** The CICSCLI /D or CICSCLI /T command to start client tracing can include an optional numeric value specifying the maximum size of data to be traced. The specified size is either not a numeric value or is outside the permitted range of 1 - 32767.

**System Action:** CICSCLI performs no action.

**User Response:** Reissue the command with the correct value.

**CCL8057I Client error and security pop-ups have been enabled**

**Explanation:** The CICSCLI /E command was used to enable presentation of messages in pop-ups.

**System Action:** Any client errors and secure server requests are presented via a pop-up.

**User Response:** Any pop-ups have to be acknowledged by the user before further processing can continue.

**CCL8058I Client error and security pop-ups have been disabled**

**Explanation:** The CICSCLI /N command was used to disable presentation of messages in pop-ups. This allows the client to run without any user interaction.

**System Action:** Client error messages will be written to the client error log and secure server logon requests will fail.

**User Response:** No further action is required.

**CCL8059I Error and security pop-ups were already enabled**

**Explanation:** The CICSCLI /E command was used to enable error and security pop-ups but they were already enabled.

**System Action:** Error and security pop-ups remain enabled.

**User Response:** No further action is required.

**CCL8060I Error and security pop-ups were already disabled**

**Explanation:** The CICSCLI /N command was used to disable error and security pop-ups but they were already disabled.

**System Action:** Error and security pop-ups remain disabled.

**User Response:** No further action is required.

---

**CCL806IE Unable to set error and security pop-up state**

**Explanation:** The client was unable to set the security and error pop-up state requested by a CICSCLI /E or CICSCLI /N command.

**System Action:** Error and security pop-ups remain unchanged.

**User Response:** Examine any other messages and the client error log to determine the cause of the error.

---

**CCL8062I Option *option* ignored because the CICS Client is already started**

**Explanation:** The /F option to specify a client initialization file was included on a CICSCLI command but this option can be specified only when the client is first started with the CICSCLI /S command.

**System Action:** CICSCLI performs no action.

**User Response:** If a different client initialization file is required, the client must be stopped and restarted.

---

**CCL8063E Option *option* cannot be used unless starting the CICS Client**

**Explanation:** The /F option to specify a client initialization file was included on a CICSCLI command, but this option can be specified only when starting the client with the CICSCLI /S command.

**System Action:** CICSCLI performs no action.

**User Response:** If a different client initialization file is required, the client must be stopped and restarted.

---

**CCL8064I Client trace to memory is enabled**

**Explanation:** The CICSCLI /T command was issued to start client tracing to the memory buffer.

**System Action:** Trace information is written to a memory buffer which is dumped to file on request (CICSCLI /Z) or if the client terminates abnormally.

**User Response:** You can examine the dump file by using a standard ASCII text editor.

---

**CCL8065E Unable to change status of client trace to memory buffer**

**Explanation:** The client was unable to set the requested trace state.

**System Action:** Service tracing to the memory buffer remains unchanged.

**User Response:** Examine any other messages and the client error log to determine the cause of the error.

---

**CCL8066E Unable to start client trace to memory buffer**

**Explanation:** The client was unable to start the trace to the memory buffer. This is probably because sufficient storage could not be allocated.

**System Action:** Service tracing to the memory buffer was not started.

**User Response:** Examine any other messages and the client error log to determine the cause of the error.

---

**CCL8067I Client trace in memory buffer dumped to file**

**Explanation:** The CICSCLI /Z command was issued to dump any trace in the memory buffer to file. The buffer is cleared after being dumped and tracing continues.

**System Action:** Tracing to a memory buffer is enabled using the CICSCLI /T command. The trace records in the buffer are dumped to file on request (using CICSCLI /Z) or if the client terminates abnormally (some platforms).

**User Response:** You can examine the dump file by using a standard ASCII text editor.

---

**CCL8068E Error dumping trace file in memory buffer to file**

**Explanation:** The client was unable to dump the trace memory buffer to the dump file. Ensure the dump file name (in CTG.INI) is valid and the disk is not full.

**System Action:** Service tracing to the memory buffer is not dumped.

**User Response:** Examine any other messages and the client error log to determine the cause of the error.

---

**CCL8069E Userid *userid* exceeds the maximum supported length**

**Explanation:** The length of the userid string supplied exceeds the maximum supported by the CICS Client.

**System Action:** CICSCLI performs no action.

**User Response:** Reissue the command with a valid string.

---

**CCL8070E Password *password* exceeds the maximum supported length**

**Explanation:** The length of the password string supplied exceeds the maximum supported by the CICS Client.

**System Action:** CICSCLI performs no action.

**User Response:** Reissue the command with a valid string.

**CCL8212E The cell administrator userid and password must be supplied**

**Explanation:** CCLAUTH requires the DCE cell administrator userid and password to be supplied in order for it to be able to set the necessary security configuration.

**System Action:** CCLAUTH performs no action.

**User Response:** Reissue the command supplying the correct userid and password.

**CCL8213E DCE is not available on this system**

**Explanation:** DCE security configuration is only required and can on be run on a system that has a correctly installed and configured DCE environment.

**System Action:** CCLAUTH performs no action.

**User Response:** Install and configure DCE if it is required.

**CCL8214E Unable to login as cell administrator - check userid and password**

**Explanation:** CCLAUTH requires the DCE cell administrator userid and password to be supplied in order for it to be able to set the necessary security configuration. The supplied userid or password was incorrect.

**System Action:** CCLAUTH performs no action.

**User Response:** Reissue the command supplying the correct userid and password.

**CCL8215E Unable to connect to security server**

**Explanation:** CCLAUTH is unable to contact the DCE security and registry services. This indicates a problem with DCE or the DCE configuration.

**System Action:** CCLAUTH performs no action.

**User Response:** Correct the DCE problem and reissue the command.

**CCL8216E Unable to create DCE client principal**

**Explanation:** CCLAUTH is unable to create the required client login principal. This indicated a problem with DCE or the DCE configuration.

**System Action:** CCLAUTH performs no action.

**User Response:** Correct the DCE problem and reissue the command.

**CCL8217E Unable to create DCE client security key**

**Explanation:** CCLAUTH is unable to create the required client security key. This indicated a problem with DCE or the DCE configuration.

**System Action:** CCLAUTH performs no action.

**User Response:** Correct the DCE problem and reissue the command.

**CCL8218E DCE reports:** *error*

**Explanation:** The error message *error* was reported by DCE. This message may help explain the reason for a preceeding error encountered by CCLAUTH.

**System Action:** CCLAUTH performs no action.

**User Response:** Correct the DCE problem and reissue the command.

**CCL8219I DCE Security Configuration completed successfully**

**Explanation:** The CICS Client has been correctly configured for use with DCE authenticated RPC security.

**System Action:** CCLAUTH completed successfully.

**User Response:** No further action is required.

**CCL8521E Unknown option** *option*

**Explanation:** An invalid command option was specified when using the CICSBMSC command.

**System Action:** CICSBMSC performs no action.

**User Response:** Reissue the command with the correct option.

**CCL8522E Input file** *filename* **must have .bms or .BMS suffix**

**Explanation:** The CICSBMSC command requires a BMS source file as input.

**System Action:** CICSBMSC performs no action.

**User Response:** Reissue the command with the correct BMS source file name.

**CCL8523E Unable to open BMS input file** *filename*

**Explanation:** The CICSBMSC command could not find, or could not read the supplied BMS source file.

**System Action:** CICSBMSC performs no action.

**User Response:** Check that the BMS source file name is correct, that the file is available and can be read.

**CCL8524E Unable to open output files**

**Explanation:** The CICSBMSC command is unable to create output files.

**System Action:** CICSBMSC processing ends.

**User Response:** Check that there is sufficient disk space available for the CICSBMSC output files.

**CCL8526E Error in BMS source file** *filename* **line** *number*

**Explanation:** The CICSBMSC command has found an error in an input BMS source file.

**System Action:** CICSBMSC processing ends.

**User Response:** Check the BMS macro source at the line indicated.

**CCL8707E Unable to open file** *filename*

**Explanation:** The CICSCONV utility is unable to access the file.

**System Action:** CICSCONV processing ends.

**User Response:** Check that the file specified is accessible.

**CCL8708I Process file** *filename*

**Explanation:** The CICSCONV utility is processing the file.

**System Action:** CICSCONV processing continues.

**User Response:** No further action is required.

**CCL8709I Creating file** *filename*

**Explanation:** The CICSCONV utility is generating the file.

**System Action:** CICSCONV processing continues.

**User Response:** No further action is required.

**CCL8710I Renaming file** *filename* **to file** *filename*

**Explanation:** The CICSCONV utility is changing the name of a file that already exists.

**System Action:** CICSCONV processing continues.

**User Response:** No further action is required.

**CCL8711I Processing complete**

**Explanation:** The CICSCONV utility has finished conversion.

**System Action:** CICSCONV terminates without error.

**User Response:** No further action is required.

**CCL8804E The CICS Client/Gateway is not installed properly**

**Explanation:** The CICS Transaction Gateway or CICS Univseral Client is not installed correctly. Registry keys are created at install time.

**System Action:** The command line application terminates.

**User Response:** Reinstall the CICS Transaction Gateway or CICS Universal Client.

**CCL8805E Unable to read from the registry**

**Explanation:** The program tried to read from the registry but cannot read the appropiate registry key. This is normally caused by security settings in the registry.

**System Action:** The command line application terminates.

**User Response:** Get your system administrator to give read permission to the registry key HKEY_LOCAL_MACHINE\SOFTWARE\IBM\CICS Transaction Gateway or HKEY_LOCAL_MACHINE\SOFTWARE\IBM\CICS Universal Client.

**CCL8806E Unable to write to the registry**

**Explanation:** The program tried to write to the registry but cannot write the appropiate registry key. This is normally caused by security settings in the registry.

**System Action:** The command line application terminates.

**User Response:** Logon as a user who has permissions to write to either HKEY_LOCAL_MACHINE\SOFTWARE\IBM\CICS Transaction Gateway or HKEY_LOCAL_MACHINE\SOFTWARE\IBM\CICS Universal Client. Then repeat the command line option.

**CCL8807E Unable to find a JVM**

**Explanation:** When trying to run a Java application on Windows NT and Windows 98, the application launcher tries to find a JVM, it does this by querying the registry.

**System Action:** The command line application terminates.

**User Response:** Install a JVM, or use the CTGJAVA command to point to the Java Virtual Machine to run.

**CCL8808E Unable to find** *rt.jar* **or** *classes.zip*

**Explanation:** When trying to run a Java application on Windows NT and Windows 98 we need to find the appropriate Java Library to use. It will not be there if you have not installed a JVM correctly.

**System Action:** The command line application terminates.

**User Response:** Reinstall your Java Virtul Machine.

---

**CCL8810I Current JVM set to:** *%s1*

**Explanation:** The CTGJAVA program on Windows NT and Windows 98 can be used to decide which JVM to use. The message shows the currently selected JVM.

**System Action:** This message is for information only.

**User Response:** No further action is required

## User messages

# Chapter 2. Error log messages

This section describes the messages written to the CICS Client error log and trace log.

**CCL1046 Error in function *function* (Error code = *error*)**

**Explanation:** An system or internal client function failed.

**System Action:** The message is written to the error log. The function name and error code are logged.

**User Response:** If the problem persists, contact your service organization.

**CCL1047 Error in the EPI (*function*, *error*, *termid*)**

**Explanation:** An internal EPI error has occurred.

**System Action:** The message is written to the error log. The function name and error code are logged, along with the TermId.

**User Response:** If the problem persists, contact your service organization.

**CCL1200 Unable to open binary trace file *filename***

**Explanation:** The client cannot open the binary trace file for writing.

**System Action:** The client will continue, but tracing will be disabled.

**User Response:** Check that the directory into which the binary trace file is being written is not write-protected.

**CCL1202 Unable to write to binary trace file**

**Explanation:** Although the binary trace file was opened okay, an error occurred whilst writing a trace point to the file.

**System Action:** The client will continue, but tracing will be disabled.

**User Response:** Check that the permissions of the trace file have not changed.

**CCL1203 Out of memory whilst running with trace on**

**Explanation:** There is not enough available memory for tracing to continue.

**System Action:** The client will continue, but tracing will be disabled.

**User Response:** Make more memory available to the client and then try again.

**CCL1204 No memory buffer - unable to perform tracing operations**

**Explanation:** This is a serious internal error, and means that the client is unable to buffer its writes to the trace file.

**System Action:** The client will continue, but tracing will be disabled.

**User Response:** If the problem persists, contact your service organisation.

**CCL1205 Invalid system return code (*Function*, *Error code*)**

**Explanation:** An internal system function has returned an error whilst tracing was turned on.

**System Action:** The client will continue, but tracing will be disabled.

**User Response:** If the problem persists, contact your service organisation.

**CCL1206 Internal error (*Function*, *Error*)**

**Explanation:** An internal function has returned an error whilst tracing was turned on.

**System Action:** The client will continue, but tracing will be disabled.

**User Response:** If the problem persists, contact your service organisation.

**CCL1207 Security failure - the owner of the binary trace file has changed**

**Explanation:** This message will appear in the log file on UNIX if the owner of the trace file changes. For security reasons, you may not change the owner of the trace file whilst trace is running.

**System Action:** The client will continue, but tracing will be disabled.

**User Response:** Restart trace and make sure that the owner of the trace file does not change.

**CCL1210 Trace component *component* is invalid and will be ignored**

**Explanation:** You are trying to trace to a component that is not valid.

**System Action:** Tracing will continue, but the component you have specified will be ignored.

**User Response:** If you do not want this message to

appear in the log, remove the reference to the invalid component.

**CCL2010 Internal transport error (*function*, *error*)**

**Explanation:** An internal client function failed.

**System Action:** The message is written to the error log and the client terminates. The function name and error code are logged.

**User Response:** If the problem persists, contact your service organization.

**CCL2011 Out of memory creating session control structure**

**Explanation:** The client was unable to allocate enough memory to create an internal control structure for a new server conversation.

**System Action:** The message is written to the error log and processing continues. The conversation is not created.

**User Response:** Try altering the values in the client initialization file to provide the client with a larger free memory pool or to reduce the overall memory requirements.

**CCL2012 Unable to find session entry (*session id*, *slot number*)**

**Explanation:** The client was unable to locate an internal control structure for an existing server conversation.

**System Action:** The message is written to the error log and processing continues. The conversation is ended.

**User Response:** If the problem persists, contact your service organization.

**CCL2013 Out of memory creating terminal control structure**

**Explanation:** The client was unable to allocate enough memory to create an internal control structure associated with an emulator or EPI terminal.

**System Action:** The message is written to the error log and processing continues. The control structure is not created.

**User Response:** Try altering the values in the client initialization file to provide the client with a larger free memory pool or to reduce the overall memory requirements.

**CCL2014 Unable to find terminal entry (*session id*, *index*, *name*)**

**Explanation:** The client was unable to locate an internal control structure associated with a terminal or EPI program. This may occur if an ATI request is received for a terminal that no longer exists because either the client or the terminal was shut while the ATI was in transit.

**System Action:** The message is written to the error log and client processing continues but the emulator may be unable to continue.

**User Response:** If the problem persists, contact your service organization.

**CCL2015 Unable to find server entry (*link id*)**

**Explanation:** The client received an event associated with an unconnected server.

**System Action:** The message is written to the error log and processing continues. The event is ignored.

**User Response:** If the problem persists, contact your service organization.

**CCL2016 Invalid communications event identifier (*event id*)**

**Explanation:** The client received an unknown event from one of the communications protocol drivers.

**System Action:** The message is written to the error log and processing continues. The event is ignored.

**User Response:** If the problem persists, contact your service organization.

**CCL2018 Internal OS/2 function error (*function*, *error*)**

**Explanation:** An internal OS/2 function failed.

**System Action:** The message is written to the error log and processing continues. The function name and error code are logged.

**User Response:** If the problem persists, contact your service organization.

**CCL2021 ECI request (SessId=*session*, Slot=*slot*, Type=*type*)**

**Explanation:** The client received an ECI request. The *session* value identifies the application and the *slot* value identifies the unit of work. The message is followed by the application's ECI parameter block.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL2022 ECI COMMAREA Data: Length=*length***

**Explanation:** The client received an ECI request. The *length* value gives the length of the application's COMMAREA. The message is followed by the COMMAREA data.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL2023 Client Response (SessId=*session*, Slot=*slot*, ReqRC=*value1*, AppRC=*value2*)**

**Explanation:** The client sent a response to an application. The *session* and *slot* values identify the application receiving the response. The *value1* and *value2* values contain the response return codes.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL2024 Unknown application (SessId=*session*)**

**Explanation:** The client tried to send a response to an unknown application. Often this occurs if the application has ended before its response was received. The *session* value identifies the application.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL2025 ECI request held (SessId=*session*, Slot=*slot*)**

**Explanation:** The client held an ECI request while the connection to a server was established. The *session* and *slot* values identify the application.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL2026 Server List request (SessId=*session*, Space=*size*)**

**Explanation:** The client received a request to supply a list of server names and descriptions. The *session* value identifies the application issuing the request and the *size* value indicates the amount of space provided to contain the result.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL2027 Server Status request (SessId=*session*, Space=*size*)**

**Explanation:** The client received a request to supply a list of active server names, descriptions, and status. The *session* value identifies the application issuing the request and the *size* value indicates the amount of space provided to contain the result.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL2028 Server Start request (SessId=*session*)**

**Explanation:** The client received a request to start a connection to a server. The *session* value identifies the application issuing the request. The message is followed by the name of the required server.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL2029 Server Stop request (SessId=*session*, Type=*type*)**

**Explanation:** The client received a request to stop a connection to a server. The *session* value identifies the application issuing the request and *type* indicates if a stop or abort was requested. The message is followed by the name of the required server.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL2030 Server Security request (SessId=*session*)**

**Explanation:** The client received a request to set server security logon information. The *session* value identifies the application issuing the request. The message is followed by the new security details.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL2031 Closedown request (SessId=*session*, Type=*type*)**

**Explanation:** The client received a request to terminate. The *session* value identifies the application issuing the request and *type* indicates if a stop or abort was requested.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL2032 Application terminated (SessId=*session*)**

**Explanation:** The client received a notification that an application has terminated. The *session* value identifies the terminating application.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL2034 TR Install request (SessId=*session*, TermId=*index*, Type=*type*)**

**Explanation:** The client received a request to install a terminal from an emulator or EPI application. The *session* value identifies the application, the *index* value gives the EPI terminal index and the *type* value identifies an emulator or EPI request. The message is followed by the requested terminal install details.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL2035 TR request (SessId=*session*, TermId=*index*, Type=*type*)**

**Explanation:** The client received a terminal request from an emulator or EPI application. The *session* value identifies the application, the *index* value gives the EPI terminal index and the *type* value identifies an emulator or EPI request.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL2036 TR TIOA Data: Length=*length***

**Explanation:** The client received a terminal request from an emulator or EPI application. The *length* value gives the length of the terminal or application's TIOA data. The message is followed by the TIOA data.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL2037 TR Install request held (SessId=*session*, TermId=*index*)**

**Explanation:** The client held an emulator or EPI request while the connection to a server was established. The *session* and *index* values identify the application and terminal.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL2038 Service Trace Enable request (SessId=*session*)**

**Explanation:** The client received a request to enable service tracing. The *session* value identifies the application issuing the request.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL2039 **** CICS Client for *system* - Service Trace Begins ******

**Explanation:** The client enabled service tracing. The *system* value identifies the client product.

**System Action:** Tracing begins.

**User Response:** None.

**CCL2040 Service Trace Disable request (SessId=*session*)**

**Explanation:** The client received a request to disable service tracing. The *session* value identifies the application issuing the request.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL2041 ***** CICS Client for *system* - Service Trace Ends *******

**Explanation:** The client disabled service tracing. The *system* value identifies the client product.

**System Action:** Tracing stops.

**User Response:** None.

**CCL2044 Out of memory creating client control structure**

**Explanation:** The client was unable to allocate enough memory to create an internal control structure.

**System Action:** The message is written to the error log and processing continues if possible.

**User Response:** Try altering the values in the client initialization file to provide the client with a larger free memory pool or to reduce the overall memory requirements.

**CCL2045 Request directed to server *server***

**Explanation:** The client routed a request to the named server. The *server* value identifies the server.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

---

**CCL2048 Maximum trace data size set to** *size*

**Explanation:** The client set the maximum trace data size. The *size* value indicates the new maximum trace data size.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

---

**CCL2050 Error and Logon prompt request (SessId=***session***, Type=***type***)**

**Explanation:** The client received a request to enable or suppress error and logon prompts. The *session* value identifies the application issuing the request and the *type* value indicates the request type.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

---

**CCL2051 CICS system transaction** *tran* **failed on server** *server*

**Explanation:** A client system transaction failed to run on the named server. If connecting to a CICS/400 server, a failure of transaction CCIN indicates that the server is not available. The *tran* value indicates the name of the failed transaction and *server* indicates the failing server.

**System Action:** The message is written to the error log and processing continues.

**User Response:** If the problem persists, contact your service organization.

---

**CCL2055 Connection with server established (LinkId=***link***)**

**Explanation:** The client established a connection with a server. The *link* value identifies the server to which the connection was established.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

---

**CCL2056 Connection with server lost (LinkId=***link***)**

**Explanation:** The client lost a connection with a server. The *link* value identifies the server to which the connection was lost.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

---

**CCL2057 Incoming conversation request (LinkId=***link***, ConvId=***conv***)**

**Explanation:** The client received an indication from a server that a new conversation should be established. The *link* value identifies the server requesting the conversation and the *conv* value identifies the new conversation.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

---

**CCL2058 Incoming conversation data (ConvId=***conv***)**

**Explanation:** The client received an indication from a server that data had been received for an active conversation; *conv* identifies the conversation on which data was received.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

---

**CCL2059 Request directed to unknown server** *server*

**Explanation:** The client received a requested routed to an unknown server; *server* identifies the unknown server.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

---

**CCL2060 Request directed to closing server** *server*

**Explanation:** The client received a request routed to a server which was closing down; *server* identifies the closing server.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

---

**CCL2061** *MaxServers* **limit of** *number* **has been reached**

**Explanation:** The client received a request routed to a new server but was already connected to its MaxServers limit; *number* indicates the current MaxServers setting.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

---

**CCL2062** **TR Delete request (SessId=*session*, TermId=*index*, TermName=*name*)**

**Explanation:** The client received a request to delete a terminal from an emulator or EPI application. The *session* value identifies the application, the *index* value gives the EPI terminal index and the *name* gives the terminal name.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL2063** **TR EXIT detected (SessId=*session*, TermId=*index*, TermName=*name*)**

**Explanation:** The client detected the special terminal EXIT transaction code from an emulator or EPI application. The *session* value identifies the application, the *index* value gives the EPI terminal index, and the *name* gives the terminal name.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL2064** **TR issue ATI *tran* (SessId=*session*, TermId=*index*, TermName=*name*)**

**Explanation:** The client requested an emulator or EPI application terminal to start an ATI transaction. The *tran* value gives the name of the ATI, the *session* value identifies the application, the *index* value gives the EPI terminal index, and the *name* gives the terminal name.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL2065** ***MaxRequests* limit of *number* has been reached**

**Explanation:** The client received a request for a new conversation but was already at its MaxRequests limit; *number* indicates the current MaxRequests setting.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL2070** ***MaxBufferSize* limit of *size* exceeded by incoming data length *length***

**Explanation:** The client received data from a server that exceeded the MaxBufferSize limit. The *size* value indicates the current MaxBufferSize setting and the *length* value indicates the length of the incoming data.

**System Action:** The message is written to the error log and processing continues.

**User Response:** Try altering the value of MaxBufferSize in the client initialization file to at least the size of the incoming data.

**CCL2074** ****** CICS Client for *system* - Service Trace to Memory Begins ******

**Explanation:** The client enabled service tracing to the memory buffer. The *system* value identifies the client product.

**System Action:** Tracing to the memory buffer starts.

**User Response:** None.

**CCL2075** ****** CICS Client for *system* - Service Trace to Memory Ends ******

**Explanation:** The client disabled service tracing to the memory buffer. The *system* value identifies the client product.

**System Action:** Tracing to the memory buffer stops.

**User Response:** None.

**CCL2076** ****** CICS Client for *system* - Service Trace to Memory Dumping to File ******

**Explanation:** The service trace in the memory buffer is being dumped to the Dump file. The *system* value identifies the client product. The trace is dumped on request (CICSCLI /Z) or if the client terminates abnormally (on some platforms).

**System Action:** The trace buffer is written to the Dump File.

**User Response:** None.

**CCL2077** ****** CICS Client for *system* - Service Trace to Memory Dumped to File ******

**Explanation:** The service trace in the memory buffer has been dumped to the Dump file. The *system* value identifies the client product. The buffer is cleared after dumping and this trace message will always be the first entry written to the buffer after it has been dumped and cleared.

**System Action:** The trace buffer has written to the Dump File.

**User Response:** None.

**CCL2079** **Failed to put transport message, retry number *number***

**Explanation:** The client has attempted to put a transport message to a window owned by the client application. This has failed, the most likely reason for this is that the message queue associated with the windows client application is full.

**System Action:** The message is written to the trace

file if tracing is enabled. The client allows windows to process messages in the queue and then tries posting the transport message again, a number of times.

**User Response:** Try using SetMessageQueue to increase the size of the message queue.

---

**CCL2080 Extended request continued with server** *server*

**Explanation:** The client continued routing an extended unit of work request to the named server. The *server* value identifies the server.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

---

**CCL2081 CCIN install response: transaction not present**

**Explanation:** The server does not support the CCIN client install transation, this is a normal response from certain older host CICS servers. These servers cannot support client 3270 terminal emulation or user written EPI applications.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

---

**CCL2082 CCIN install response: transaction requires security**

**Explanation:** The server requires a userid and password to be supplied before it will run the CCIN client install transaction. If no userid and password has been made available the client will prompt for one before retrying the connection to the server.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

---

**CCL2083 DCE login context provided**

**Explanation:** The client received an application request with a DCE login context. If authenticated RPC communications is in use the context will be propagated to the server for authorization.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

---

**CCL2084 DCE CDS server** *server* **available**

**Explanation:** The client located the server *server* within the current DCE cell using the DCE cell directory service.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

---

**CCL2085 DCE CDS server** *server* **no longer available**

**Explanation:** The server *server* that was previously located in the current DCE cell is now no longer available.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

---

**CCL2086 DCE cell directory searching enabled**

**Explanation:** The client will make use of the DCE cell directory service to locate possible CICS servers, in addition to servers listed in the client initialization file.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

---

**CCL2092 About to display a popup message**

**Explanation:** A popup message is about to be displayed on the CICS Client System.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

---

**CCL2093 Returned from displaying a Popup Message**

**Explanation:** The popup message has been cleared, processing continues.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

---

**CCL3102 Inbound GDS data error (***gds***,** *length***,** *size***)**

**Explanation:** The client received invalid data from a server. The data does not have a valid CICS data stream header. The *gds* value should be either 0x12F2 or 0x12FF, the *length* value gives the expected data length, and *size* gives the true received length.

**System Action:** The message is written to the error log and processing continues.

**User Response:** If the problem persists, contact your service organization.

---

**CCL3103 Inbound SYNC/ROLL data error (*length*, *data*)**

**Explanation:** The client received invalid data from a server. The data was not part of a valid SYNC or ROLLBACK flow. The *length* values gives the length of the received data and *data* contains the first four bytes.

**System Action:** The message is written to the error log and processing continues.

**User Response:** If the problem persists, contact your service organization.

**CCL3104 Inbound FMH43 header error (*length*)**

**Explanation:** The client received invalid data from a server. The data was not a valid CICS FMH43 header. The *length* values gives the length of the received data.

**System Action:** The message is written to the error log and processing continues.

**User Response:** If the problem persists, contact your service organization.

**CCL3105 Inbound CICS data stream error (*type*, *value1*, *value2*)**

**Explanation:** The client received invalid data from a server. The data was not a valid CICS data stream. The *type* value indicates which part of the data contained errors, *value1* and *value2* further identify the error. This message may be received if the server does not support signon capable terminals.

**System Action:** The message is written to the error log and processing continues.

**User Response:** Use the option to request a non-signon capable terminal. If the problem persists, contact your service organization.

**CCL3106 Out of memory unpacking CICS data stream**

**Explanation:** The client was unable to allocate enough memory to save data from a server. Normally such data is a COMMAREA associated with an emulator terminal.

**System Action:** The message is written to the error log and processing continues.

**User Response:** Try altering the values in the client initialization file, to provide the client with a larger free memory pool or to reduce the overall memory requirements.

**CCL3107 Inbound FMH5 transaction name incorrect (*data*)**

**Explanation:** The client received an invalid transaction attach request from a server. The only valid transaction name is CRSR, used to start ATI requests to a terminal. The *data* value indicates the requested transaction name.

**System Action:** The message is written to the error log and processing continues.

**User Response:** If the problem persists, contact your service organization.

**CCL3108 Inbound FMH5 header error (*length*)**

**Explanation:** The client received invalid data from a server. The data was not a valid CICS FHM5 header. The *length* value gives the length of the received data.

**System Action:** The message is written to the error log and processing continues.

**User Response:** If the problem persists, contact your service organization.

**CCL3113 CCIN install request: AppId=*applid*, Codepage=*codepage***

**Explanation:** The client invoked the CCIN system install transaction on a server. The *applid* and *codepage* values indicate the requested client APPLID and codepage.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL3114 CCIN install response: AppId=*applid*, Codepage=*codepage*, RC=*error***

**Explanation:** The client received the CCIN system install response from a server. The *applid* and *codepage* values indicate the returned client APPLID and codepage. The *error* values are non-zero if any errors occurred.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL3115 CTIN install request: NetName=*name*, Model=*model*, Codepage=*codepage***

**Explanation:** The client invoked the CTIN terminal install transaction on a server. The *name*, *model* and *codepage* values indicate the requested terminal name, model and codepage.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL3116 CTIN install response: NetName=*name*, TermId=*term*, Rc=*error***

**Explanation:** The client received the CTIN terminal install response from a server. The *name* and *term* values indicate the returned terminal NetName and TermName. The *error* values are non-zero if any errors occurred.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL3117 PEM verify request: UserId=*name***

**Explanation:** The client invoked the password expiry management (PEM) transaction on a server to verify a userid and password. The specified user id is indicated.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL3118 PEM change request: UserId=*name***

**Explanation:** The client invoked the password expiry management (PEM) transaction on a server to change the password of the specified user id.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL3119 PEM response: Rc=*error*, CompStatus=*value***

**Explanation:** The client received the password expiry management (PEM) transaction response from a server. The error value is non-zero if an error occured. The *CompStatus* value is set as follows: 1 = Unknown user id 2 = Userid ok, password invalid 3 = Userid ok, password expired 4 = New password unacceptable 5 = Security function failure 6 = Incorrect data format

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL3120 PEM response data error: ReqType=*Request Type*, ErrorValue=*value***

**Explanation:** The client received the password expiry management (PEM) transaction response from a server. The data contained in the corresponding PEM request contained a data formatting error.

**System Action:** The message is written to the error log and processing continues.

**User Response:** If the problem persists, contact your service organization.

**CCL3209 Link to server (*link*) is no longer available**

**Explanation:** The client tried to communicate with a server whose connection is currently unavailable.

**System Action:** The message is written to the error log and processing continues. The server is ignored.

**User Response:** If the problem persists, contact your service organization.

**CCL3221 Unable to find link control structure (*link*)**

**Explanation:** The client was unable to locate an internal control structure associated with a connection to a server.

**System Action:** The message is written to the error log and processing continues. The server is ignored.

**User Response:** If the problem persists, contact your service organization.

**CCL3222 Unable to find conversation control structure (*conv*)**

**Explanation:** The client was unable to locate an internal control structure associated with a server conversation.

**System Action:** The message is written to the error log and processing continues. The conversation is ignored.

**User Response:** If the problem persists, contact your service organization.

**CCL3225 Invalid communications event identifier (*event*)**

**Explanation:** The client received an invalid internal event from a communications protocol driver.

**System Action:** The message is written to the error log and processing continues. The event is ignored.

**User Response:** If the problem persists, contact your service organization.

**CCL3227 Conversation (*conv*) is not in *state* state**

**Explanation:** The client detected a conversation to a server in an incorrect state.

**System Action:** The message is written to the error log and processing continues. The request is ignored.

**User Response:** If the problem persists, contact your service organization.

**CCL3228 Unable to find protocol control structure (*name*, *driver*)**

**Explanation:** The client was unable to locate an internal control structure associated with a communications protocol driver.

**System Action:** The message is written to the error log and processing continues. The driver is ignored.

**User Response:** If the problem persists, contact your service organization.

**CCL3230 Out of memory creating communications control structure**

**Explanation:** The client was unable to allocate enough memory to create an internal control structure associated with communications.

**System Action:** The message is written to the error log and processing continues.

**User Response:** Try altering the values in the client initialization file, to provide the client with a larger free memory pool or to reduce the overall memory requirements.

**CCL3231 Comms Load request (Driver=*name*)**

**Explanation:** The client communications has received a request to load a new protocol driver. The *name* value indicates the required protocol driver DLL name.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL3232 Comms Load completed (Driver=*name*, DrvId=*driver*, RC=*error*)**

**Explanation:** The client completed a request to load a new protocol driver. The *name* value indicates the required protocol driver DLL name, the *driver* value indicates the assigned driver identifier and *error* indicates any error codes.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL3233 Loading protocol driver (DLL=*fullname*)**

**Explanation:** The client wss loading a new communications protocol driver DLL; *fullname* gives the full file system name of the DLL to be loaded.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL3234 Comms Unload completed (Driver=*name*, RC=*error*)**

**Explanation:** The client completed a request to unload a protocol driver. The *name* value indicates the protocol driver DLL name and *error* indicates any error codes.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL3235 Freeing protocol driver (DLL=*fullname*)**

**Explanation:** The client was freeing a communications protocol driver DLL; *fullname* gives the full file system name of the DLL to be freed.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL3236 Comms Open completed (Server=*name*, LinkId=*link*, RC=*error*)**

**Explanation:** The client completed a request to try to establish a new connection with a server. The *name* value indicates the server, the *link* value contains the assigned link identifier, and *error* indicates any error codes.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL3237 Comms Close completed (LinkId=*link*, RC=*error*)**

**Explanation:** The client completed a request to close a connection with a server. The *link* value identifies the server connection to be closed and *error* indicates any error codes.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL3238 Comms Allocate completed (LinkId=*link*, ConvId=*conv*, RC=*error*)**

**Explanation:** The client completed a request to allocate a new conversation with a server.

**System Action:** The message is written to the trace file if tracing is enabled. The *link* value identifies the server, the *conv* value contains the assigned conversation identifier, and *error* indicates any error codes.

**User Response:** None.

**CCL3239 Comms Deallocate completed (ConvId=*conv*, Reason=*reason*, RC=*error*)**

**Explanation:** The client completed a request to deallocate a conversation with a server. The *conv* value identifies the conversation, the *reason* and *error* values indicate any errors.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL3240 Comms Accept completed (ConvId=*conv*, RC=*error*)**

**Explanation:** The client completed a request to accept a new incoming conversation with a server. The *conv* value identifies the conversation and *error* indicates any error codes.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL3241 Comms Send completed (ConvId=*conv*, RC=*error*)**

**Explanation:** The client completed a request to send data over a conversation with a server. The *conv* value identifies the conversation and *error* indicates any error codes.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL3242 Comms Send request: Length=*length* (ConvId=*conv*)**

**Explanation:** The client communications has received a request to send data over a conversation with a server. The *conv* value identifies the conversation and *length* indicates the length of data to be sent.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL3243 Comms Wait completed (ConvId=*conv*, RC=*error*)**

**Explanation:** The client completed a request to wait for a response over a conversation with a server. The *conv* value identifies the conversation and *error* indicates any error codes.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL3244 Comms Receive request (ConvId=*conv*)**

**Explanation:** The client communications has received a request to receive data from a conversation with a server; *conv* identifies the conversation.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL3245 Comms Receive completed: Length=*length* (ConvId=*conv*, Reason=*reason*, RC=*error*)**

**Explanation:** The client completed a request to receive data from a conversation with a server. The conversation remains active. The *conv* value identifies the conversation, the *length* value contains the length of the data received, and *reason* and *error* indicate any error codes. The message is followed by the data received.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL3246 Comms Complete completed (ConvId=*conv*, RC=*error*)**

**Explanation:** The client completed a request to finish with data received from a conversation with a server. The *conv* value identifies the conversation and *error* indicates any error codes.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL3247 Error loading DLL *fullname* (*function*, *error*)**

**Explanation:** The client was unable to load a communications protocol driver DLL.

**System Action:** The message is written to the error log and processing continues. The name of the failing DLL and the loading function and error codes are logged.

**User Response:** Ensure the DLL is available and correctly named. If the problem persists contact your service organization.

**CCL3248 Comms Unload request (Driver=*name*)**

**Explanation:** The client communications has received a request to unload a protocol driver; *name* indicates the protocol driver DLL name.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

---

**CCL3249 Comms Open request (Server=*name*, Driver=*protocol*)**

**Explanation:** The client communications has received a request to try to establish a new connection with a server. The *name* value indicates the server and *protocol* indicates the required communications protocol driver.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

---

**CCL3250 Comms Close request (LinkId=*link*)**

**Explanation:** The client communications has received a request to close a connection with a server; *link* identifies the server connection.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

---

**CCL3251 Comms Allocate request (LinkId=*link*, Tran=*tran*)**

**Explanation:** The client communications has received a request to allocate a new conversation with a server. The *link* value identifies the server and the *tran* value gives the name of the transaction to be attached on the server.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

---

**CCL3252 Comms Deallocate request (ConvId=*conv*)**

**Explanation:** The client communications has received a request to deallocate a conversation with a server; *conv* identifies the conversation.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

---

**CCL3253 Comms Accept request (ConvId=*conv*)**

**Explanation:** The client communications has received a request to accept a new incoming conversation with a server; *conv* identifies the conversation.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

---

**CCL3254 Comms Wait request (ConvId=*conv*)**

**Explanation:** The client communications has received a request to wait for a response over a conversation with a server; *conv* identifies the conversation.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

---

**CCL3255 Comms Complete request (ConvId=*conv*)**

**Explanation:** The client communications has received a request to finish with data received from a conversation with a server; *conv* identifies the conversation.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

---

**CCL3256 Comms Receive completed (last): Length=*length* (ConvId=*conv*, Reason=*reason*, RC=*error*)**

**Explanation:** The client completed a request to receive data from a conversation with a server. The conversation has been ended. The *conv* value identifies the conversation, the *length* value contains the length of the data received, and *reason* and *error* indicate any error codes. The message is followed by the data received.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

---

**CCL3299 Internal communications error (*function*, *error*)**

**Explanation:** An internal client communications function failed.

**System Action:** The message is written to the error log and the client terminates. The function name and error code are logged.

**User Response:** If the problem persists, contact your service organization.

---

**CCL3300 Before *EPI Calltype task information***

**Explanation:** The client EPI application is about to issue EPI call *EPI Calltype*

**System Action:** The message is written to the trace file if tracing is enabled and ApiTrace is specified in the client initialization file.

**User Response:** None.

**CCL3301** **After** *EPI Calltype* **Event:** *Event*, **rc=***rc task information*

**Explanation:** The client EPI application has issued EPI call *EPI Calltype* which ended with return code *rc*. Event *Event* also occurred.

**System Action:** The message is written to the trace file if tracing is enabled and ApiTrace is specified in the client initialization file.

**User Response:** None.

**CCL3302** *EPI event information*,**Idx=***Idx task information*

**Explanation:** This trace entry is issued for a number of reasons during EPI processing. The text of the message will describe the particular instance. Idx is the EPI terminal index.

**System Action:** The message is written to the trace file if tracing is enabled and ApiTrace is specified in the client initialization file.

**User Response:** None.

**CCL3303** **Before** *EPI Calltype* **continued** *extended call information task information*

**Explanation:** The client EPI application is about to issue EPI call *EPI Calltype*. This trace entry contains more information relating to the EPI call detailed in a previous CCL3300 trace entry for this task. The data that follows this trace entry is : for CICS_EpiStartTran : Data for CICS_EpiReply : Data As provided by the application.

**System Action:** The message is written to the trace file if tracing is enabled and ApiTrace and ApiTraceLevel = 2 is specified in the client initialization file.

**User Response:** None.

**CCL3304** **After** *EPI Calltype* **continued** *extended call information task information*

**Explanation:** The client EPI application has issued EPI call *EPI Calltype*. This trace entry contains more information relating to the EPI call detailed in a previous CCL3301 trace entry for this task. The data that follows this trace entry is : for CICS_EpiListSystems : List for CICS_EpiAddterminal : Details structure for CICS_EpiEvent : Event As returned to the application.

**System Action:** The message is written to the trace file if tracing is enabled and ApiTrace and ApiTraceLevel = 2 is specified in the client initialization file.

**User Response:** None.

**CCL3310** **Before** *ECI Calltype*, **UOW Token=***UOW token task information*

**Explanation:** The client ECI application is about to issue ECI call *ECI Calltype*

**System Action:** The message is written to the trace file if tracing is enabled and ApiTrace is specified in the client initialization file.

**User Response:** None.

**CCL3311** **After** *ECI Calltype*, **rc=***rc*, **UOW Token=***UOW token task information*

**Explanation:** The client ECI application has issued ECI call *ECI Calltype*, which completed with return code *rc*.

**System Action:** The message is written to the trace file if tracing is enabled and ApiTrace is specified in the client initialization file.

**User Response:** None.

**CCL3312** *Notification type* **for call** *ECI Calltype task information*

**Explanation:** The client ECI application has been notified using *notification type* for ECI call *ECI CallType*.

**System Action:** The message is written to the trace file if tracing is enabled and ApiTrace is specified in the client initialization file.

**User Response:** None.

**CCL3313** **Before** *function name task information*

**Explanation:** The client application is about to issue the call *function name*

**System Action:** The message is written to the trace file if tracing is enabled and ApiTrace is specified in the client initialization file.

**User Response:** None.

**CCL3314** **After** *function name*, **rc=***rc task information*

**Explanation:** The client application has issued the call *function name*, which completed with return code *rc*.

**System Action:** The message is written to the trace file if tracing is enabled and ApiTrace is specified in the client initialization file.

**User Response:** None.

**CCL3400 Unable to find an LDAP server (rc=rc, Reason=*reason*)**

**Explanation:** Trying to find the LDAP server using DNS.

**System Action:** The message is written to the log file if the error occurs. The system will continue by using the existing configuration file.

**User Response:** To use LDAP contact your system administrator to set up their DNS correctly. Otherwise disable LDAP by using the environment variable or the */z* option on the command line.

**CCL3401 Unable to retrieve user credentials**

**Explanation:** Trying to get the users credentials for use with LDAP.

**System Action:** The message is written to the log file if the error occurs. The system will continue by using the existing configuration file.

**User Response:** To rectify this problem it is recommended you reinstall the product.

**CCL3402 Unable to retrieve LDAP password (rc=rc, Reason=*reason*)**

**Explanation:** To retrieve encrypted LDAP password from the registry.

**System Action:** The message is written to the log file if the error occurs. The system will continue by using the existing configuration file.

**User Response:** The user must have a user ID set up to talk to the LDAP server. Check that this has been done.

**CCL3403 Unknown LDAP error (rc=rc, Reason=*reason*)**

**Explanation:** An unknown error has occurred while using LDAP.

**System Action:** The message is written to the log file if the error occurs. The system will continue by using the existing configuration file. Look at the reason for more guidance.

**User Response:** None.

**CCL3404 Binding error to LDAP server (rc=rc, Reason=*reason*)**

**Explanation:** An error occured while trying to bind to the LDAP server.

**System Action:** The message is written to the log file if the error occurs. The system will continue by using the existing configuration file. Look at the reason for more guidance.

**User Response:** None.

**CCL3405 Unbinding error to LDAP server (rc=rc, Reason=*reason*)**

**Explanation:** An error occurred while trying to bind to the LDAP server.

**System Action:** The message is written to the log file if the error occurs. The system will continue by using the new configuration file. Look at the reason for more guidance.

**User Response:** None.

**CCL3406 Trying to read LDAP entry error (rc=rc, Reason=*reason*, entry=*baseDN*)**

**Explanation:** The Client tried to read an LDAP entry, but was unable too.

**System Action:** This message is written to the trace file, if tracing is enabled. Look at the reason for more guidance

**User Response:** None.

**CCL3407 Entry does not exist (rc=rc, Reason=*reason*, entry=*baseDN*)**

**Explanation:** The Client tried to test for existence of an LDAP entry.

**System Action:** This message is written to the trace file, if tracing is enabled. Look at the reason for more guidance

**User Response:** None.

**CCL3408 Unable to find any configuration entry in LDAP server**

**Explanation:** An error occurred while trying to find a configuration entry on the LDAP server, but was unable to find any.

**System Action:** The message is written to the log file if the error occurs. The system will continue by using the existing configuration file.

**User Response:** None.

**CCL3409 Unable to write to file**

**Explanation:** An error occurred while trying to write the new configuration file.

**System Action:** The message is written to the log file if the error occurs. The system will continue by using the existing configuration file.

**User Response:** Check the security settings on the ctg.ini file.

**CCL4201 NetBIOS out of memory for control blocks**

**Explanation:** The NetBIOS protocol driver could not allocate enough memory to create an internal control structure.

**System Action:** The message is written to the error log and processing continues.

**User Response:** Try altering the values in the client initialization file, to provide the client with a larger free memory pool or to reduce the overall memory requirements.

**CCL4202 NetBIOS out of memory for command blocks**

**Explanation:** The NetBIOS protocol driver could not allocate enough memory to create the NCB structures.

**System Action:** The message is written to the error log and processing continues.

**User Response:** Try altering the values in the client initialization file, to provide the client with a larger free memory pool or to reduce the overall memory requirements.

**CCL4203 NetBIOS out of memory for data buffers**

**Explanation:** The NetBIOS protocol driver could not allocate enough memory to create the internal data buffers.

**System Action:** The message is written to the error log and processing continues.

**User Response:** Try altering the values in the client initialization file, to provide the client with a larger free memory pool or to reduce the overall memory requirements.

**CCL4204 NetBIOS *function* call failed: RC=*error***

**Explanation:** The NetBIOS protocol driver failed when issuing a NetBIOS command.

**System Action:** The message is written to the error log and processing continues. The failing NetBIOS command and error code are logged.

**User Response:** If the problem persists, contact your service organization.

**CCL4205 NetBIOS resources *type*: Sessions=*sessions*, Commands=*commands*, Names=*names***

**Explanation:** The NetBIOS protocol driver requested or obtained the listed number of NetBIOS resources. The *type* value indicates if the values logged represent the number of resources required or the number of resources actually obtained.

**System Action:** The message is written to the error log and processing continues.

**User Response:** Alter the system NetBIOS configuration to ensure enough of the requested resources are available, or alter values in the client initialization file to reduce resource requirements. If the problem persists contact your service organization.

**CCL4206 NetBIOS LAN adapter number *number* is invalid**

**Explanation:** The NetBIOS protocol driver detected an invalid network LAN adapter number. Adapter numbers should be in the range 0 to 3; *number* indicates the incorrect value.

**System Action:** The message is written to the error log and processing continues.

**User Response:** Alter the Adapter value in the client initialization file to ensure it is in the range 0 to 3.

**CCL4207 NetBIOS received unknown data type (*type*)**

**Explanation:** The NetBIOS protocol driver has received an invalid header type. *type* indicates the incorrect header type.

**System Action:** The message is written to the error log and processing continues.

**User Response:** If the problem persists, contact your service organization.

**CCL4208 NetBIOS received attach for invalid transaction (*tran*)**

**Explanation:** The NetBIOS protocol driver has received a request to attach an invalid transaction. The only transaction the client can run is CRSR to schedule a terminal ATI; *tran* indicates the incorrect transaction name.

**System Action:** The message is written to the error log and processing continues.

**User Response:** If the problem persists, contact your service organization.

**CCL4209 NetBIOS received data for invalid conversation (*conv*)**

**Explanation:** The NetBIOS protocol driver has received data for an invalid conversation; *conv* indicates the incorrect conversation identifier.

**System Action:** The message is written to the error log and processing continues.

**User Response:** If the problem persists, contact your service organization.

---

**CCL4210 NetBIOS send data for conversation** *adapter*/*session*: **Length=***length*

**Explanation:** The NetBIOS protocol driver was about to send data to a server. The *adapter* and *session* values identify the server, the *length* value contains the length of the data to be sent. This message is followed by the data to be sent, which consists of the client data and a NetBIOS protocol header.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

---

**CCL4211 NetBIOS receive data for conversation** *adapter*/*session*: **Length=***length*

**Explanation:** The NetBIOS protocol driver has received data from a server. The *adapter* and *session* values identify the server, the *length* value contains the length of the data received. This message is followed by the data received, which consists of the data returned to the client and a NetBIOS protocol header.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

---

**CCL4212 NetBIOS receive pdu for conversation** *adapter*/*session*: **Type=***type*

**Explanation:** The NetBIOS protocol driver has received data from a server. The *adapter* and *session* values identify the server, the *type* value identifies the type of data received.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

---

**CCL4213 NetBIOS support may be unavailable on this system**

**Explanation:** The NetBIOS protocol driver could not issue NetBIOS calls on this system. NetBIOS may not be installed or may not be active on the system.

**System Action:** The message is written to the error log and processing continues.

**User Response:** Alter the system configuration to try to activate or install NetBIOS. If the problem persists, contact your service personnel.

---

**CCL4214 NetBIOS close conversation** *adapter*/*session*

**Explanation:** The NetBIOS protocol driver has received a request to close a conversation with a server. The *adapter* and *session* values identify the server.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

---

**CCL4215 NetBIOS name added for adapter** *adapter*

**Explanation:** The NetBIOS protocol driver added a NetBIOS name to a LAN adapter. The *adapter* value indicates the LAN adapter to which the name was added. This message is followed by the actual NetBIOS name used; this name varies for each client session.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

---

**CCL4216 NetBIOS connection established for conversation** *adapter*/*session*

**Explanation:** The NetBIOS protocol driver has established a NetBIOS connection to a server. The *adapter* and *session* values identify the server.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

---

**CCL4217 NetBIOS command failed for conversation** *adapter*/*session*: **Cmd=***command*, **RC=***error*

**Explanation:** The NetBIOS protocol driver has detected a failure in an outstanding NetBIOS command. This usually means the server is no longer available. The *adapter* and *session* values identify the server, the *command* and *error* values identify the failed command.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

---

**CCL4401 TCP/IP not available**

**Explanation:** The TCP/IP protocol driver cannot issue TCP/IP calls on this system. TCP/IP may not be installed or may not be active on the system.

**System Action:** The message is written to the error log and processing continues.

**User Response:** Alter the system configuration to try to activate or install TCP/IP. If the problem persists,

contact your service personnel.

**CCL4402 TCP/IP out of memory for data buffers**

**Explanation:** The TCP/IP protocol driver could not allocate enough memory to create the internal data buffers.

**System Action:** The message is written to the error log and processing continues.

**User Response:** Try altering the values in the client initialization file, to provide the client with a larger free memory pool or to reduce the overall memory requirements.

**CCL4403 TCP/IP (to *server*) no resources available: RC=*error***

**Explanation:** The TCP/IP protocol driver could not create a TCP/IP socket. The *server* value identifies the server and the *error* value identifies the TCP/IP error code.

**System Action:** The message is written to the error log and processing continues.

**User Response:** If the problem persists, contact your service organization.

**CCL4404 TCP/IP (to *server*) unable to resolve name: RC=*error***

**Explanation:** The TCP/IP protocol driver could not resolve a server host IP name. The *server* value identifies the server and the *error* value gives the TCP/IP error code.

**System Action:** The message is written to the error log and processing continues.

**User Response:** Ensure the server IP name or address is correct and can be resolved by the local name server. If the problem persists contact your service organization.

**CCL4405 TCP/IP (to *server*) out of memory for control blocks**

**Explanation:** The TCP/IP protocol driver could not allocate enough memory to create the internal control structures; *server* identifies the server.

**System Action:** The message is written to the error log and processing continues.

**User Response:** Try altering the values in the client initialization file, to provide the client with a larger free memory pool or to reduce the overall memory requirements.

**CCL4406 TCP/IP (to *server*) send failed: RC=*error***

**Explanation:** The TCP/IP protocol driver could not send data to a server. The *server* value identifies the server and the *error* value gives the TCP/IP error code.

**System Action:** The message is written to the error log and processing continues.

**User Response:** If the problem persists, contact your service organization.

**CCL4407 TCP/IP (to *server*) connect failed: RC=*error***

**Explanation:** The TCP/IP protocol driver could not connect to a server. The *server* value identifies the server and the *error* value gives the TCP/IP error code.

**System Action:** The message is written to the error log and processing continues.

**User Response:** If the problem persists, contact your service organization.

**CCL4408 TCP/IP (to *server*) recv failed: RC=*error***

**Explanation:** The TCP/IP protocol driver could not receive data from a server. The *server* value identifies the server and the *error* value gives the TCP/IP error code.

**System Action:** The message is written to the error log and processing continues.

**User Response:** If the problem persists, contact your service organization.

**CCL4409 TCP/IP (to *server*) error starting listener**

**Explanation:** The TCP/IP protocol driver could not start an internal listener function; *server* identifies the server.

**System Action:** The message is written to the error log and processing continues.

**User Response:** If the problem persists, contact your service organization.

**CCL4410 TCP/IP requires WINSOCK v*required* (v*found* installed)**

**Explanation:** The TCP/IP protocol driver for Windows requires a WINSOCK DLL of at least version *required*, however version *found* was found to be installed. The *required* and *found* values indicate the version of WINSOCK required and found respectively.

**System Action:** The message is written to the error log and processing continues.

**User Response:** Ensure your TCP/IP product provides a suitable level of the WINSOCK DLL and the system is

correctly configured. If the problem persists contact your service organization.

---

**CCL4411 TCP/IP (to *server*) send data: Length=*length***

**Explanation:** The TCP/IP protocol driver was about to send data to a server. The *server* value identifies the server and the *length* value contains the length of the data to be sent. This message is followed by the data to be sent, which consists of the client data and a TCP/IP protocol header. The total data flow may be sent in several pieces.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

---

**CCL4412 TCP/IP (to *server*) receive data: Length=*length***

**Explanation:** The TCP/IP protocol driver received data from a server. The *server* value identifies the server and the *length* value contains the length of the data received. This message is followed by the data received, which consists of the data returned to the client and a TCP/IP protocol header. The total data flow may be received in several pieces.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

---

**CCL4413 TCP/IP (to *server*) address=*ipaddr*, port=*port*, socket=*socket***

**Explanation:** The TCP/IP protocol driver has established a connection to a server. The *server* value identifies the server. The *ipaddr*, *port*, and *socket* values give the server's IP address, the server's port number, and the socket used, respectively.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

---

**CCL4414 TCP/IP (to *server*) link failed: RC=*error***

**Explanation:** The TCP/IP link failed. The *server* value identifies the server and the *error* value gives the TCP/IP error code.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** If the problem persists, contact your service organization.

---

**CCL4415 TCP/IP (to *server*) abend received: RC=*sense code***

**Explanation:** The transaction failed. The *server* value identifies the server and the *sense code* value gives the FMH7 sense data.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** If the problem persists, contact your service organization.

---

**CCL4416 TCP/IP (to *server*) Waiting for Connect Timer**

**Explanation:** The ConnectTimeOut value in the Server Section of the initialisation file has been set to a value other than 0 (the default), and the connection timer is running.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

---

**CCL4417 TCP/IP (to *server*) TimeOut on Connect: Rc=*error***

**Explanation:** The timer for ConnectTimeOut has expired and the attempt to connect to the Server is being terminated.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

---

**CCL4499 TCP/IP (to *server*) data to be received overflows buffer**

**Explanation:** The TCP/IP protocol driver received more data than its buffer space could hold. The *server* value identifies the server, the data is discarded.

**System Action:** The message is written to the error log and processing continues.

**User Response:** If the problem persists, contact your service organization.

---

**CCL4601 Memory allocation of display buffer failed**

**Explanation:** There was insufficient free memory for the client SNA protocol driver to retrieve LU 6.2 information from the APPC provider.

**System Action:** The SNA protocol driver is not initialized.

**User Response:** Free up some memory and retry.

---

**CCL4602 Memory allocation of receive buffer pool failed**

**Explanation:** There was insufficient free memory for the client SNA protocol driver to receive data from any servers.

**System Action:** The SNA protocol driver is not initialized.

**User Response:** Free up some memory and retry.

**CCL4603 Memory allocation of send buffer failed**

**Explanation:** There was insufficient free memory for the client SNA protocol driver to send data to any servers.

**System Action:** The SNA protocol driver is not initialized.

**User Response:** Free up some memory and retry.

**CCL4605 Begin thread to handle inbound ATI requests failed**

**Explanation:** Creation of thread, to handle inbound allocate requests from ATI to client terminals, failed.

**System Action:** The SNA protocol driver is not initialized.

**User Response:** Contact your system administrator.

**CCL4607 Netname** *partner LU name* **has invalid length**

**Explanation:** The specified partner LU name in the *NetName* parameter in your client initialization file has an invalid length for this SNA protocol driver. The maximum length is 17 characters (********.********) for a fully qualified name and 8 characters otherwise. The minimum length is 1.

**System Action:** The connection to the partner LU is not started.

**User Response:** See the CICS Client Administration book to determine whether or not this SNA protocol driver requires a fully qualified name. Correct the *NetName* parameter in your client initialization file and retry.

**CCL4608 Memory allocation of link data block failed for** *server* **connection**

**Explanation:** There was insufficient free memory for the client SNA protocol driver to open a link to the specified server.

**System Action:** The client continues.

**User Response:** Free up some memory and retry.

**CCL4609 Retrieval of LU 6.2 information for** *server* **connection failed, APPC return code** *primary RC*, *secondary RC*

**Explanation:** The client SNA protocol driver was unable to open a link to the specified server because retrieval of LU6.2 information from the APPC provider failed. The APPC DISPLAY verb failed with the specified return code.

**System Action:** The client continues.

**User Response:** Contact your system administrator.

**CCL4610 LU 6.2 information for** *server* **connection not all returned on retrieval**

**Explanation:** The client SNA protocol driver was unable to open a link to the specified server because not all of the LU6.2 information could be obtained from the APPC provider.

**System Action:** The client continues.

**User Response:** Contact your system administrator.

**CCL4611 Local LU** *LU name* **is not defined**

**Explanation:** The client SNA protocol driver was unable to open any link using this LU because it is not defined to IBM Communications Manager/2.

**System Action:** The client continues.

**User Response:** Define the LU to IBM Communications Manager/2 or correct the *LocalLUName* parameter in your client initialization file and retry.

**CCL4612 Partner LU** *partner LU name* **is not defined**

**Explanation:** The client SNA protocol driver was unable to open any link using this partner LU because it is not defined to IBM Communications Manager/2.

**System Action:** The client continues.

**User Response:** Define the partner LU to IBM Communications Manager/2 or correct the *NetName* parameter in your client initialization file and retry.

**CCL4613 Communication Subsystem not loaded, APPC return code** *primary RC*

**Explanation:** APPC could not execute the verb because Communications Manager had not started APPC. Either Communications Manager has not been started or it has not been configured correctly for the application.

**System Action:** The client continues.

**User Response:** Start Communications Manager if it is not active. If the necessary APPC profiles are not

configured, configure them and restart Communications Manager.

**CCL4614** **Open connection to** *server* **failed with APPC return code** *primary RC*, *secondary RC*

**Explanation:** The APPC verb TP_STARTED failed with the specified return code. The connection to the specified server has not been started.

**System Action:** The client continues.

**User Response:** See your APPC Programming Reference information for further information, or contact your system administrator.

**CCL4615** **Close connection to** *server* **failed with APPC return code** *primary RC*, *secondary RC*

**Explanation:** The APPC verb TP_ENDED failed with the specified return code.

**System Action:** The client connection to the specified server is closed.

**User Response:** See your APPC Programming Reference information for further information, or contact your system administrator.

**CCL4616** **Memory allocation of conversation data block failed for** *server* **connection**

**Explanation:** There was insufficient free memory for the client SNA protocol driver to start a conversation with the specified server.

**System Action:** The client continues.

**User Response:** Free up some memory and retry.

**CCL4617** **No contention-winner sessions free**

**Explanation:** APPC could not allocate a conversation because no free sessions were available.

**System Action:** The client continues.

**User Response:** Wait until one or more existing conversations have finished and retry.

**CCL4618** **SNASVCMG or CPSVCMG is not a valid mode name, APPC return code** *primary RC*, *secondary RC*

**Explanation:** An attempt was made to allocate a client conversation in SNASVCMG or CPSVCMG sessions mode. These are not valid modes.

**System Action:** The client continues.

**User Response:** Change the *Modename* parameter in your client initialization file and retry.

**CCL4619** **Mode name** *mode name* **is not configured, APPC return code** *primary RC*, *secondary RC*

**Explanation:** The specified mode has not been defined.

**System Action:** The client continues.

**User Response:** Check whether the mode has been configured, and the spelling of the *ModeName* parameter in your client initialization file.

**CCL4620** **Allocate session to** *server* **failed with APPC return code** *primary RC*, *secondary RC*

**Explanation:** The APPC verb ALLOCATE failed with the specified return code. A conversation with the specified server has not been started.

**System Action:** The client continues.

**User Response:** See your APPC Programming Reference information for further information, or contact your system administrator.

**CCL4621** **Deallocate session to** *server* **failed with APPC return code** *primary RC*, *secondary RC*

**Explanation:** The APPC verb DEALLOCATE failed with the specified return code.

**System Action:** The client continues.

**User Response:** See your APPC Programming Reference information for further information, or contact your system administrator.

**CCL4622** **Begin thread to receive data for ATI conversation with** *server* **failed**

**Explanation:** A client thread could not be created to process the queue holding inbound attach requests; _beginthread failed.

**System Action:** The SNA protocol driver is not initialized.

**User Response:** Contact your system administrator.

**CCL4623** **Begin thread to receive data for conversation with** *server* **failed**

**Explanation:** A client thread could not be created to receive data from the server for the current conversation; _beginthread failed.

**System Action:** The conversation is deallocated.

**User Response:** Contact your system administrator.

**CCL4624 Inbound ATI initialisation failed with return code** *return code*

**Explanation:** An attempt to allocate resources for inbound allocate requests from ATI to client terminals has failed.

**System Action:** The client continues but all inbound request processing is terminated.

**User Response:** Check SNA configuration for ATI Side Information Profile entry.

**CCL4625 Error processing inbound ATI request, all ATI processing terminated**

**Explanation:** An error occurred during an inbound ATI request.

**System Action:** The client continues but all inbound request processing is terminated.

**User Response:** Contact your system administrator.

**CCL4626 Attach Manager stopped, APPC return code** *primary RC*, *secondary RC*

**Explanation:** The client received an inbound ATI attach request but the receive allocate failed because the attach manager was stopped.

**System Action:** The inbound allocate request is rejected.

**User Response:** Start the attach manager.

**CCL4629 Receive_allocate for ATI request failed with APPC return code** *primary RC*, *secondary RC*

**Explanation:** The APPC verb RECEIVE_ALLOCATE failed with the specified return code.

**System Action:** The inbound allocate request is rejected.

**User Response:** See your APPC Programming Reference information for further information, or contact your system administrator.

**CCL4630 Send data to** *server* **failed with APPC return code** *primary RC*, *secondary RC*

**Explanation:** The APPC verb SEND_DATA failed with the specified return code.

**System Action:** The client sends a deallocate abend to the server to terminate the conversation.

**User Response:** See your APPC Programming Reference information for further information, or contact your system administrator.

**CCL4631 Receive data from** *server* **failed with APPC return code** *primary RC*, *secondary RC*

**Explanation:** The APPC verb RECEIVE_AND_POST failed with the specified return code.

**System Action:** The client sends a deallocate abend to the server to terminate the conversation.

**User Response:** See your APPC Programming Reference information for further information, or contact your system administrator.

**CCL4632 Error receiving data from** *server*, **APPC what_received return parameter is** *APPC return code*

**Explanation:** An APPC RECEIVE returned an unexpected data_received or conversation status_received indicator.

**System Action:** The client sends a deallocate abend to the server to terminate the conversation.

**User Response:** See your APPC Programming Reference information for further information, or contact your system administrator.

**CCL4633 Memory allocation of receive buffer failed**

**Explanation:** There was insufficient free memory for the client SNA protocol driver to receive data on the current conversation.

**System Action:** The client sends a deallocate abend to the server to terminate the conversation.

**User Response:** Free up some memory and retry the transaction.

**CCL4634 Create semaphore for ReceiveAndPost failed, return code** *operating system return code*

**Explanation:** An attempt to set post semaphore for receiving conversation data failed.

**System Action:** The client sends a deallocate abend to the server to terminate the conversation.

**User Response:** Contact your system administrator.

**CCL4635 Allocate session to** *server* **failed, TP name** *TP name* **unknown, APPC return code** *primary RC*, *secondary RC*

**Explanation:** The server rejected the incoming attach because the client or client application specified a TP name that the server did not recognize.

**System Action:** The client continues.

**User Response:** Ensure that the specified TP is installed on the server.

---

**CCL4636** **Allocate session to** *server* **failed with APPC return code** *primary RC*, *secondary RC*. **Retry**

**Explanation:** APPC could not allocate a conversation but retrying without intervention may succeed. APPC was probably unable to activate a link or a session.

**System Action:** The client continues.

**User Response:** Retry the operation. If retrying does not succeed, see your APPC Programming Reference information for further information or contact your system administrator.

---

**CCL4637** **Allocate session to** *server* **failed with APPC return code** *primary RC*, *secondary RC*; **no retry**

**Explanation:** APPC could not allocate a conversation and intervention is required to correct the problem.

**System Action:** The client continues.

**User Response:** See your APPC Programming Reference information for further information or contact your system administrator.

---

**CCL4638** **Prepare to receive data from** *server* **failed with APPC return code** *primary RC*, *secondary RC*

**Explanation:** The APPC verb PREPARE_TO_RECEIVE failed with the specified return code.

**System Action:** The client sends a deallocate abend to the server to terminate the conversation.

**User Response:** See your APPC Programming Reference information for further information, or contact your system administrator.

---

**CCL4639** **Receive data from** *server* **failed with APPC return code** *primary RC*, *secondary RC*

**Explanation:** The APPC verb RECEIVE_IMMEDIATE failed with the specified return code.

**System Action:** The client sends a deallocate abend to the server to terminate the conversation.

**User Response:** See your APPC Programming Reference information for further information, or contact your system administrator.

---

**CCL4640** **Error receiving data from** *server*, **APPC what_received return parameter is** *APPC return code*

**Explanation:** An APPC RECEIVE returned an unexpected data_received or conversation status_received indicator.

**System Action:** The client sends a deallocate abend to the server to terminate the conversation.

**User Response:** See your APPC Programming Reference information for further information, or contact your system administrator.

---

**CCL4641** **Send data and deallocate to** *server* **failed with APPC return code** *primary RC*, *secondary RC*

**Explanation:** The APPC verb SEND_DATA with DEALLOCATE type failed with the specified return code.

**System Action:** The client sends a deallocate abend to the server to terminate the conversation.

**User Response:** See your APPC Programming Reference information for further information, or contact your system administrator.

---

**CCL4642** **Invalid number of parameters defined for TP program** *EXE name*

**Explanation:** The TP program was invoked because of an inbound allocate request but the IBM Communications Manager definition for the TP program has an invalid number of parameters defined.

**System Action:** The inbound allocate request is rejected.

**User Response:** See the CICS Client Administration book, correct the TP definition, and retry.

---

**CCL4643** **Invalid TP name parameter defined for TP program** *EXE name*

**Explanation:** The TP program was invoked because of an inbound allocate request but the IBM Communications Manager definition for the TP program has an invalid TP name parameter defined.

**System Action:** The inbound allocate request is rejected.

**User Response:** See the CICS Client Administration book, correct the TP definition, and retry.

---

**CCL4646** **APPC send data: Length=***length*

**Explanation:** This is trace data sent to the server by the client SNA protocol driver.

**System Action:** None.

**User Response:** None.

**CCL4647 APPC receive data: Length=***length*

**Explanation:** This is trace data received from the server by the client SNA protocol driver.

**System Action:** None.

**User Response:** None.

**CCL4648 Receive data from** *server* **unsuccessful after partial data received, APPC return code** *primary RC*, *secondary RC*

**Explanation:** The APPC verb RECEIVE_IMMEDIATE failed with the specified return code after some data had already been received.

**System Action:** The client sends a deallocate abend to the server to terminate the conversation.

**User Response:** Try altering the *SnaRetryCount* parameter in your client initialization file and retry. Refer to the CICS Client Administration book.

**CCL4649 Post on receipt of data from** *server* **failed with APPC return code** *primary RC*, *secondary RC*

**Explanation:** The APPC verb POST_ON_RECEIPT failed with the specified return code.

**System Action:** The client sends a deallocate abend to the server to terminate the conversation.

**User Response:** See your APPC Programming Reference information for further information, or contact your system administrator.

**CCL4650 Partner LU name** *netname* **is not fully qualified**

**Explanation:** The partner LU name specified in the *NetName* parameter in your client initialization file was not qualified by the SNA network name (NETID). The format is NETID.LU_NAME.

**System Action:** The connection to the partner LU is not started.

**User Response:** If using real LU names, correct the *NetName* parameter and retry. If using alias LU names, set the *LUAliasNames* parameter to *Y* and retry.

**CCL4651 Allocate session to** *server* **unsuccessful after** *number* **retry attempts, APPC return code** *primary RC*, *secondary_RC*

**Explanation:** The attempt to allocate a session to the specified server was unsuccessful after the number of retries specified in your client initialization file.

**System Action:** The client continues.

**User Response:** Contact your system administrator.

**CCL4652 Conversation with** *server* **failed, APPC return code** *primary RC*, **retry**

**Explanation:** A temporary failure prematurely ended the conversation.

**System Action:** The client continues.

**User Response:** Retry the operation. If retrying does not succeed, see your APPC Programming Reference information for further information, or contact your system administrator.

**CCL4653 Conversation with** *server* **failed, APPC return code** *primary RC*, **no retry**

**Explanation:** A permanent failure prematurely ended the conversation. The link to the server has probably failed.

**System Action:** The client continues.

**User Response:** See your APPC Programming Reference information for further information or contact your system administrator.

**CCL4654 CNOS for mode** *mode* **sessions to** *server* **failed, APPC return code** *primary RC*, *secondary RC*

**Explanation:** The APPC verb CNOS failed with the specified return code. The connection to the specified server has not been started.

**System Action:** The client continues.

**User Response:** See your APPC Programming Reference information for further information, or contact your system administrator.

**CCL4655 Partner LU** *server* **maximum session limit is zero, APPC return code** *primary RC*, *secondary RC*

**Explanation:** The APPC verb CNOS failed because the local maximum session limit of the partner LU is 0. The connection to the specified server has not been started.

**System Action:** The client continues.

**User Response:** See your APPC Programming Reference information for further the particular error, or contact your system administrator.

**CCL4656 Mode** *mode* **not recognized by partner LU, APPC return code** *primary RC*, *secondary RC*

**Explanation:** The APPC verb CNOS failed as the partner LU does not recognize the specified mode name. The connection to the specified server has not been started.

**System Action:** The client continues.

**User Response:** See your APPC Programming Reference information for further the particular error, or contact your system administrator.

---

**CCL4657 SNA Server has no active services**

**Explanation:** The client SNA protocol driver was unable to open a link to the specified server because SNA server has no active services.

**System Action:** The client continues.

**User Response:** Activate the relevant service via SNA Server Administration or contact your system administrator.

---

**CCL4658 WinAPPCStartup failed with return code** *Windows APPC RC*

**Explanation:** The client SNA protocol driver was unable to open a link to the specified server because it failed to register itself with the currently installed Windows APPC implementation.

**System Action:** The client continues.

**User Response:** Contact your system administrator.

---

**CCL4659 Local LU** *LU name* **is not defined on an active service/node**

**Explanation:** The client SNA protocol driver was unable to open any link using this LU because it is not defined on an active SNA service/node.

**System Action:** The client continues.

**User Response:** Activate the relevant service SNA service or define the LU to an active SNA service/node or correct the *LocalLUName* parameter in your client initialization file and retry.

---

**CCL4660 Partner LU** *partner LU name* **is not defined in a connection with local LU** *LU name*

**Explanation:** The client SNA protocol driver was unable to open any link using this partner LU because it is not defined in a connection with the specified local LU on an active SNA service/node.

**System Action:** The client continues.

**User Response:** Define a connection between the specified local and partner LUs on an active SNA service/node or correct the *NetName* parameter in your client initialization file and retry.

---

**CCL4661 SNA stack component not loaded or terminated, APPC return code** *primary RC*

**Explanation:** APPC could not execute the verb because a component of the product providing the SNA transport could not be loaded or terminated whilst processing the verb.

**System Action:** The client continues.

**User Response:** Contact your system administrator.

---

**CCL4662 Load of SNA stack library** *DLL name* **failed, return code** *Windows return code*

**Explanation:** LoadLibrary failed, the client could not load the specified dynamic link library.

**System Action:** The client SNA protocol driver is not loaded.

**User Response:** Install and configure one of the supported products for SNA transport, if this has not already been done, and ensure that the DLL is in the LoadLibrary search path; then retry. Or contact your system administrator.

---

**CCL4663 Load of client driver** *DLL name* **by TP program** *EXE name* **failed with return code** *Windows return code*

**Explanation:** LoadLibrary failed, the client inbound ATI TP could not load the specified dynamic link library.

**System Action:** The ATI request is rejected.

**User Response:** Contact your system administrator.

---

**CCL4664 TP program** *EXE name* **failed with an internal error, return code** *value*

**Explanation:** The client inbound ATI TP failed due to an internal error.

**System Action:** The ATI request is rejected.

**User Response:** Contact your system administrator.

---

**CCL4665 TP program not available on** *server*, **APPC return code** *primary RC*, *secondary RC*, **retry**

**Explanation:** APPC could not start a remote TP program retrying without intervention may succeed.

**System Action:** The client continues.

**User Response:** Retry the operation. If retrying does not succeed, see your APPC Programming Reference information for further information or contact your system administrator.

---

**CCL4666 TP program not available on** *server*, **APPC return code** *primary RC*, *secondary RC*, **no retry**

**Explanation:** APPC could not start a remote TP program and intervention is required to correct the problem. The remote server is probably not available.

**System Action:** The client continues.

**User Response:** Ensure the remote server is available and if the problem persits ee your APPC Programming Reference information for further information or contact your system administrator.

---

**CCL4667 SNA Server not configured for inbound requests, APPC return code** *primary RC*, *secondary RC*

**Explanation:** The APPC verb RECEIVE_ALLOCATE failed as SNA Server has not been correctly configured to receive inbound ATI requests.

**System Action:** The client continues.

**User Response:** If inbound ATI request processing is required, refer to the CICS Client Administration book on how to configure SNA Server.

---

**CCL4668 SNA node not started, APPC return code** *primary RC*

**Explanation:** The SNA node has not been started.

**System Action:** The client continues.

**User Response:** Start the SNA node and retry the operation.

---

**CCL4669 Real LU names not supported by current client SNA driver.**

**Explanation:** The client SNA driver specified in the client initialization file does not support LU names specified as real names.

**System Action:** The connection to the server is not started.

**User Response:** Specify the local and partner LU names in the client initialization file as alias names, set the *LUAliasNames* parameter to *Y* and retry.

---

**CCL4670 Communications subsystem abended, APPC return code** *primary RC*, *secondary RC*

**Explanation:** Either the APPC provider has abended or has not been configured properly on the client machine.

**System Action:** The connection to the server is not started.

**User Response:** Check the configuration of the APPC provider on the CICS client machine. If using NWSAA v2.2, check that the NDS configuration for the NWSAA client is correct.

---

**CCL4671 LU alias** *LU alias name* **is not defined**

**Explanation:** The specified LU alias is not defined in the APPC provider.

**System Action:** The connection to the server is not started.

**User Response:** If using alias LU names, define the LU alias and retry. If using real LU names, set the *LUAliasNames* parameter to *N* in the client initialisation file and retry.

---

**CCL4672 Partner LU alias** *LU alias name* **or mode** *mode name* **is unknown, APPC return code** *primary RC*, *secondary RC*

**Explanation:** If using real LU names, the specified mode is not defined. If using alias LU names, either the partner LU alias supplied in the *Netname* parameter in the client initialisation file is not defined, or the mode is not defined, in the APPC provider.

**System Action:** The connection to the server is not started.

**User Response:** Check that the parameters in the client initialisation match the definitions in the APPC provider.

---

**CCL4673 Conversation with** *server* **failed, APPC return code** *primary RC*, *secondary RC*

**Explanation:** There has been a failure to allocate a conversation or to receive data from the specified server. The link to the server has probably failed.

**System Action:** The client continues.

**User Response:** See your APPC Programming Reference information for further information or contact your system administrator.

---

**CCL4674 TP program** *TP name* **abended on server** *server*, **APPC return code** *primary RC*

**Explanation:** The specified TP program abended on the given server. If the TP program is CTIN, then this may be because the server has been recycled since running CCIN.

**System Action:** The client continues. If the TP program is CTIN, the client will attempt to reconnect to the server and retry the request.

**User Response:** Contact your system administrator.

**CCL4675** **Wait on semaphore for ReceiveAndPost failed, return code** *operating system return code*

**Explanation:** An attempt to wait on a semaphore for conversation data, after successful completion of the receive verb, failed.

**System Action:** The client sends a deallocate abend to the server to terminate the conversation.

**User Response:** Contact your system administrator.

**CCL4676** **Protocol driver failed to notify client of a data event, return code** *internal return code*

**Explanation:** The protocol driver's attempt to notify the client of a data event failed.

**System Action:** The protocol driver sends a deallocate abend to the server to terminate the conversation.

**User Response:** Contact your system administrator.

**CCL4677** **Callback for RECEIVE_AND_POST for unknown convid** *APPC conv_id* **received**

**Explanation:** The protocol driver's callback function was called for the specified conversation, but it was not in the list of conversations for which a RECEIVE_AND_POST call had been made.

**System Action:** The event is ignored.

**User Response:** Contact your system administrator.

**CCL5608** **RPC DCE reports:** *error*

**Explanation:** The RPC protocol driver received an error response from DCE. The *error* message is the text describing the error supplied by DCE. This text may help to diagnose DCE problems.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL5609** **RPC (to** *server***) binding to:** *name*

**Explanation:** The RPC protocol driver has established a connection to a server. The *server* value identifies the server. The *name* values give RPC binding used to identify the server.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL5610** **RPC (to** *server***) send data: Length=***length*

**Explanation:** The RPC protocol driver was about to send data to a server. The *server* value identifies the server and the *length* value contains the length of the data to be sent. This message is followed by the data to be sent, which consists of the client data and an RPC protocol header.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL5611** **RPC (to** *server***) receive data: Length=***length***, Response=***resp*

**Explanation:** The RPC protocol driver received data from a server. The *server* value identifies the server and the *length* value contains the length of the data received. This message is followed by the data received, which consists of the data returned to the client and an RPC protocol header. The *resp* value indicates the RPC response code returned to the server.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL5612** **RPC authentication to** *system* **is** *state*

**Explanation:** The RPC authentication for the *system* (either client or server) has been set to *state* (either enabled or disabled). If the state is disabled authenticated RPC will not be possible.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** Ensure the RPC Security Configuration program CCLAUTH has been run to enable authenticated RPC.

**CCL5613** **RPC (to** *server***) server not available**

**Explanation:** The RPC protocol driver was unable to connect to the server *server*.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL5614** **RPC (to** *server***) caller principal:** *name*

**Explanation:** The RPC protocol driver will pass a caller DCE principal to the server to enable a co-ordinated security sign-on. The *name* value shows the DCE principal involved.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

---

**CCL5801 Starting TCP62 failed, TCP62 node is running but AnyNet is not configured**

**Explanation:** The IBM Personal Communications (PComm) node (Windows NT/95) or IBM Communications Server (OS/2) is already running and the configuration does not support AnyNet.

**System Action:** The message is written to the error log and processing continues, but the TCP62 protocol is unavailable for use.

**User Response:** Stop the TCP62 node and then retry the operation.

---

**CCL5802 Starting TCP62 failed due to a TCP62 configuration API call failure**

**Explanation:** A call to the IBM Personal Communications (PComm) (Windows NT/95) or IBM Communications Server (OS/2) configuration API failed.

**System Action:** The message is written to the error log and processing continues, but the TCP62 protocol is unavailable for use.

**User Response:** Collect problem determination data by turning on all tracing and recreating the problem. Also, capture any log data. Contact your service organization.

---

**CCL5803 TCP62 node failed to start**

**Explanation:** The IBM Personal Communications (PComm) node (Windows NT/95) or an IBM Communications Server (OS/2) component failed to start.

**System Action:** The message is written to the error log and processing continues, but the TCP62 protocol is unavailable for use.

**User Response:** Collect problem determination data by turning on all tracing and recreating the problem. Also, capture any log data. Contact your service organization.

---

**CCL5804 TCP62 node started but configuration unsuccessful**

**Explanation:** The IBM Personal Communications (PComm) node (Windows NT/95) or IBM Communications Server (OS/2) started but some configuration was not succsessful. The most likely cause is that the AnyNet component is not installed.

**System Action:** The message is written to the error log and processing continues, but the TCP62 protocol is unavailable for use.

**User Response:** Stop the TCP62 node. Ensure that the AnyNet component of the communications product is installed and retry. If the problem persists, collect problem determination data by turning on all tracing and recreating the problem. Also, capture any log data. Contact your service organization.

---

**CCL5805 TCP62 node already running**

**Explanation:** The IBM Personal Communications (PComm) node (Windows NT/95) or IBM Communications Server (OS/2) is already running using different parameters than those supplied on starting TCP62. The domain name suffix or CP name supplied in the client initialization file may not have been used and the AnyNet timers may not have the default values.

**System Action:** The message is written to the error log and processing continues, The TCP62 protocol is available for use.

**User Response:** If problems are experienced, stop the CICS Client, stop the TCP62 node and then restart the CICS Client.

---

**CCL5806 Dynamic *type* name generation failed**

**Explanation:** Dynamic name generation failed. Likely causes are TCP/IP is not installed, configured and active, or an invalid partner LU name, if *type* is PLU.

**System Action:** The message is written to the error log and processing continues.

**User Response:** Ensure that TCP/IP is configured and active and that the *NetName* parameter in the client initialization file is correct.

---

**CCL5807 Starting TCP62 failed with TCP62 return code *primary RC*, *secondary RC***

**Explanation:** The TCP62 verb START_TN62 failed with the specified return code.

**System Action:** The message is written to the error log and processing continues, but the TCP62 protocol is unavailable for use.

**User Response:** Refer to the TCP62 communications product's system management programming reference for information on the return code, or contact your system administrator.

---

**CCL5808 CP name *CP name* is invalid**

**Explanation:** The specified CP name in the *CPName* parameter in your client initialization file is invalid. This parameter must contain either a real fully-qualified CP name or a template for a fully- qualified CP name. The net ID must not contain any template replacement characters and the CP name must not begin with a replacement character.

**System Action:** The message is written to the error log and processing continues, but the TCP62 protocol is unavailable for use.

**User Response:** Correct the *CPName* parameter in your client initialization file and retry.

---

**CCL5809 Local LU name *LU name* is invalid**

**Explanation:** The specified LU name in the *LocalLUName* parameter in your client initialization file is invalid. The LU name must not begin with a replacement character.

**System Action:** The message is written to the error log and processing continues, but the server connection is not started.

**User Response:** Correct the *LocalLUName* parameter in your client initialization file and retry.

---

**CCL5810 Partner LU name *partner LU name* is invalid**

**Explanation:** The specified partner LU name in the *NetName* parameter in your client initialization file is invalid. This parameter must contain either a real fully-qualified partner LU name or a template for a fully-qualified partner LU name. The net ID must not contain any template replacement characters and the partner LU name must not begin with a replacement character.

**System Action:** The message is written to the error log and processing continues, but the server connection is not started.

**User Response:** Correct the *NetName* parameter in your client initialization file and retry.

---

**CCL5811 Define local LU name *LU name* failed with TCP62 return code *primary RC*, *secondary RC***

**Explanation:** The TCP62 verb DEFINE_LOCAL_LU_TN62 failed with the specified return code.

**System Action:** The message is written to the error log and processing continues, but the server connection is not started.

**User Response:** Refer to the TCP62 communications product's system management programming reference for information on the return code, or contact your system administrator.

---

**CCL5812 Define partner LU name *partner LU name* failed with TCP62 return code *primary RC*, *secondary RC***

**Explanation:** The TCP62 verb DEFINE_PARTNER_LU_TN62 failed with the specified return code.

**System Action:** The message is written to the error log and processing continues, but the server connection is not started.

**User Response:** Refer to the TCP62 communications product's system management programming reference for information on the return code, or contact your system administrator.

---

**CCL5813 Define TP CRSR failed with TCP62 return code *primary RC*, *secondary RC***

**Explanation:** The CRSR TP could not be defined to handle inbound ATI requests.

**System Action:** The message is written to the error log and processing continues, but without TCP62 inbound ATI request processing.

**User Response:** Refer to the TCP62 communications product's system management programming reference for information on the return code, or contact your system administrator.

---

**CCL5814 Define mode *mode name* failed with TCP62 return code *primary RC*, *secondary RC***

**Explanation:** The APPC verb DEFINE_MODE failed with the specified return code.

**System Action:** The message is written to the error log and processing continues, but the server connection is not started.

**User Response:** Refer to the TCP62 communications product's system management programming reference for information on the return code, or contact your system administrator.

---

**CCL5815 Query mode definition *mode name* failed with TCP62 return code *primary RC*, *secondary RC***

**Explanation:** The APPC verb QUERY_MODEDEFINITION failed with the specified return code.

**System Action:** The message is written to the error log and processing continues, but the server connection is not started.

**User Response:** Refer to the TCP62 communications product's system management programming reference for information on the return code, or contact your system administrator.

---

**CCL5816 Unable to create TCP62 configuration as the keylock is on**

**Explanation:** The Communications Server configuration for TCP62 cannot be created as the Communications Server keylock feature is enabled.

**System Action:** The message is written to the error log and processing continues, but the server connection is not started.

**User Response:** Unlock the keylock feature and retry.

---

**CCL5817 Stopping TCP62 failed with TCP62 return code** *primary RC*, *secondary RC*

**Explanation:** The TCP62 verb STOP_TN62 failed with the specified return code.

**System Action:** The message is written to the error log and processing continues.

**User Response:** Refer to the TCP62 communications product's system management programming reference for information on the return code, or contact your system administrator.

---

**CCL5818 SNASVCMG is not a valid mode name for data communication**

**Explanation:** NetWare LU6.2 does not accept SNASVCMG as the mode name for a single session connection to communicate data between transaction programs.

**System Action:** The connection to the partner LU is not started.

**User Response:** Change the *Modename* parameter in your client initialization file and retry.

---

**CCL5819 Mode name for** *server* **connection is not a type A symbol string**

**Explanation:** The mode name did not consist only of uppercase letters and numbers.

**System Action:** The connection to the partner LU is not started.

**User Response:** Correct the *Modename* parameter in your client initialization file and retry.

---

**CCL7101 Internal transport error (***function***,** *error***)**

**Explanation:** An internal emulator function failed.

**System Action:** The message is written to the error log and the emulator terminates. The function name and error code are logged.

**User Response:** If the problem persists, contact your service organization.

---

**CCL7102 Internal OS/2 function error (***function***,** *error***)**

**Explanation:** An internal OS/2 function failed.

**System Action:** The message is written to the error log and the emulator terminates. The function name and error code are logged.

**User Response:** If the problem persists, contact your service organization.

---

**CCL7103 Internal Windows function error (***function***)**

**Explanation:** An internal Windows function failed.

**System Action:** The message is written to the error log and the emulator terminates. The function name is logged.

**User Response:** If the problem persists, contact your service organization.

---

**CCL7104 Emulator has insufficient memory to create an internal control structure**

**Explanation:** The emulator is unable to allocate enough memory to create an internal control structure.

**System Action:** The message is written to the error log and the emulator terminates.

**User Response:** Try altering the system configuration to provide more free memory when the emulator is started, then restart the emulator.

---

**CCL7105 Emulator received incorrect sized data block(s) (***size1***,** *size2***)**

**Explanation:** The emulator has received incorrect data from the CICS Client.

**System Action:** The message is written to the error log and the emulator continues.

**User Response:** If the problem persists, contact your service organization.

---

**CCL7106 Emulator ATI table error**

**Explanation:** The emulator has received too many outstanding ATI requests from the server.

**System Action:** The message is written to the error log and the emulator continues.

**User Response:** Try to establish why the emulator is receiving large numbers of ATI requests while being unable to initiate them. If the problem persists contact your service organization.

---

**CCL7110 CICS_EciInitializeExit returned rc =** *rc*

**Explanation:** The ECI user exit function CICS_EciInitializeExit was called and returned a return code of *rc*.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

---

**CCL7111 CICS_EciTerminateExit returned rc =** *rc*

**Explanation:** The ECI user exit function CICS_EciTerminateExit was called and returned a return code of *rc*.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL7112 CICS_EciExternalCallExit1 returned rc =** *rc*

**Explanation:** The ECI user exit function CICS_EciExternalCallExit1 was called and returned a return code of *rc*.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL7113 CICS_EciExternalCallExit2 returned rc =** *rc*

**Explanation:** The ECI user exit function CICS_EciExternalCallExit2 was called and returned a return code of *rc*.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL7114 CICS_EciSystemIdExit returned rc =** *rc*

**Explanation:** The ECI user exit function CICS_EciSystemIdExit was called and returned a return code of *rc*.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL7115 CICS_EciDataSendExit returned rc =** *rc*

**Explanation:** The ECI user exit function CICS_EciDataSendExit was called and returned a return code of *rc*.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL7116 CICS_EciDataReturnExit returned rc =** *rc*

**Explanation:** The ECI user exit function CICS_EciDataReturnExit was called and returned a return code of *rc*.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL7117 CICS ECI user exit loaded,** *DLLPathName*

**Explanation:** The ECI user exit CICSECIX.DLL has been successfully loaded, the full path name being *DLLPathName*.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL7118 About to call CICS_ECIEXITINIT**

**Explanation:** The ECI user exit initialization function CICS_ECIEXITINIT is about to be called.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL7119 CICS ECI user exit active flag =** *flag*

**Explanation:** The ECI user exit initialization has completed and the boolean ECI exit active *flag* is 0 for inactive and 1 for active.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL7120 About to call ECI user exit**

**Explanation:** An ECI user exit function is about to be called.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL7121 CICS EPI user exit loaded,** *DLLPathName*

**Explanation:** The EPI user exit CICSEPIX.DLL has been successfully loaded, the full path name being *DLLPathName*.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL7122 About to call CICS_EPIEXITINIT**

**Explanation:** The EPI user exit initialization function CICS_EPIEXITINIT is about to be called.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL7123 CICS EPI user exit active flag =** *flag*

**Explanation:** The EPI user exit initialization has completed and the boolean EPI exit active *flag* is 0 for inactive and 1 for active.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL7124 CICS EPI user exit unloaded,** *DLLPathName*

**Explanation:** The EPI user exit CICSEPIX.DLL has been successfully unloaded, the full path name being *DLLPathName*.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL7125 CICS_EpiInitializeExit returned rc =** *rc*

**Explanation:** The EPI user exit function CICS_EpiInitializeExit was called and returned a return code of *rc*.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL7126 CICS_EpiTerminateExit returned rc =** *rc*

**Explanation:** The EPI user exit function CICS_EpiTerminateExit was called and returned a return code of *rc*.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL7127 CICS_EpiAddTerminalExit returned rc =** *rc*, **System** *system*, **NetName** *netname*, **DevType** *devtype*

**Explanation:** The EPI user exit function CICS_EpiAddTerminalExit was called and returned a return code of *rc*. The output values of System, NetName and DevType are provided.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL7128 CICS_EpiTermIdExit returned rc =** *rc*, **TermIndex** *termindex*, **System** *system*

**Explanation:** The EPI user exit function CICS_EpiTermIdExit was called and returned a return code of *rc*. The values of TermIndex and System are provided.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL7129 CICS_EpiStartTranExit returned rc =** *rc*, **TranId** *tranid*, **Data size** *size*:

**Explanation:** The EPI user exit function CICS_EpiStartTranExit was called and returned a return code of *rc*. The values of TranId and Data Size are provided, along with a dump of the data itself.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL7130 CICS_EpiReplyExit returned rc =** *rc*, **TermIndex** *termindex*, **Data size** *size*:

**Explanation:** The EPI user exit function CICS_EpiReplyExit was called and returned a return code of *rc*. The values of TermIndex and Data Size are provided, along with a dump of the data itself.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL7131 CICS_EpiDelTerminalExit returned rc =** *rc*, **TermIndex** *termindex*

**Explanation:** The EPI user exit function CICS_EpiDelTerminalExit was called and returned a return code of *rc*. The value of TermIndex is provided.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL7132 CICS_EpiGetEventExit returned rc =** *rc*, **TermIndex** *termindex*, **EventData:**

**Explanation:** The EPI user exit function CICS_EpiGetEventExit was called and returned a return code of *rc*. The value of TermIndex is provided, along with a dump of the EventData.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL7133 CICS_EpiTranFailExit returned rc =** *rc*, **TermIndex** *termindex*, **EventData:**

**Explanation:** The EPI user exit function CICS_EpiTranFailExit was called and returned a return code of *rc*. The value of TermIndex is provided, along

with a dump of the EventData.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

---

**CCL7134 CICS_EpiSystemIdExit returned rc = *rc*, System *system*, NetName *netname*, DevType *devtype***

**Explanation:** The EPI user exit function CICS_EpiSystemIdExit was called and returned a return code of *rc*. The output values of System, NetName and DevType are provided.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

---

**CCL7629 Unable to read Load Manager INI file *filename* (reason = *error*)**

**Explanation:** The client is unable to read the Load Manager INI configuration file.

**System Action:** The message is written to the log and the Load Manager is disabled.

**User Response:** Make sure that the INI file being read exists on the disk, and that it is not read-protected. If the problem persists, contact your service organisation.

---

**CCL7639 Duplicate parameter *parameter* found - ignored**

**Explanation:** This is a warning. A duplicate parameter heading has been detected in your Load Manager INI file.

**System Action:** The client will continue to read the Load Manager INI file, but will ignore all duplicate parameters will be ignored.

**User Response:** If you do not want this warning to appear in the log, make sure that there are no duplicate parameters in your Load Manager INI file.

---

**CCL7640 Duplicate section *section* found - ignored**

**Explanation:** This is a warning. A duplicate section heading has been detected in your Load Manager INI file.

**System Action:** The client will continue to read the Load Manager INI file, but will ignore all duplicate sections will be ignored.

**User Response:** If you do not want this warning to appear in the log, make sure that there are no duplicate sections in your Load Manager INI file.

---

**CCL7643 Incomplete or missing section *section* - error**

**Explanation:** Your Load Manager INI file is invalid, and the client could not read the specified section.

**System Action:** The Load Manager will be disabled.

**User Response:** Make sure that the Load Manager INI file is valid.

---

**CCL7644 Cannot find parameter *parameter* - error**

**Explanation:** A mandatory parameter could not be found in the Load Manager INI file.

**System Action:** The Load Manager will be disabled.

**User Response:** Make sure that the Load Manager INI file is valid.

---

**CCL7645 Paramater *parameter* has invalid value *value***

**Explanation:** The specified parameter has an invalid value.

**System Action:** The Load Manager will be disabled.

**User Response:** Make sure that the Load Manager INI file is valid.

---

**CCL7646 Value for parameter *parameter* is too large**

**Explanation:** The value of the specified parameter is too large.

**System Action:** The Load Manager will be disabled.

**User Response:** Make sure that the value specified is within the required range.

---

**CCL7135 About to call EPI user exit**

**Explanation:** An EPI user exit function is about to be called.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

---

**CCL8601 Exception *exception* raised in *class*::*method* [during *call* request]**

**Explanation:** The exception named, one of the Ccl::ExCode enumeration values, was raised during the execution of the *class*::*method* method during the server request indicated by the *call* value. An exception object encapsulating details of the exception is passed to CclECI::handleException or CclEPI::handleException and/or thrown via the C++ exception mechanism.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL8602 Sending *call* request to server**

**Explanation:** The client class library is about to make a server call. The *call* value indicates the request type.

**System Action:** The message is written to the trace file if tracing is enabled, followed by a dump of the flow object state.

**User Response:** None.

**CCL8603 Received *call* reply from server**

**Explanation:** The client class library has received a reply from the server. The *call* value indicates the request type.

**System Action:** The message is written to the trace file if tracing is enabled, followed by a dump of the flow object state.

**User Response:** None.

**CCL8604 EPI call *calltype*: RC=*error*, Event=*event***

**Explanation:** The client class library completed an EPI call. The *error* value indicates any errors, the *event* value indicates the EPI event returned on GetEvent calls.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL8605 Method *class*::*method* invoked: State=*state*, ExCode=*exception***

**Explanation:** The client class library method indicated by the *class* and *method* values has been invoked. The *state* and *exception* values indicate the current object state and exception code.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL9119 Attempt to get addressability to an IPC queue element failed with rc = *value1***

**Explanation:** On a Windows platform an attempt to get addressability to a shared memory element has failed for some operating system reason.

**System Action:** The message is written to the error log and the client tries to continue, although attempts to reference this shared memory element will result in a trap.

**User Response:** If the problem persists, contact your service organization.

**CCL9300 Invalid number of parameters (*count*) in call to CICS_RexxEciDrop**

**Explanation:** One or more parameters were specified on the call to this function. This function does not expect to be called with any parameters.

**System Action:** The message is written to the error log and the call fails.

**User Response:** Modify your application to use the correct form of the function call.

**CCL9301 Invalid number of parameters (*count*) in call to CICS_RexxEciLoad**

**Explanation:** One or more parameters were specified on the call to this function. This function does not expect to be called with any parameters.

**System Action:** The message is written to the error log and the call fails.

**User Response:** Modify your application to use the correct form of the function call.

**CCL9302 Invalid number of parameters (*count*) in call to CICS_RexxEciCall**

**Explanation:** This function requires exactly two parameters. Some other number of parameters was specified in the function call.

**System Action:** The message is written to the error log and the call fails.

**User Response:** Modify your application to use the correct form of the function call.

**CCL9303 Memory allocation failed. DosAllocMem return code: *error***

**Explanation:** The client was unable to allocate the amount of memory required to hold the returned data. The *error* value indicates the failing return code.

**System Action:** The message is written to the error log and the call fails.

**User Response:** Try reducing the amount of data requested and try again. If the problem persists contact your support organization.

**CCL9304 Invalid call type (*call_type*)**

**Explanation:** An invalid value (as indicated by *call_type*) was specified for the ECI call type.

**System Action:** The message is written to the error log and the call fails.

**User Response:** Correct the program so that a supported value is specified for the ECI call type.

**CCL9305 The commarea size specified exceeds the length of data supplied**

**Explanation:** The value specified for the length of the commarea is larger than the amount of data supplied.

**System Action:** The message is written to the error log and the call fails.

**User Response:** Correct the program so that the specified length of the commarea is equal to, or exceeds, the amount of data being passed to the ECI.

**CCL9306 The value specified for Extended Mode (*value*) is invalid**

**Explanation:** An invalid value (as indicated by *value*) has been specified for the ECI extended mode parameter.

**System Action:** The message is written to the error log and the call fails.

**User Response:** Correct the program so that a supported value is specified for the extended mode parameter.

**CCL9307 The ECI version specified (*version*) is not valid**

**Explanation:** An invalid value (as indicated by *value*) has been specified for the ECI version parameter.

**System Action:** The message is written to the error log and the call fails.

**User Response:** Correct the program so that a supported value is specified for the ECI version parameter.

**CCL9308 Invalid number of parameters (*count*) in call to CICS_RexxEciListSystems**

**Explanation:** This function requires exactly three parameters. Some other number of parameters was specified in the function call.

**System Action:** The message is written to the error log and the call fails.

**User Response:** Modify your application to use the correct form of the function call.

**CCL9309 RxCICS_Eci_Drop called. Debug mode = *dbg*, number of arguments = *argc***

**Explanation:** The client has received a request to deregister the Rexx functions for the ECI. A non-zero value of dbg indicates that the function is being run in debug mode. The value of argc indicates the number of parameters specified on the function call.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL9310 RxCICS_Eci_Load called. Debug mode = *dbg*, number of arguments = *argc***

**Explanation:** The client has received a request to register the Rexx functions for the ECI.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL9311 RxCICS_ExternalCall called. Debug mode = *dbg*, number of arguments = *argc***

**Explanation:** The client has received a request to call CICS_ExternalCall using the Rexx interface. A non-zero value of dbg indicates that the function is being run in debug mode. The value of argc indicates the number of parameters specified on the function call.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL9312 ECI stem name is *stemname***

**Explanation:** The value *stemname* identifies the name of the Rexx stem variable which contains the ECI parameter block.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL9313 Base address for returned commarea allocated by DosAllocMem is *addr***

**Explanation:** The client has successfully allocated a buffer in memory to hold the data returned in the commarea by the call to the ECI. The value of addr indicates the address at which the buffer has been allocated.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL9314 eci_call_type = *type***

**Explanation:** The value *type* identifies the type of call which has been requested.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL9315 eci_program_name =** *progname*

**Explanation:** The value *progname* identifies the name of the program which is to be invoked by this ECI call.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL9316 eci_userid =** *userid*

**Explanation:** The value *userid* identifies the userid to be used for security checking.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL9317 eci_password =** *password*

**Explanation:** The value *password* identifies the password to be used for security checking.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL9318 eci_transid =** *tid*

**Explanation:** The value *tid* identifies the TranID specified for this call.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL9319 User specified commarea length was** *length*

**Explanation:** The value *length* indicates an explicit length for the commarea, as set by the caller. A value of *-1* indicates that no specific value was provided and so the length of the user supplied commarea will be used instead.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL9320 eci_commarea_length =** *len*

**Explanation:** A commarea length of *len* will be used on the ECI call. It was either set explicitly by the caller or else is taken as the length of the commarea passed as input to the call.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL9321 eci_timeout =** *seconds*

**Explanation:** The value *seconds* specifies the maximum time that the request from the application should be allowed to take. This parameter is provided for backwards compatibility only and should normally be set to 0.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL9322 eci_extend_mode =** *mode*

**Explanation:** The value *mode* specifies the extended mode for this ECI call.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL9323 eci_message_id =** *msgid*

**Explanation:** The value *msgid* is a user-provided reference to an asynchronous call.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL9324 eci_luw_token =** *luwid*

**Explanation:** The value *luwid* is an identifier for a logical unit of work.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL9325 eci_sysid =** *sysid*

**Explanation:** The value *sysid* is reserved for future use and should generally be left null.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL9326 eci_version =** *version*

**Explanation:** The value *version* specified the version of the ECI for which the application is coded. This should be set to the value *ECI_VERSION_1* or *ECI_VERSION_1A*.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL9327 eci_system_name =** *name*

**Explanation:** The value *name* specifies the name of the system to which the ECI call should be directed.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL9328 eci_userid2 =** *userid*

**Explanation:** The value *userid* specifies a userid to be used for security checking.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL9329 eci_password2 =** *password*

**Explanation:** The value *password* specifies a password to be used for security checking.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL9330 eci_tpn =** *tpn*

**Explanation:** The value *tpn* specifies a transaction name under which this program should execute in the target system.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL9331 Returned eci_return_code =** *rc*

**Explanation:** The value *rc* is the return code from the ECI call.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** Refer to the CICS Client/Server Programming book for the meaning of these codes and the action required.

**CCL9332 Returned eci_abend_code =** *abend*

**Explanation:** The value *abend* is the abend code for a failed program.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** Refer to the CICS Client/Server Programming book for the meaning of these codes and the action required.

**CCL9334 Returned eci_luw_token =** *luwid*

**Explanation:** The value *luwid* is an identifier for a logical unit of work. This is the value which was returned by the call to the ECI.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL9335 Returned eci_commarea_length =** *length*

**Explanation:** The value *length* specifies the amount of data returned in the commarea after the call to the ECI.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL9336 Cammarea base address =** *addr***, length =** *length*

**Explanation:** The call to the ECI has completed successfully. The value of *addr* indicates the location in memory of the returned commarea and the value of *length* indicates the amount of data returned in the commarea.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL9337 Error during ECI call. Commarea not returned.**

**Explanation:** The call to the ECI has completed but an error was detected. No data is returned to the calling application in the commarea.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** Information on the type of error can be determined from earlier entries in the trace log.

**CCL9338 RxCICS_EciListSystems called. Debug mode =** *dbg***, number of arguments =** *argc*

**Explanation:** The client has received a request to call CICS_EciListSystems using the Rexx interface. A non-zero value of dbg indicates that the function is being run in debug mode. The value of argc indicates the number of parameters specified on the function call.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL9339 System count variable is** *name*

**Explanation:** The value *name* indicates the variable into which the number of systems found will be written.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL9340 List stem name is** *stemname*

**Explanation:** The value *stemname* specifies the stem of the variable into which the returned system information is to be written.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL9341 Number of systems found is** *count*

**Explanation:** The value of *count* specifies the number of systems found. Data can only be returned for a number of systems up to the implementation maximum (currently 100). If the value of *count* is larger than the maximum number specified then some data will have been lost.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL9342 System name =** *sysname* **(Description =** *desc***)**

**Explanation:** For each system returned to the caller, the value *sysname* indicates the name of the system and the value *desc* its associated description.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL9343 CICS_EciListSystems call completed successfully**

**Explanation:** The call has completed successfully and control is being returned to the calling program.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** None.

**CCL9344 Error during CICS_EciListSystems call. Return code:** *rc*

**Explanation:** An error was encountered during execution of the code. The value of *rc* indicates the failing error code.

**System Action:** The message is written to the trace file if tracing is enabled.

**User Response:** Refer to the CICS Client/Server Programming book for the meaning of these codes and the action required.

# Appendix. Notices

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

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

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

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

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

**The following paragraph does not apply in the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore this statement may not apply to you.

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

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM United Kingdom Laboratories, MP151, Hursley Park, Winchester, Hampshire, England, SO21 2JN. Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

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

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

## Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

AnyNet
CICS
CICS/400
IBM
OS/2

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

Other company, product, and service names may be trademarks or service marks of others.

Program Number: 5648-B42

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

GC34-5452-01