



## CTITest Utility Program

---

This chapter describes the CTITest utility program, and the associated utility programs DispCTI and dumplog. It includes:

- An overview of CTITest
- Instructions on how to set up CTITest
- Configuration information
- The CTITest command set
- Scenario examples
- Capturing and displaying events (DispCTI)
- Using dumplog to view server information

### CTITest Overview

CTITest is a utility program that can be used in place of an interactive CTI client for debugging or demonstration purposes.

CTITest is a console program that runs in a window and displays CTI events to the screen. Depending on how CTITest is configured, it attaches to the CTI Server via a socket connection and then displays either all the events that are occurring on the switch (simulating a CTI Bridge client application) or only the events which pertain to a single desk top (simulating an agent workstation client application).

CTITest shows you every event that your integrating application receives. If CTITest has the same configuration that your application does, then the expected events and switch behavior will be seen in the CTITest program.

The CTITest program can be found in the tools directory of the Cisco CTI CD.

## CTITest Directory Setup

We recommend that you make a new directory for just the CTITest program and trace DLL:

- 
- Step 1 Make a new directory. Give it any name you like; for example, **c:\program files\geotel\cti desktop\ctitester**.
  - Step 2 Copy the **CTITest.exe** program from the CD into the newly created **\ctitester** directory.
  - Step 3 Copy the **icrmsgs.dll** library from the CD into the **\ctitester** directory.
- 

## Creating a Shortcut

You may find it helpful to create a shortcut. This allows you to easily start up the application and have a standard working directory that all your capture files will go to.

To create a shortcut:

- 
- Step 1 Right-click **CTITest.exe** and select **Create Shortcut**. This creates the shortcut.
  - Step 2 Drag-and-drop the shortcut to wherever you want it; for example, to your desktop.
  - Step 3 Change the name of the shortcut, if you wish.
  - Step 4 Right-click the shortcut icon and select **Properties**.
  - Step 5 Select the **Shortcut** tab.

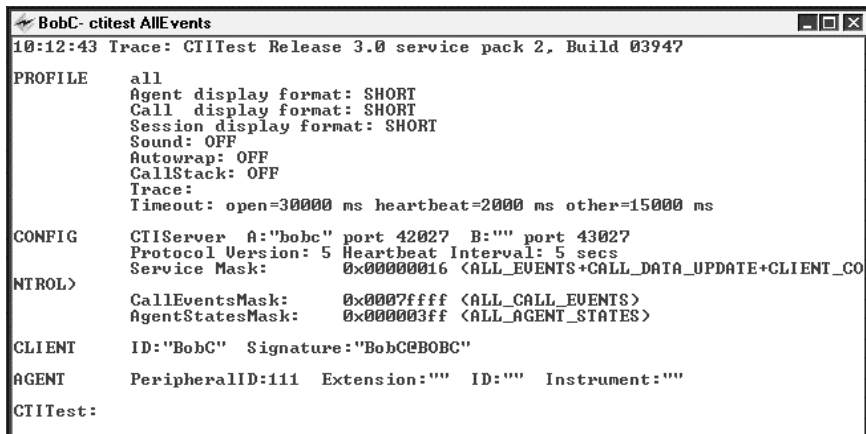
- Step 6** The Target field contains the complete path of your file. If you intend to have more than one version of CTITest, include a profile name, so as to retain a particular configuration for each of the versions. (If you are just using one version of CTITest, no profile name is necessary, since the most recent configuration of CTITest will automatically be preserved.) Assuming our example path, and that you want to use all as the profile name, the Target field entry should be:  
**c:\program files\geotel\cti desktop\ctitester\ctitest.exe /p all**
- Step 7** Click **OK**.

## Starting CTITest

No command line arguments are required to start CTITest. You can double-click the shortcut you created in the preceding section to start the application.

The screen that appears will look something like the following.

*Figure 5-1 CTITest Initial Screen*



```

BobC- ctitest AllEvents
10:12:43 Trace: CTITest Release 3.0 service pack 2, Build 03947
PROFILE    all
           Agent display format: SHORT
           Call display format: SHORT
           Session display format: SHORT
           Sound: OFF
           Autowrap: OFF
           CallStack: OFF
           Trace:
           Timeout: open=30000 ms heartbeat=2000 ms other=15000 ms
CONFIG    CTIServer A:"bobc" port 42027 B:"" port 43027
           Protocol Version: 5 Heartbeat Interval: 5 secs
           Service Mask: 0x00000016 <ALL_EVENTS+CALL_DATA_UPDATE+CLIENT_CO
CONTROL>  CallEventsMask: 0x0007ffff <ALL_CALL_EVENTS>
           AgentStatesMask: 0x000003ff <ALL_AGENT_STATES>
CLIENT    ID:"BobC" Signature:"BobCEBOBC"
AGENT     PeripheraID:111 Extension:"" ID:"" Instrument:""
CTITest:
  
```

56169

To increase the height or width of the screen buffer:

- 
- Step 1 Right-click the icon in the upper-left corner of the title bar (to the left of BobC in [Figure 5-1](#)).
  - Step 2 Select **Properties**.
  - Step 3 Select the **Layout** tab.
  - Step 4 Edit the height and width of the screen buffer size as you wish. (An initial value of 9999 for the height is reasonable.)
  - Step 5 Click **OK**.
  - Step 6 Check the **Modify shortcut** option if you want to permanently change the size of the screen buffer; check the **current window** option if you only want to change the size for the current session.
  - Step 7 Click **OK**.
- 

## CTITest Configuration

There are basically only a handful of items that need to be configured. These configurations are retained in the registry so that this configuration setup need only be performed once. The next time that CTITest is started, the profile defaults to the last used settings.

- 
- Step 1 You need to configure the IP address and port number of the CTI Server by using the following command:

```
config /hostA ctiserver_hostname /portA ctiserver_portnumber
```

Example:

```
config /hostA CiscoHost /portA 42027
```

You should see a change on the CONFIG line immediately.

If you are using CTISim on your computer, rather than using a real CTI Server, you would enter:

```
config /hostA localhost /portA 199
```

**Step 2** Next you need to configure the service mode in which you will interact with the CTI Server. These modes are:

- **ALL\_EVENTS service mode.** All the events for all stations will be delivered to the CTITest application. This would be used for simulating a CTI Bridge client.
- **CLIENT\_EVENTS service mode.** Only the events for a specific desktop station will be delivered to the CTITest application. This would be used for simulating an agent workstation client.
- **CALL\_DATA\_UPDATE service mode.** Allows for data about the call to be sent to the CTITest program.
- **CLIENT\_CONTROL service mode.** Allows CTITest to control the state of the phone or the agent.

Typical CTI Bridge service configuration:

```
config /service ALL_EVENTS+CALL_DATA_UPDATE+  
CLIENT_CONTROL
```

Typical desktop client service configuration:

```
config /service CLIENT_EVENTS+CALL_DATA_UPDATE+  
CLIENT_CONTROL
```

Similarly you can, if you wish, configure the mask of call event messages desired and the mask of agent state events desired. The list of possible values can be obtained by typing help in place of the service or mask, for example:

```
config /callmask /help
```

**Step 3** If you need to, you can change the CTI Server protocol version—discussed in the [“CTI Server Protocol Versions”](#) section on page 2-5. For example, to change to protocol version 3, enter:

```
config /version 3
```

---

## Configuring the Agent

Next, if applicable, enter configuration information for a specific agent and telephone set. This is normally used with a client-mode CTI Bridge application.

Example:

```
agent /periph 1 /ext 23901 /id 23955 /instrument 23901
```

This profile will be retained within the registry.

## Opening and Closing the CTI Server Connection

The open command performs a socket connection to the CTI Server and then issues the appropriate command to the CTI Server to establish messaging communication.

By default, CTITest repeatedly attempts to open a session until it either opens a session, or the close or quit command is given. CTITest automatically attempts to switch to the other side following a failure of the CTI Server.

When a session is opened, all messages received from the CTI Server are displayed on the screen. The messages can be tailored by using the display\_calls and display\_agents commands: long displays all message fields, short displays a one line subset of message fields, off displays no message.

## CTITest Command Set

Some examples, using the following commands, are given in the [“Scenario Examples” section on page 5-11](#).

Table 5-1 CTITest Command Set

Command	Arguments
agent	[/periph N] [/ext N] [/id AGENTID] [/instrument N]
agent_login (login)	ID PASSWORD [/periph N] [/instrument N] [/reason N][[/position N] [/supervisor N] [/autoIN] [/manualIN] [/skill N[.pri] [/skill N[.pri]] ...]
agent_logout (logout)	[/agentID ID] [/password PASSWORD] [/periph N] [/instrument N] [/reason N] [/skill N[.pri] [/skill N[.pri]] ...]
agent_not_ready (not_ready)	[/agentID ID] [/periph N] [/instrument N] [/reason N] [/skill N[.pri] [/skill N[.pri]] ...]
agent_ready (ready)	[/agentID ID] [/periph N] [/instrument N] [/reason N] [/autoIN] [/manualIN] [/skill N[.pri] [/skill N[.pri]] ...]
agent_work_ready (callwork)	[/agentID ID] [/periph N] [/instrument N] [/reason N] [/skill N[.pri] [/skill N[.pri]] ...]
alternate_call (switch)	[/periph N] [/active CALLID.DEVID] [/other CALLID.DEVID] [/stack ACTIVE OTHER]
answer_call (answer)	[/periph N] [/callid CALLID.DEVID] [/stack N] [/instrument N]
autoanswer	[/on] [/off]
autowrap	[/on] [/off]
callstack	[/on] [/off]
cancel_cmd_file	
capture	[/file FILENAME] [/maxsize N] [/off]
clear_call (clear)	[/periph N] [/callid N.DEVID] [/stack N]
clear_connection (leave)	[/periph N] [/instrument N] [/callid N.DEVID] [/stack N]
client	[/id STRING] [/password STRING] [/signature STRING]
close	

Table 5-1 CTITest Command Set (continued)

Command	Arguments
conference_call (conf)	[/periph N] [/active CALLID.DEVID] [/held CALLID.DEVID] [/stack ACTIVE HELD] [/blind DN] [/placement TYPE] [/manner TYPE] [/alertRings N] [/callopt TYPE] [/priority] [/postRoute] [/TG N] [/SG N] [/authorize CODE] [/account CODE] [keyword=VALUE ...]
config	[/hostA NAME] [/portA N] [/hostB NAME] [/portB N] [/version B] [/hb B] [/service MASK[+MASK]] [/callmask MASK[+MASK]] [/agentmask MASK[+MASK]]
consult_call (cc)	DN [/periph N] [/instrument N] [/callid N.DEVID] [/stack N] [/placement TYPE] [/manner TYPE] [/transfer] [/conference] [/alertRings N] [/callopt TYPE] [/priority] [/postRoute] [/TG N] [/SG N] [/authorize CODE] [/account CODE] [keyword=VALUE ...]
debug	[[/no]heartbeat][[/no]badmsgtype] [[/no]forcevalid] [/conndelay SECS][[/maxHBloss N]
deflect_call (deflect)	DEST [/periph N] [/callid N.DEVID] [/stack N]
delete_profile	profileName
display_agents (da)	[/long] [/short] [/off]
display_calls (dc)	[/long] [/short] [/off]
display_info (di)	
display_sessions (ds)	[/long] [/short] [/off]
echo	<STRING>
get_agent_state (state)	[/periph N] [/instrument N]
get_agent_stats (stats)	[/periph N] [/instrument N] [/ext STRING] [/id STRING]

**Table 5-1 CTITest Command Set (continued)**

Command	Arguments
get_skill_stats (sk_stats)	[/periph N] [/skill N] [/skillID N]
hold_call (hold)	[/periph N] [/callid N.DEVID] [/stack N]
link_status (status)	
list_calls (lc)	[/long]
list_links (ll)	[/up] [/down]
list_monitors (lm)	
list_profiles (lp)	
list_sessions (ls)	[/all]
make_call (mc)	DN [/periph N] [/instrument N] [/placement TYPE] [/manner TYPE] [/alertRings N] [/callopt TYPE] [/priority] [/postRoute] [/TG N] [/SG N] [/authorize CODE] [/account CODE] [keyword=VALUE ...]
make_pred_call	DN [/periph N] [/origDeviceID N] [/origLineID N] [/placement TYPE] [/callopt TYPE] [/priority] [/allocState Type] [/destination COUNTRY] [/answerDetect MODE] [/answerDetectTime SECONDS] [/answerDetectCntl1 N] [/answerDetectCntl2 N] [/TG N] [/SG N] [/authorize CODE] [/account CODE] [keyword=VALUE ...]
monitor_change (monchange)	[/monitorID N] [/callmask MASK[+MASK]] [/agentmask MASK[+MASK]]
monitor_start (monstart)	[/periph N] [/call CALLID.DEVID] [/stack N] [/devid DEVID] [/callmask MASK[+MASK]] [/agentmask MASK[+MASK]]
monitor_stop (monstop)	[/monitorID N]
msleep	<MILLISECONDS>
open	[sessionCount]
ping	
profile	[profileName]

Table 5-1 CTITest Command Set (continued)

Command	Arguments
query_device (device)	[/periph N] [/instrument N] [/dev   /addr   /agent]
quit	
read	<FILENAME>
reconnect_call (reconnect)	[/periph N] [/active CALLID.DEVID] [/held CALLID.DEVID] [/stack ACTIVE HELD]
register_variables (rv)	[/callVarMask MASK] [/namedVar NAME] [/namedArray NAME]
release	[/periph N] [/callid N.DEVID] [/stack N]
report_event (report)	[/warning] [/error] [/object NAME] TEXT ...
reset	
retrieve_call (retrieve)	[/periph N] [/callid N.DEVID] [/stack N]
send_DTMF (dtmf)	DTMFdigits [/periph N] [/callid N.DEVID] [/stack N]
session_monitor (snoop)	<ON/OFF> <SESSIONID>
set	[/periph N] [/callid N.DEVID] [/stack N] keyword=VALUE [keyword=VALUE ...]
snapshot_call (snapcall)	[/periph N] [/callid N.DEVID] [/stack N]
snapshot_device (snapdev)	[/periph N] [/instrument N]
sleep	<SECONDS>
sound	[/on] [/off]
start_link (start)	low_id [ - hi_id]
stop_link (stop)	[low_id [ - hi_id]]
timeout	[/open MS] [/heartbeat MS] [/other MS]
trace	[/agent] [/call] [/msgs] [/hb] [/debug] [/all] [/off]

**Table 5-1 CTITest Command Set (continued)**

Command	Arguments
transfer_call (transfer)	[/periph N] [/active CALLID.DEVID] [/held CALLID.DEVID] [/stack ACTIVE HELD] [/blind DN] [/placement TYPE] [/manner TYPE] [/alertRings N] [/callopt TYPE] [/priority] [/postRoute] [/TG N] [/SG N] [/authorize CODE] [/account CODE] [keyword=VALUE ...]
user_msg (send)	DISTRIBUTION [/clientID STRING] [/instrument N] [/ext STRING] [/id STRING] TEXT
wait_cmd_file	<FILENAME> [<COUNT>]
wait_session_closed	<TIMEOUT_SECONDS>
wait_session_open	<TIMEOUT_SECONDS>

## Scenario Examples

The following examples would be typed into the CTITest application. The values assumed for the examples are:

- Agent Id: 23851
- Agent Password: 23852
- Peripheral ID: 1
- Telephone set: 23801

The general syntax for the examples is found in the [“CTITest Command Set” section on page 5-6](#).

## Agent Login

Example:

```
agent_login 23851 23852 /periph 1 /instrument 23801
```

or

```
login 23851 23852 /periph 1 /instrument 23801
```

## Agent Logout

Example:

```
agent_logout /agentID 23851 /periph 1
```

or

```
logout /agentID 23851 /periph 1
```

## Ready to Accept Calls

Example:

```
agent_ready /periph 1 /instrument 23801
```

or

```
ready /periph 1 /instrument 23801
```

## Making a Call

Example:

Making a call to the phone 23802 from agent 23851 on set 23801.

```
make_call 23802 /periph 1 /instrument 23801
```

or

```
mc 23802 /periph 1 /instrument 23801
```

## Answering a Call

Example:

The callid is available from the output typically in the CALL\_DELIVERED\_EVENT displayed when the call is begun.

```
answer_call /periph 1 /callid 487.23801
```

or

```
answer /periph 1 /callid 487.23801
```

## Clearing a Call

Example:

```
clear_call /periph 1 /callid 487.23801
```

or

```
clear /periph 1 /callid 487.23801
```

# Event Capture Facility

The CTITest program has a capture command that writes the messaging information to a file to be looked at later, or to be used to document behavior. The general syntax for capture is found in the [“CTITest Command Set” section on page 5-6](#).

The CTITest program will write the file in the working directory (“start in”) defined in the shortcut for the CTI program. The event capture file is in binary format that can be read by the DispCTI utility.

To turn on the capture facility, enter:

**capture /file *filename***

To turn off the capture facility, enter:

**capture /off**

## Displaying Captured Events: DispCTI

The DispCTI (Display CTI) program will convert binary event capture information and write it into human-readable ASCII text.



### Note

---

The DispCTI utility is only available when you have done a full ICM software/CTI Server installation. It is not available on the Cisco CTI CD.

---

In a command window, change to the directory where the capture file is located, then enter the following command if you want an output file:

**dispcti *capturefile\_filename* > *output\_filename***

The DispCTI program writes to standard out. By redirecting the output to a file, you can use an editor to search or locate information within the file.

## Sample Output from DispCTI

### Example 5-1 DispCTI Release 2.5, Build 03035

```
09:46:33 Trace: Playback active from the start of file Example.cap
09:46:33 Trace: Capture session started at 12/31/1998 09:59:16:234
09:46:33 Trace: First message in playback was recorded at time 12/31/1997 09:59:19:158
ELAPSED_TIME DELTA ENTERPRISE_CTI_MESSAGE_CONTENTS
0:00:00.000 0.000 OPEN_REQ (InvokeID:0x5000000 VersionNumber:3
  IdleTimeout:4294967295 PeripheralID:2 ServiceMask:0x7
  CallMsgMask:0x7ffff AgentStateMask:0x3ff Reserved:0
  Reserved:0 Reserved:0 ClientID:"Example"
  ClientPassword:<0x> ClientSignature:"MyWorkstation"
  AgentID:"1125" )
0:00:00.000 0.000 OPEN_CONF (InvokeID:0x5000000 ServiceMask:0x7
  PGStatus:NORMAL PeripheralType:Aspect
```

```

AgentState:AVAILABLE AgentExtension:"125"
AgentID:"1125" AgentInstrument:"125" )
0:00:10.335 10.335 BEGIN_CALL_EVENT (PeripheralID:2 PeripheralType:Aspect
NumCTIClients:0 CallType:CALLTYPE_PREROUTE_ACD_IN
ConnectionDeviceIDType:CONNECTION_ID_DYNAMIC
ConnectionCallID:4 ConnectionDeviceID:"0"
ANI:"9782755100" DNIS:"0006"
DialedNumber:"8002755010" CED:"1"
RouterCallKeyDay:145000 RouterCallKeyCallID:34320231
CallVariable1:"1234567" CallVariable2:"0"
CallVariable3:"0006" CallVariable5:"26000 123-45-6789
Not Valid " CallVariable6:"1001123-45-6789
26000000000038404LPCFAA" CallVariable7:"THIBAUT,
JOHN C. " CallVariable8:"CISCO
TESPHE " )
0:00:10.736 0.401 CALL_DELIVERED_EVENT (PeripheralID:2 PeripheralType:Aspect
ConnectionDeviceIDType:CONNECTION_ID_DYNAMIC
ConnectionCallID:4 LineHandle:0
LineType:LINETYPE_INBOUND_ACD ServiceNumber:1
ServiceID:1
AlertingDeviceType:DEVID_DEVICE_IDENTIFIER
CallingDeviceType:DEVID_TRUNK_IDENTIFIER
CalledDeviceType:DEVID_DEVICE_IDENTIFIER
LastRedirectDeviceType:DEVID_NONE
LocalConnectionState:LCS_ALERTING EventCause:CEC_NONE
ConnectionDeviceID:"0" AlertingDeviceID:"125"
CallingDeviceID:"1000" CalledDeviceID:"125" )
0:00:10.946 0.210 AGENT_STATE_EVENT (PeripheralID:2 PeripheralType:Aspect
SkillGroupState:TALKING StateDuration:0
SkillGroupNumber:1 SkillGroupID:21
SkillGroupPriority:0 AgentState:TALKING
ClientSignature:"MyWorkstation" AgentID:"1125"
AgentExtension:"125" AgentInstrument:"125" )
0:00:11.036 0.090 CALL_DATA_UPDATE_EVENT (PeripheralID:2
PeripheralType:Aspect NumCTIClients:1
CallType:CALLTYPE_PREROUTE_ACD_IN
ConnectionDeviceIDType:CONNECTION_ID_DYNAMIC
ConnectionCallID:4
NewConnectionDeviceIDType:CONNECTION_ID_DYNAMIC
NewConnectionCallID:4 ConnectionDeviceID:"0"
NewConnectionDeviceID:"0"
CTIClientSignature:"MyWorkstation"
CTIClientTimestamp:0x34aaa421 (12/31/97 14:59:29) )
0:00:11.226 0.190 CALL_ESTABLISHED_EVENT (PeripheralID:2
PeripheralType:Aspect
ConnectionDeviceIDType:CONNECTION_ID_DYNAMIC
ConnectionCallID:4 LineHandle:0
LineType:LINETYPE_INBOUND_ACD ServiceNumber:1

```

## ■ Displaying Captured Events: DispCTI

```

ServiceID:1 SkillGroupNumber:1 SkillGroupID:21
SkillGroupPriority:0
AnsweringDeviceType:DEVID_DEVICE_IDENTIFIER
CallingDeviceType:DEVID_TRUNK_IDENTIFIER
CalledDeviceType:DEVID_DEVICE_IDENTIFIER
LastRedirectDeviceType:DEVID_NONE
LocalConnectionState:LCS_CONNECT EventCause:CEC_NONE
ConnectionDeviceID:"0" AnsweringDeviceID:"125"
CallingDeviceID:"1000" CalledDeviceID:"0006" )
0:00:40.308 29.082 CALL_CLEARED_EVENT (PeripheralID:2 PeripheralType:Aspect
ConnectionDeviceIDType:CONNECTION_ID_DYNAMIC
ConnectionCallID:4 LocalConnectionState:LCS_NONE
EventCause:CEC_NONE ConnectionDeviceID:"0" )
0:00:40.418 0.110 AGENT_STATE_EVENT (PeripheralID:2 PeripheralType:Aspect
SkillGroupState:WORK_READY StateDuration:0
SkillGroupNumber:1 SkillGroupID:21
SkillGroupPriority:0 AgentState:WORK_READY
ClientSignature:"MyWorkstation" AgentID:"1125"
AgentExtension:"125" AgentInstrument:"125" )
0:00:40.538 0.120 CALL_DATA_UPDATE_EVENT (PeripheralID:2
PeripheralType:Aspect NumCTIclients:0
CallType:CALLTYPE_PREROUTE_ACD_IN
ConnectionDeviceIDType:CONNECTION_ID_DYNAMIC
ConnectionCallID:4
NewConnectionDeviceIDType:CONNECTION_ID_DYNAMIC
NewConnectionCallID:4 ConnectionDeviceID:"0"
NewConnectionDeviceID:"0" CallWrapupData:"Autowrap #6
by Example" )
0:00:40.548 0.010 SET_CALL_DATA_REQ (InvokeID:0x5000001 PeripheralID:2
ConnectionDeviceIDType:CONNECTION_ID_DYNAMIC
ConnectionCallID:4 ConnectionDeviceID:"0"
CallWrapupData:"Autowrap #6 by Example" )
0:00:40.659 0.111 SET_CALL_DATA_CONF (InvokeID:0x5000001 )
0:00:41.600 0.941 RELEASE_CALL_CONF (InvokeID:0x5000002 )
0:00:41.600 0.000 RELEASE_CALL_REQ (InvokeID:0x5000002 PeripheralID:2
ConnectionDeviceIDType:CONNECTION_ID_DYNAMIC
ConnectionCallID:4 ConnectionDeviceID:"0" )
0:00:45.375 3.775 END_CALL_EVENT (PeripheralID:2 PeripheralType:Aspect
ConnectionDeviceIDType:CONNECTION_ID_DYNAMIC
ConnectionCallID:4 ConnectionDeviceID:"0" )
0:00:45.415 0.040 AGENT_STATE_EVENT (PeripheralID:2 PeripheralType:Aspect
SkillGroupState:AVAILABLE StateDuration:0
SkillGroupNumber:1 SkillGroupID:21
SkillGroupPriority:0 AgentState:AVAILABLE
ClientSignature:"MyWorkstation" AgentID:"1125"
AgentExtension:"125" AgentInstrument:"125" )
0:00:51.324 5.909 CLOSE_REQ (InvokeID:0x5000003 Status:E_CTI_NO_ERROR )
0:00:51.344 0.020 CLOSE_CONF (InvokeID:0x5000003 )

```

```
09:46:34 Trace: Playback session complete from playback file Example.cap.  
09:46:34 Trace: Closing playback file Example.cap. Current file position 1498
```

## dumplog

The dumplog utility is a program that produces an ASCII text file from the EMS log file for any process on an ICM software system. All ICM software processes use the EMS logging facility for logging information. If you need to see the information regarding a server, you would use the dumplog facility.



### Note

---

The dumplog utility is only available when you have done a full ICM software/CTI Server installation. It is not available on the Cisco CTI CD.

---

For help on dumplog, enter the following in a command window:

```
dumplog /?
```

To use the facility, you need to be in the logfile directory.

To navigate to the CTI Server Gateway (CG), you need to know the peripheral number of the CG. The following command changes to that directory:

```
cdlog customer_number CG_or_PG_number
```

Example:

```
cdlog MySystem CG2a
```

This should place you into the directory in the format:

```
\icr\ customer_number\ CG_or_PG_number\logfiles
```

In our example, this would be:

```
icr\MySystem\CG2a\logfiles
```

dumplog will produce a lot of information. We recommend using the switches to segment the time period you are interested in. For example:

```
dumplog ctisvr /bt 8:00
```

## ■ dumplog

where 8:00 is the current time of the current day.