



Cisco Unified CME Commands: Debug

Revised: October 22, 2009

This chapter contains commands to configure and maintain Cisco Unified Communications Manager Express (formally known as Cisco Unified CallManager Express). The commands are presented in alphabetical order. Some commands required for configuring Cisco Unified Communications Manager Express (Cisco Unified CME) may be found in other Cisco IOS references. Use the reference master index or search online to find these commands.

debug callmonitor

To collect and display debugging traces for call monitor, use the **debug callmonitor** command in privileged EXEC mode. To disable debugging, use the **no** form of this command.

debug callmonitor {all | core | detail | errors | events | hwconf | info | xml}

no debug command {all | core | detail | errors | events | hwconf | info | xml}

Syntax Description

all	All call-monitor debugging traces.
core	Core information debugging traces.
detail	Detailed debugging traces.
errors	Call-monitor error debugging traces.
events	Call-monitor event debugging traces.
hwconf	Debugging traces related to hardware configuration.
info	Call-monitor information debugging traces.
xml	Call-monitor XML encoding debugging traces.

Command Default

There is no default for this command.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.4(11)XW2	This command was introduced.

Examples

The following example is partial output from this command:

```
Router# debug callmonitor all
```

```
Syslog logging: enabled (11 messages dropped, 2 messages rate-limited,  
0 flushes, 0 overruns, xml disabled, filtering disabled)
```

```
No Active Message Discriminator.
```

```
No Inactive Message Discriminator.
```

```
Console logging: disabled  
Monitor logging: level debugging, 0 messages logged, xml disabled,  
filtering disabled  
Buffer logging: level debugging, 444378 messages logged, xml disabled,  
filtering disabled  
Logging Exception size (4096 bytes)  
Count and timestamp logging messages: disabled  
Persistent logging: disabled
```

Trap logging: level informational, 461 message lines logged

Log Buffer (1000000 bytes):

```

Jun  4 22:30:24.222: //CMM/INFO:
Jun  4 22:30:24.222: //CMM/INFO:
Jun  4 22:30:24.222: //CMM/INFO:cmm_notify_trigger() 15, callID 99685, 5114016,
1884814040, 1632257208
Jun  4 22:30:24.222: //CMM/INFO:      target_node 0
Jun  4 22:30:24.222: //CMM/INFO:Lineinfo node Search FAILED
Jun  4 22:30:24.222: //CMM/INFO:create_lineinfo_node
Jun  4 22:30:24.222: //CMM/INFO:      target_node 66AF3714
Jun  4 22:30:24.222: //CMM/INFO:      - dn 4016

Jun  4 22:30:24.222: //CMM/INFO: CallEntry 709C3FB8
Jun  4 22:30:24.222: //CMM/INFO: dstCallID -1
Jun  4 22:30:24.222: //CMM/INFO: line_info 66AF3720, dn 4016
Jun  4 22:30:24.222: //CMM/INFO:      * cmm_crs_proc_tr_rpt_orig
Jun  4 22:30:24.222: //CMM/INFO:      callID = 99685, CG 5114016, GCID
=05591A85-122211DC-8645A1CA-4B604A7A
Jun  4 22:30:24.222: //CMM/INFO:increase_gcid_ref_count 99685 0
Jun  4 22:30:24.222: //CMM/INFO:find_gcidinfo_node
Jun  4 22:30:24.222: //CMM/INFO:      target_node 0
Jun  4 22:30:24.222: //CMM/INFO:      Gcidinfo node Search FAILED
Jun  4 22:30:24.222: //CMM/INFO:create_gcidinfo_node
Jun  4 22:30:24.222: //CMM/INFO:      target_node 6544A9CC
Jun  4 22:30:24.222: //CMM/INFO:      - gcid 05591A85-122211DC-8645A1CA-4B604A7A
Jun  4 22:30:24.222: //CMM/INFO:      count = 1
Jun  4 22:30:24.222: //CMM/INFO:insert_ssptrs_to_gcid for line_info 66AF3720 (dn 4016),
GCID 05591A85-122211DC-8645A1CA-4B604A7A
Jun  4 22:30:24.222:      ss_ptr list :-
Jun  4 22:30:24.222:      ss_ptr list :-
Jun  4 22:30:24.222: //CMM/INFO:
Jun  4 22:30:24.222: //CMM/INFO:
Jun  4 22:30:24.222: //CMM/INFO:cmm_notify_trigger() 1, callID 99685, 5114016, 16,
1695547392
Jun  4 22:30:24.222: //CMM/INFO:      target_node 66AF3714
Jun  4 22:30:24.222: //CMM/INFO:      - dn 4016

Jun  4 22:30:24.222: //CMM/INFO: CallEntry 709C3FB8
Jun  4 22:30:24.222: //CMM/INFO: dstCallID -1
Jun  4 22:30:24.222: //CMM/INFO: line_info 66AF3720, dn 4016
Jun  4 22:30:24.222: //CMM/INFO:      * cmm_crs_proc_tr_call_orig
Jun  4 22:30:24.222: //CMM/INFO:      orig --> callID 99685, line_info 66AF3720,
call_inst 655AF384, gcid 05591A85-122211DC-8645A1CA-4B604A7A
Jun  4 22:30:24.222: //CMM/INFO:is_sccp_endpoint DN 4016
Jun  4 22:30:24.222: //CMM/INFO:
Jun  4 22:30:24.222:      sccp endpoint TRUE
Jun  4 22:30:24.222: //CMM/INFO:find_gcidinfo_node
Jun  4 22:30:24.222: //CMM/INFO:      target_node 6544A9CC
Jun  4 22:30:24.222: //CMM/INFO:      - gcid 05591A85-122211DC-8645A1CA-4B604A7A
Jun  4 22:30:24.222: //CMM/INFO:cmm_send_dialog_notify sub_info 0
Jun  4 22:30:24.222:      ss_ptr list :-
Jun  4 22:30:24.222: //CMM/INFO:      <== DIALOG MGR ==>
Jun  4 22:30:24.222: //CMM/INFO:      :: CMM_EV_CALL_CONN_ORIGINATED
Jun  4 22:30:24.222: //CMM/INFO:      - Gcid
05591A85-122211DC-8645A1CA-4B604A7A
Jun  4 22:30:24.222: //CMM/INFO:      - Calling 4016
Jun  4 22:30:24.222: //CMM/INFO:      - Called
Jun  4 22:30:24.222: //CMM/INFO:      - ConnAddr 4016
Jun  4 22:30:24.222: //CMM/INFO:      - Type 0
Jun  4 22:30:24.222: //CMM/INFO:      - parentGcid
00000000-00000000-00000000-00000000
Jun  4 22:30:24.222: //CMM/INFO:find_gcidinfo_node

```

debug callmonitor

```

Jun  4 22:30:24.222: //CMM/INFO:      target_node 6544A9CC
Jun  4 22:30:24.222: //CMM/INFO:      - gcid 05591A85-122211DC-8645A1CA-4B604A7A
Jun  4 22:30:24.222: //CMM/DETAIL: type: CMM_EV_CALL_CONN_ORIGINATED, filter analyzing....
[4016, , 4016]
Jun  4 22:30:24.222: //CMM/INFO:find_gcidinfo_node
Jun  4 22:30:24.222: //CMM/INFO:      target_node 6544A9CC
Jun  4 22:30:24.222: //CMM/INFO:      - gcid 05591A85-122211DC-8645A1CA-4B604A7A
Jun  4 22:30:24.222: //CMM/DETAIL:gcid is not part of conference. [4016, , 4016] checking
originateFilter...
Jun  4 22:30:24.222: //CMM/DETAIL:originateFilter[callid=99685, pdn=16, pchan=1] is not
set. [4016, , 4016] is not filtered
Jun  4 22:30:24.222: //CMM/INFO:find_gcidinfo_node
Jun  4 22:30:24.222: //CMM/INFO:      target_node 6544A9CC
Jun  4 22:30:24.222: //CMM/INFO:      - gcid 05591A85-122211DC-8645A1CA-4B604A7A
Jun  4 22:30:24.222: //CMM/INFO:cmm_send_dialog_notify sub_info 0
Jun  4 22:30:24.222:      ss_ptr list :-
Jun  4 22:30:24.222: //CMM/INFO:      <== DIALOG MGR ==>
Jun  4 22:30:24.222: //CMM/INFO:      :: CMM_EV_CALL_CONN_ACTIVE
Jun  4 22:30:24.222: //CMM/INFO:      - Gcid
05591A85-122211DC-8645A1CA-4B604A7A
Jun  4 22:30:24.222: //CMM/INFO:      - Calling 4016
Jun  4 22:30:24.222: //CMM/INFO:      - Called
Jun  4 22:30:24.222: //CMM/INFO:      - ConnAddr 4016
Jun  4 22:30:24.222: //CMM/INFO:      - LastRedirectAddr
Jun  4 22:30:24.222: //CMM/INFO:      - Type 0
Jun  4 22:30:24.222: //CMM/INFO:      - parentGcid
00000000-00000000-00000000-00000000
Jun  4 22:30:24.222: //CMM/INFO:find_gcidinfo_node
Jun  4 22:30:24.222: //CMM/INFO:      target_node 6544A9CC
Jun  4 22:30:24.222: //CMM/INFO:      - gcid 05591A85-122211DC-8645A1CA-4B604A7A
Jun  4 22:30:24.222: //CMM/DETAIL: type: CMM_EV_CALL_CONN_ACTIVE, filter analyzing....
[4016, , 4016]
Jun  4 22:30:24.222: //CMM/DETAIL:called number is not specified. [4016, , 4016]
Jun  4 22:30:24.222: //CMM/DETAIL:originateFilter[callid=99685, pdn=16, pchan=1] is not
set, [4016, , 4016] is not filtered
Jun  4 22:30:25.670: //CMM/INFO:
Jun  4 22:30:25.670: //CMM/INFO:
Jun  4 22:30:25.670: //CMM/INFO:cmm_notify_trigger() 14, callID 99686, 8101, 1902058375, 0
Jun  4 22:30:25.670: //CMM/INFO:      target_node 65DB15E4
Jun  4 22:30:25.670: //CMM/INFO:      - dn 8101
.
.
.

```

Related Commands

Command	Description
callmonitor	Enable call monitoring messaging functionality on a SIP endpoint in a VoIP network.
gcid	Enable Global Call ID (Gcid) for every call on an outbound leg of a VoIP dial peer for a SIP endpoint.

debug capf-server

To collect debug information about the CAPF server, use the **debug capf-server** command in privileged EXEC mode. To disable collection of debug information, use the **no** form of this command.

debug capf-server {all | error | events | messages}

no debug capf-server

Syntax Description

all	Collect all CAPF information available.
error	Collect only information about CAPF errors.
events	Collect only information about CAPF status events.
messages	Collect only CAPF system messages.

Command Default

Collection of CAPF debug information is disabled.

Command Modes

Privileged EXEC

Command History

Cisco IOS Release	Modification
12.4(4)XC	This command was introduced.
12.4(9)T	This command was integrated into Cisco IOS Release 12.4(9)T.

Usage Guidelines

This command is used with Cisco Unified CallManager Express phone authentication.

Examples

The following example shows debug messages for the CAPF server.

```
Router# debug capf-server all

001891: .Jul 21 18:17:07.014: %IPPHONE-6-UNREGISTER_NORMAL: ephone-1:SEP000E325C9A43
IP:10.10.10.194 So
cket:3 DeviceType:Phone has unregistered normally.
001892: .Jul 21 18:17:20.495: New Connection from phone, socket 1
001893: .Jul 21 18:17:20.495: Created New Handshake Process
001894: .Jul 21 18:17:20.499: SSL Handshake Error -6983
001895: .Jul 21 18:17:21.499: SSL Handshake Error -6983
001896: .Jul 21 18:17:22.555: SSL Handshake Successful
001897: .Jul 21 18:17:22.555: ephone_capf_send_auth_req:
001898: .Jul 21 18:17:22.555: ephone_capf_ssl_write: 12 bytes
001899: .Jul 21 18:17:22.711: ephone_capf_ssl_read: Read 35 bytes
001900: .Jul 21 18:17:22.711: ephone_capf_handle_phone_msg: msgtype 2
001901: .Jul 21 18:17:22.711: ephone_capf_process_auth_res_msg: SEP000E325C9A43 AuthMode 2
001902: .Jul 21 18:17:22.711: ephone_capf_send_delete_cert_req_msg: SEP000E325C9A43
001903: .Jul 21 18:17:22.711: ephone_capf_ssl_write: 8 bytes
001904: .Jul 21 18:17:23.891: ephone_capf_ssl_read: Read 12 bytes
001905: .Jul 21 18:17:23.891: ephone_capf_handle_phone_msg: msgtype 14
001906: .Jul 21 18:17:23.891: certificate delete successful for SEP000E325C9A43
```

```
001907: .Jul 21 18:17:24.695: ephone_capf_release_session: SEP000E325C9A43
001908: .Jul 21 18:17:24.695: ephone_capf_send_end_session_msg: SEP000E325C9A43
001909: .Jul 21 18:17:24.695: ephone_capf_ssl_write: 12 bytes
001910: .Jul 21 18:17:25.095: %IPPHONE-6-REG_ALARM: 22: Name=SEP000E325C9A43 Load=7.2(2.0)
Last=Reset
t-Reset
001911: .Jul 21 18:17:25.099: %IPPHONE-6-REGISTER: ephone-1:SEP000E325C9A43
IP:10.10.10.194 Socket:2 DeviceType:Phone has registered.

001912: .Jul 21 18:18:05.171: %IPPHONE-6-UNREGISTER_NORMAL: ephone-1:SEP000E325C9A43
IP:1.1.1.127 Socket:2 DeviceType:Phone has unregistered normally.
001913: .Jul 21 18:18:18.288: New Connection from phone, socket 1
001914: .Jul 21 18:18:18.288: Created New Handshake Process
001915: .Jul 21 18:18:18.292: SSL Handshake Error -6983
001916: .Jul 21 18:18:19.292: SSL Handshake Error -6983
001917: .Jul 21 18:18:20.348: SSL Handshake Successful
001918: .Jul 21 18:18:20.348: ephone_capf_send_auth_req:
001919: .Jul 21 18:18:20.348: ephone_capf_ssl_write: 12 bytes^Z

001920: .Jul 21 18:18:20.492: ephone_capf_ssl_read: Read 35 bytes
001921: .Jul 21 18:18:20.492: ephone_capf_handle_phone_msg: msgtype 2
001922: .Jul 21 18:18:20.492: ephone_capf_process_auth_res_msg: SEP000E325C9A43 AuthMode 2
001923: .Jul 21 18:18:20.492: ephone_capf_send_PhKeyGenReq_msg: SEP000E325C9A43 KeySize
1024
001924: .Jul 21 18:18:20.492: ephone_capf_ssl_write: 13 bytes
001925: .Jul 21 18:18:20.540: ephone_capf_ssl_read: Read 8 bytes
001926: .Jul 21 18:18:20.540: ephone_capf_handle_phone_msg: msgtype 17
001927: .Jul 21 18:18:20.540: ephone_capf_process_req_in_progress: SEP000E325C9A43 delay
0sh
001928: .Jul 21 18:18:21.924: %SYS-5-CONFIG_I: Configured from console by user1 on console
```

debug cch323 video

To provide debugging output for video components within the H.323 subsystem, use the **debug cch323 video** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug cch323 video

no debug cch323 video

Syntax Description This command has no arguments or keywords.

Command Modes Privileged EXEC

Command History	Cisco IOS Release	Modification
	12.4(4)XC	This command was introduced.
	12.4(9)T	This command was integrated into Cisco IOS Release 12.4(9)T.

Usage Guidelines Use this command to enable a debugging trace for the video component in an H.323 network.

Examples

Originating Gateway Example

The following is sample output of the debugging log for an originating Cisco Unified CallManager Express (Cisco Unified CME) gateway after the **debug cch323 video** command was enabled:

```
Router# show log

Syslog logging: enabled (11 messages dropped, 487 messages rate-limited,
                  0 flushes, 0 overruns, xml disabled, filtering disabled)
  Console logging: disabled
  Monitor logging: level debugging, 0 messages logged, xml disabled,
                  filtering disabled
  Buffer logging: level debugging, 1144 messages logged, xml disabled,
                  filtering disabled
  Logging Exception size (4096 bytes)
  Count and timestamp logging messages: disabled
  Trap logging: level informational, 1084 message lines logged

Log Buffer (6000000 bytes):

Jun 13 09:19:42.006: //103030/C7838B198002/H323/cch323_get_peer_info: Entry
Jun 13 09:19:42.006: //103030/C7838B198002/H323/cch323_get_peer_info: Have peer
Jun 13 09:19:42.006: //103030/C7838B198002/H323/cch323_set_pref_codec_list: First
preferred codec(bytes)=16(20)
Jun 13 09:19:42.006: //103030/C7838B198002/H323/cch323_get_peer_info: Flow Mode set to
FLOW_THROUGH
Jun 13 09:19:42.006: //103030/C7838B198002/H323/cch323_get_caps_chn_info: No peer leg
setup params
Jun 13 09:19:42.006: //103030/C7838B198002/H323/cch323_get_caps_chn_info: Setting
CCH323_SS_NTFY_VIDEO_INFO
```

```

Jun 13 09:19:42.006: //103030/C7838B198002/H323/cch323_set_h323_control_options_outgoing:
h245 sm mode = 8463
Jun 13 09:19:42.006: //103030/C7838B198002/H323/cch323_set_h323_control_options_outgoing:
h323_ctl=0x20
Jun 13 09:19:42.010: //103030/C7838B198002/H323/cch323_rotary_validate: No peer_ccb
available

```

Terminating Gateway Example

The following is sample output of the debugging log for a terminating Cisco Unified Survivable Remote Site Telephony (Cisco Unified SRST) gateway after the **debug cch323 video** command was enabled:

Router# **show log**

```

Syslog logging: enabled (11 messages dropped, 466 messages rate-limited,
0 flushes, 0 overruns, xml disabled, filtering disabled)
  Console logging: disabled
  Monitor logging: level debugging, 0 messages logged, xml disabled,
filtering disabled
  Buffer logging: level debugging, 829 messages logged, xml disabled,
filtering disabled
  Logging Exception size (4096 bytes)
  Count and timestamp logging messages: disabled
  Trap logging: level informational, 771 message lines logged

```

Log Buffer (200000 bytes):

```

Jun 13 09:19:42.011: //103034/C7838B198002/H323/setup_ind: Receive bearer cap infoXRate
24, rateMult 12
Jun 13 09:19:42.011: //103034/C7838B198002/H323/cch323_set_h245_state_mc_mode_incoming:
h245 state m/c mode=0x10F, h323_ctl=0x2F
Jun 13 09:19:42.015: //-1/xxxxxxxxxxxx/H323/cch245_event_handler: callID=103034
Jun 13 09:19:42.019: //-1/xxxxxxxxxxxx/H323/cch245_event_handler: Event
CC_EV_H245_SET_MODE: data ptr=0x465D5760
Jun 13 09:19:42.019: //-1/xxxxxxxxxxxx/H323/cch323_set_mode: callID=103034, flow Mode=1
spi_mode=0x6
Jun 13 09:19:42.019: //103034/C7838B198002/H323/cch323_do_call_proceeding: set_mode NOT
called yet...saved deferred CALL_PROC
Jun 13 09:19:42.019: //103034/C7838B198002/H323/cch323_h245_connection_sm: state=0,
event=0, ccb=4461B518, listen state=0
Jun 13 09:19:42.019: //103034/C7838B198002/H323/cch323_process_set_mode: Setting inbound
leg mode flags to 0x10F, flow-mode to FLOW_THROUGH
Jun 13 09:19:42.019: //103034/C7838B198002/H323/cch323_process_set_mode: Sending deferred
CALL_PROC
Jun 13 09:19:42.019: //103034/C7838B198002/H323/cch323_do_call_proceeding: set_mode called
so we can proceed with CALLPROC
Jun 13 09:19:42.027: //103034/C7838B198002/H323/cch323_h245_connection_sm: state=1,
event=2, ccb=4461B518, listen state=1
Jun 13 09:19:42.027: //103034/C7838B198002/H323/cch323_send_cap_request: Setting mode to
VIDEO MODE
Jun 13 09:19:42.031: //103034/C7838B198002/H323/cch323_h245_cap_ind: Masks au=0xC data=0x2
uinp=0x32

```

Related Commands

Command	Description
debug ephone video	Sets video debugging for the Cisco Unified IP phone.
show call active video	Displays call information for SCCP video calls in progress.

Command (continued)	Description
show call history video	Displays call history information for SCCP video calls.
show debugging	Displays information about the types of debugging that are enabled for your router.

debug credentials

To set debugging on the credentials service that runs between the Cisco Unified CME CTL provider and CTL client or between the Cisco Unified SRST router and Cisco Unified CallManager, use the **debug credentials** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug credentials

no debug credentials

Syntax Description This command has no arguments or keywords.

Command Modes Privileged EXEC

Command History	Cisco IOS Release	Modification
	12.3(14)T	This command was introduced for Cisco Unified SRST.
	12.4(4)XC	This command was introduced for Cisco Unified CME.
	12.4(9)T	This command was integrated into Cisco IOS Release 12.4(9)T for Cisco Unified CME.

Usage Guidelines

Cisco Unified CME

Use this command with Cisco Unified CME phone authentication to monitor a CTL provider as it provides credentials to the CTL client.

Cisco Unified SRST

Use this command to monitor Cisco Unified CallManager while it requests certificates from the Cisco Unified SRST router. It sets debugging on the credentials service that runs between the SRST router and Cisco Unified CallManager

Examples

Cisco Unified CME

The following sample output displays the CTL provider establishing a TLS session with the CTL client and providing all the relevant credentials for the services that are running on this router to the CTL client.

```
Router# debug credentials
```

```
Credentials server debugging is enabled
```

```
May 25 12:08:17.944: Credentials service: Start TLS Handshake 1 10.5.43.174 4374
```

```
May 25 12:08:17.948: Credentials service: TLS Handshake returns OPSSLReadWouldBlockErr
```

```
May 25 12:08:18.948: Credentials service: TLS Handshake returns OPSSLReadWouldBlockErr
```

```
May 25 12:08:19.948: Credentials service: TLS Handshake returns OPSSLReadWouldBlockErr
```

```
May 25 12:08:20.964: Credentials service: TLS Handshake completes
```

Cisco Unified SRST

The following is sample output showing the credentials service that runs between the Cisco Unified SRST router and Cisco Unified CallManager. The credentials service provides Cisco Unified CallManager with the certificate from the SRST router.

```
Router# debug credentials
```

```
Credentials server debugging is enabled
```

```
Router#
```

```
May 25 12:08:17.944: Credentials service: Start TLS Handshake 1 10.5.43.174 4374
```

```
May 25 12:08:17.948: Credentials service: TLS Handshake returns OPSSLReadWouldBlockErr
```

```
May 25 12:08:18.948: Credentials service: TLS Handshake returns OPSSLReadWouldBlockErr
```

```
May 25 12:08:19.948: Credentials service: TLS Handshake returns OPSSLReadWouldBlockErr
```

```
May 25 12:08:20.964: Credentials service: TLS Handshake completes
```

Table 4 describes the significant fields shown in the display.

Table 4 *debug credentials Field Descriptions*

Field	Description
Start TLS Handshake 1 10.5.43.174 4374	Indicates the beginning of the TLS handshake between the secure Cisco Unified SRST router and Cisco Unified CallManager. In this example, 1 indicates the socket, 10.5.43.174 is the IP address, and 4374 is the port of Cisco Unified CallManager.
TLS Handshake returns OPSSLReadWouldBlockErr	Indicates that the handshake is in process.
TLS Handshake completes	Indicates that the TLS handshake has finished and that the Cisco Unified CallManager has received the secure Cisco Unified SRST device certificate.

Related Commands

Command	Description
credentials	Enters credentials configuration mode to configure a Cisco Unified CME CTL provider certificate or a Cisco Unified SRST router certificate.
ctl-service admin	Specifies a user name and password to authenticate the CTL client during the CTL protocol.
ip source-address (credentials)	Enables the Cisco Unified CME or SRST router to receive messages through the specified IP address and port.
show credentials	Displays the credentials settings on a Cisco Unified CME or SRST router.
show debugging	Displays information about the types of debugging that are enabled for your router.
trustpoint (credentials)	Specifies the name of the trustpoint to be associated with a Cisco Unified CME CTL provider certificate or with a Cisco Unified SRST router certificate.

debug cti

To enable debugging on the CTI interface in Cisco Unified CME, use the **debug cti** command in privileged EXEC mode. To disable debugging, use the **no** form of this command.

debug cti {all | callcontrol | core | dmgr | lm | protoif | session | xml}

no debug cti {all | callcontrol | core | dmgr | lm | protoif | session | xml}

Syntax Description

all	All CTI debugging traces.
callcontrol	CTI call control debugging traces.
core	Basic call debugging traces.
dmgr	CTI device manager debugging traces.
lm	CTI line monitoring debugging traces.
protoif	CTI protocol interface debugging traces.
session	CTI session debugging traces.
xml	CTI xml debugging traces.

Command Default

Debugging on the CTI interface is disabled.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
15.0(1)XA	This command was introduced.

Usage Guidelines

This command sets debugging for the CTI interface in Cisco Unified CME.

Examples

The following partial output from the **debug cti core** command shows the events from the time a call is placed to when the connection is established:

```
Router# debug cti core
Core CTI debug flags are on
.
.
.
Router#
Jun 17 23:12:09.885: //CTI/PI:cti_frontend_proc [BB5C]: received CC Event [19]:
CC_EV_CALL_INFO
Jun 17 23:12:09.885: //CTI/PI:pi_process_service_event event 19
Jun 17 23:12:09.885: //CTI/PI: got CC_EV_CALL_INFO callID 47964
Jun 17 23:12:09.885: //CTI/PI:pi_parse_service event 0
.
.
.
```

```
Jun 17 23:12:09.889: //CTI/CC:Fsm_Idle_MakeCall calling 201, called 204
Jun 17 23:12:09.889: //CTI/DMGR:
Jun 17 23:12:09.889: MakeCall event sent to Device Manager.callID 47964, Mac:0019E83B211D,
CallingNum:201, CalledNum:204
Jun 17 23:12:09.889: //CTI/DMGR:
Jun 17 23:12:09.889: MakeCall event sent to skinny server.Mac:0019E83B211D,
CallingNum:201, CalledNum:204
Jun 17 23:12:09.893: //CTI/CM:-- trigger 1, callID 59291, dn 201, reason 0, result 0
Jun 17 23:12:09.893: //CTI/CM: line_info 87674E4C, dn 201
Jun 17 23:12:09.893: //CTI/CM: * cmm_crs_proc_tr_call_orig
Jun 17 23:12:09.893: //CTI/CM:increase_gcid_ref_count 59291 0
Jun 17 23:12:09.893: //CTI/CM: Gcidinfo node Search FAILED
Jun 17 23:12:09.893: //CTI/CM:create_gcidinfo_node 59291
Jun 17 23:12:09.893: //CTI/CM: orig --> callID 59291, line_info 87674E4C, call_inst
88B0B070, gcid 1E2E3483-5ACB11DE-BA9EF925-DF2AFB55
Jun 17 23:12:09.893: === EVENT EV_ORIGINATED
Jun 17 23:12:09.893: 201 --> . cause normal
.
.
.
Jun 17 23:12:19.217: //CTI/PI:pi_process_service_event event 20
Jun 17 23:12:19.217: //CTI/PI: got CC_EV_CALL_INFO_ACK callID 47964
Jun 17 23:12:19.217: //CTI/SM:sm_handle_cc_service event 77
Jun 17 23:12:19.217: //CTI/SM:sm_find_scb_node_by_context context_id 47964
Jun 17 23:12:19.217: //CTI/SM: to return 86B88298
Jun 17 23:12:19.217: //CTI/SM: got CTI_EV_ACK, callID 47964
Jun 17 23:12:19.221: //CTI/PI:cti_frontend_proc [E750]: received CC Event [20]:
CC_EV_CALL_LOOPBACK_DONE
Jun 17 23:12:19.221: //CTI/PI:pi_process_service_event event 20
Jun 17 23:12:19.221: //CTI/PI: got CC_EV_CALL_INFO_ACK callID 59216
Jun 17 23:12:19.221: //CTI/SM:sm_handle_cc_service event 77
Jun 17 23:12:19.221: //CTI/SM:sm_find_scb_node_by_context context_id 59216
Jun 17 23:12:19.221: //CTI/SM: to return 87396644
Jun 17 23:12:19.221: //CTI/SM: got CTI_EV_ACK, callID 59216
UC520#
```

Related Commands	Command	Description
	show debugging	Displays information about the types of debugging that are enabled for your router.

debug ctl-client

To collect debug information about the CTL client, use the **debug ctl-client** command in privileged EXEC configuration mode. To disable collection of debug information, use the **no** form of this command.

debug ctl-client

no debug ctl-client

Syntax Description This command has no arguments or keywords.

Command Default Collection of CTL client debug information is disabled.

Command Modes Privileged EXEC

Command History	Cisco IOS Release	Modification
	12.4(4)XC	This command was introduced.
	12.4(9)T	This command was integrated into Cisco IOS Release 12.4(9)T.

Usage Guidelines This command is used with Cisco Unified CME phone authentication.

Examples The following example shows debug messages for the CTL client:

```
Router# debug ctl-client

001954: .Jul 21 18:23:02.136: ctl_client_create_ctlfile:
001955: .Jul 21 18:23:02.272: create_ctl_record: Function 0 Trustpoint cisco1
001956: .Jul 21 18:23:02.276: create_ctl_record: record added for function 0
001957: .Jul 21 18:23:02.276: create_ctl_record: Function 0 Trustpoint sast2
001958: .Jul 21 18:23:02.280: create_ctl_record: record added for function 0
001959: .Jul 21 18:23:02.280: create_ctl_record: Function 1 Trustpoint cisco1
001960: .Jul 21 18:23:02.284: create_ctl_record: record added for function 1
001961: .Jul 21 18:23:02.284: create_ctl_record: Function 3 Trustpoint cisco1
001962: .Jul 21 18:23:02.288: create_ctl_record: record added for function 3
001963: .Jul 21 18:23:02.288: create_ctl_record: Function 4 Trustpoint cisco1
001964: .Jul 21 18:23:02.292: create_ctl_record: record added for function 4
001965: .Jul 21 18:23:02.424: ctl_client_create_ctlfile: Signature length 128
001966: .Jul 21 18:23:02.640: CTL File Created Successfully
```

debug ephone alarm

To set SkinnyStation alarm messages debugging for the Cisco IP phone, use the **debug ephone alarm** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ephone alarm [**mac-address** *mac-address*]

no debug ephone alarm [**mac-address** *mac-address*]

Syntax Description

mac-address	(Optional) Defines the MAC address of the Cisco IP phone.
<i>mac-address</i>	(Optional) Specifies the MAC address of the Cisco IP phone.

Defaults

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(2)XT	This command was introduced on the following platforms: Cisco 1750, Cisco 1751, Cisco 2600 series and Cisco 3600 series multiservice routers; and Cisco IAD2420 series Integrated Access Devices (IADs).
12.2(8)T	This command was implemented on the Cisco 3725 and Cisco 3745 routers.
12.2(8)T1	This command was implemented on the Cisco 2600-XM and Cisco 2691 routers.
12.2(11)T	This command was implemented on the Cisco 1760 routers.

Usage Guidelines

The **debug ephone alarm** command shows all the SkinnyStation alarm messages sent by the Cisco IP phone. Under normal circumstances, this message is sent by the Cisco IP phone just before it registers, and the message has the severity level for the alarm set to “Informational” and contains the reason for the phone reboot or re-register. This type of message is entirely benign and does not indicate an error condition.

If the **mac-address** keyword is not used, the **debug ephone alarm** command debugs all Cisco IP phones that are registered to the router. You can remove debugging for the Cisco IP phones that you do not want to debug by using the **mac-address** keyword with the **no** form of this command.

You can enable or disable debugging on any number of Cisco IP phones. To see the Cisco IP phones that have debugging enabled, enter the **show ephone** command and look at the debug field in the output. When debugging is enabled for a Cisco IP phone, the debug output is displayed for the directory numbers associated with the Cisco IP phone.

Examples

The following example shows a SkinnyStation alarm message that is sent before the Cisco IP phone registers:

```
Router# debug ephone alarm

phone keypad reset
CM-closed-TCP
CM-bad-state
```

Related Commands

Command	Description
debug ephone detail	Sets detail debugging for the Cisco IP phone.
debug ephone error	Sets error debugging for the Cisco IP phone.
debug ephone keepalive	Sets keepalive debugging for the Cisco IP phone.
debug ephone loopback	Sets MWI debugging for the Cisco IP phone.
debug ephone pak	Provides voice packet level debugging and prints the contents of one voice packet in every 1024 voice packets.
debug ephone raw	Provides raw low-level protocol debugging display for all SCCP messages.
debug ephone register	Sets registration debugging for the Cisco IP phone.
debug ephone state	Sets state debugging for the Cisco IP phone.
debug ephone statistics	Sets statistics debugging for the Cisco IP phone.
show debugging	Displays information about the types of debugging that are enabled for your router.

debug ephone blf

To display debugging information for Busy Lamp Field (BLF) presence features, use the **debug ephone blf** command in privileged EXEC mode. To disable debugging, use the **no** form of this command.

debug ephone blf [*mac-address mac-address*]

no debug ephone blf [*mac-address mac-address*]

Syntax Description	mac-address mac-address (Optional) Specifies the MAC address of a specific IP phone.
---------------------------	---------------------------------------------------------------------------------------------

Command Modes	Privileged EXEC
----------------------	-----------------

Command History	Release	Modification
	12.4(11)XJ	This command was introduced.
	12.4(15)T	This command was integrated into Cisco IOS Release 12.4(15)T.

Usage Guidelines	Use this command for troubleshooting BLF speed-dial and BLF call-list features for phones in a presence service.
-------------------------	------------------------------------------------------------------------------------------------------------------

Examples	The following is sample output from the debug ephone blf command.
-----------------	--------------------------------------------------------------------------

```
Router# debug ephone blf

EPHONE BLF debugging is enabled

*Sep  4 07:18:26.307: skinny_asnl_callback: subID 16 type 4
*Sep  4 07:18:26.307: ASNL_RESP_NOTIFY_INDICATION
*Sep  4 07:18:26.307: ephone-1[1]:ASNL notify indication message, feature index 4, subID
[16]
*Sep  4 07:18:26.307: ephone-1[1]:line status 6, subID [16]
*Sep  4 07:18:26.307: ephone-1[1]:StationFeatureStatV2Message sent, status 2
*Sep  4 07:18:26.307: skinny_asnl_callback: subID 23 type 4
*Sep  4 07:18:26.307: ASNL_RESP_NOTIFY_INDICATION
*Sep  4 07:18:26.307: ephone-2[2]:ASNL notify indication message, feature index 2, subID
[23]
*Sep  4 07:18:26.311: ephone-2[2]:line status 6, subID [23]
*Sep  4 07:18:26.311: ephone-2[2]:StationFeatureStatV2Message sent, status 2
*Sep  4 07:18:28.951: skinny_asnl_callback: subID 16 type 4
*Sep  4 07:18:28.951: ASNL_RESP_NOTIFY_INDICATION
*Sep  4 07:18:28.951: ephone-1[1]:ASNL notify indication message, feature index 4, subID
[16]
*Sep  4 07:18:28.951: ephone-1[1]:line status 1, subID [16]
*Sep  4 07:18:28.951: ephone-1[1]:StationFeatureStatV2Message sent, status 1
*Sep  4 07:18:28.951: skinny_asnl_callback: subID 23 type 4
*Sep  4 07:18:28.951: ASNL_RESP_NOTIFY_INDICATION
*Sep  4 07:18:28.951: ephone-2[2]:ASNL notify indication message, feature index 2, subID
[23]
*Sep  4 07:18:28.951: ephone-2[2]:line status 1, subID [23]
```

```
*Sep  4 07:18:28.951: ephone-2[2]:StationFeatureStatV2Message sent, status 1
```

Related Commands

Command	Description
blf-speed-dial	Enables BLF monitoring for a speed-dial number on a phone registered to Cisco Unified CME.
presence call-list	Enables BLF monitoring for call lists and directories on phones registered to a Cisco Unified CME router.
show presence global	Displays configuration information about the presence service.
show presence subscription	Displays information about active presence subscriptions.

debug ephone ccm-compatible

To display Cisco CallManager notification updates for calls between Cisco CallManager and Cisco CallManager Express, use the **debug ephone ccm-compatible** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ephone ccm-compatible [*mac-address mac-address*]

no debug ephone ccm-compatible [*mac-address mac-address*]

Syntax Description

mac-address mac-address (Optional) Specifies the MAC address of a Cisco IP phone for debugging.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.3(7)T	This command was introduced.

Usage Guidelines

This command displays call flow notification information for all calls between Cisco CallManager and Cisco CallManager Express, but it is most useful for filtering out specific information for transfer and forward cases. For basic call information, use the **debug ephone state** command.

If you do not specify the **mac-address** keyword, the **debug ephone ccm-compatible** command debugs all Cisco IP phones that are registered to the router. You can remove debugging for the Cisco IP phones that you do not want to debug by using the **no** form of this command with the **mac-address** keyword.

Debugging can be enabled or disabled on any number of Cisco IP phones. Cisco IP phones that have debugging enabled are listed in the debug field of the **show ephone** command output. When debugging is enabled for a Cisco IP phone, debug output is displayed for all phone extensions (virtual voice ports) associated with that phone.

Examples

The following sample output displays call flow notifications between Cisco CallManager and Cisco CallManager Express:

```
Router# debug ephone ccm-compatible

*May 1 04:30:02.650:ephone-2[2]:DtAlertingTone/DtHoldTone - mediaActive reset during
CONNECT
*May 1 04:30:02.654:ephone-2[2]:DtHoldTone - force media STOP state
*May 1 04:30:02.654://93/xxxxxxxxxxxx/CCAPI/ccCallNotify:(callID=0x5D,nData->
bitmask=0x00000007)
*May 1 04:30:02.654://93/xxxxxxxxxxxx/VTSP:(50/0/3):-1:0:5/vtsp_process_event:
vtsp:[50/0/3 (93), S_CONNECT, E_CC_SERVICE_MSG]
*May 1 04:30:02.654://93/xxxxxxxxxxxx/VTSP:(50/0/3):-1:0:5/act_service_msg_dow
n:.
*May 1 04:30:02.658:dn_callerid_update DN 3 number= 12009 name= CCM7960 in state
CONNECTED
*May 1 04:30:02.658:dn_callerid_update (incoming) DN 3 info updated to
*May 1 04:30:02.658:calling= 12009 called= 13003 origCalled=
```

```

*May 1 04:30:02.658:callingName= CCM7960, calledName= , redirectedTo =
*May 1 04:30:02.658:ephone-2[2][SEP003094C2999A]:refreshDisplayLine for line 1
  DN 3 chan 1
*May 1 04:30:03.318:ephone-2[2]:DisplayCallInfo incoming call
*May 1 04:30:03.318:ephone-2[2]:Call Info DN 3 line 1 ref 24 called 13003 calling 12009
origcalled 13003 calltype 1
*May 1 04:30:03.318:ephone-2[2]:Original Called Name UUT4PH3
*May 1 04:30:03.318:ephone-2[2]:CCM7960 calling
*May 1 04:30:03.318:ephone-2[2]:UUT4PH3

```

Related Commands

Command	Description
debug ephone state	Displays call state information.
show debugging	Displays information about the types of debugging that are enabled for your router.
show ephone	Displays information about registered Cisco IP phones.

debug ephone detail

To set detail debugging for the Cisco IP phone, use the **debug ephone detail** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ephone detail [**mac-address** *mac-address*]

no debug ephone detail [**mac-address** *mac-address*]

Syntax Description

mac-address	(Optional) Defines the MAC address of the Cisco IP phone.
<i>mac-address</i>	(Optional) Specifies the MAC address of the Cisco IP phone.

Defaults

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(5)YD	This command was introduced on the following platforms: Cisco 2600 series and Cisco 3600 series multiservice routers, and Cisco IAD2420 series Integrated Access Devices (IADs).
12.2(2)XT	This command was implemented on the Cisco 1750 and Cisco 1751 multiservice routers.
12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T and implemented on the Cisco 3725 and Cisco 3745 routers.
12.2(8)T1	This command was implemented on the Cisco 2600-XM and Cisco 2691 routers.
12.2(11)T	This command was implemented on the Cisco 1760 routers.

Usage Guidelines

The **debug ephone detail** command includes the error and state levels.

If the **mac-address** keyword is not used, the **debug ephone detail** command debugs all Cisco IP phones that are registered to the router. You can remove debugging for the Cisco IP phones that you do not want to debug by using the **mac-address** keyword with the **no** form of this command.

You can enable or disable debugging on any number of Cisco IP phones. To see the Cisco IP phones that have debugging enabled, enter the **show ephone** command and look at the debug field in the output. When debugging is enabled for a Cisco IP phone, the debug output is displayed for the directory numbers associated with the Cisco IP phone.

Examples

The following is sample output of detail debugging of the Cisco IP phone with MAC address 0030.94c3.8724. The sample is an excerpt of some of the activities that takes place during call setup, connected state, active call, and the call being disconnected.

```
Router# debug ephone detail mac-address 0030.94c3.8724
```

```
Ephone detail debugging is enabled
1d04h: ephone-1[1]:OFFHOOK
.
.
1d04h: Skinny Call State change for DN 1 SIEZE
.
.
1d04h: ephone-1[1]:SetCallState line 1 DN 1 TsOffHook
.
.
1d04h: ephone-1[1]:SetLineLamp 1 to ON
.
.
1d04h: ephone-1[1]:KeypadButtonMessage 5
.
.
1d04h: ephone-1[1]:KeypadButtonMessage 0
.
.
1d04h: ephone-1[1]:KeypadButtonMessage 0
.
.
1d04h: ephone-1[1]:KeypadButtonMessage 2
.
.
1d04h: ephone-1[1]:Store ReDial digit: 5002
.
SkinnyTryCall to 5002 instance 1
.
.
1d04h: ephone-1[1]:Store ReDial digit: 5002
1d04h: ephone-1[1]:
SkinnyTryCall to 5002 instance 1
.
.
1d04h: Skinny Call State change for DN 1 ALERTING
.
.
1d04h: ephone-1[1]:SetCallState line 1 DN 1 TsRingOut
.
.
1d04h: ephone-1[1]:SetLineLamp 1 to ON
1d04h: SetCallInfo calling dn 1 dn 1
calling [5001] called [5002]
.
.
1d04h: ephone-1[1]: Jane calling
1d04h: ephone-1[1]: Jill
.
.
1d04h: SkinnyUpdateDnState by EFXS_RING_GENERATE
for DN 2 to state RINGING
.
.
1d04h: SkinnyGetCallState for DN 2 CONNECTED
.
.
1d04h: ephone-1[1]:SetLineLamp 3 to ON
1d04h: ephone-1[1]:UpdateCallState DN 1 state 4 calleddn 2
.
.
1d04h: Skinny Call State change for DN 1 CONNECTED
```

debug ephone detail

```

.
.
1d04h: ephone-1[1]:OpenReceive DN 1 codec 4:G711Ulaw64k duration 10 ms bytes 80
.
.
1d04h: ephone-1[1]:OpenReceiveChannelAck 1.2.172.21 port=20180
1d04h: ephone-1[1]:Outgoing calling DN 1 Far-ephone-2 called DN 2
1d04h: SkinnyGetCallState for DN 1 CONNECTED
.
.
1d04h: ephone-1[1]:SetCallState line 3 DN 2 TsOnHook
.
.
1d04h: ephone-1[1]:SetLineLamp 3 to OFF
.
.
1d04h: ephone-1[1]:SetCallState line 1 DN 1 TsOnHook
.
.
1d04h: ephone-1[1]:Clean Up Speakerphone state
1d04h: ephone-1[1]:SpeakerPhoneOnHook
1d04h: ephone-1[1]:Clean up activeline 1
1d04h: ephone-1[1]:StopTone sent to ephone
1d04h: ephone-1[1]:Clean Up phone offhook state
1d04h: SkinnyGetCallState for DN 1 IDLE
1d04h: called DN -1, calling DN -1 phone -1
1d04h: ephone-1[1]:SetLineLamp 1 to OFF
1d04h: UnBinding ephone-1 from DN 1
1d04h: UnBinding called DN 2 from DN 1
1d04h: ephone-1[1]:ONHOOK
1d04h: ephone-1[1]:SpeakerPhoneOnHook
1d04h: ephone-1[1]:ONHOOK NO activeline
.

```

Related Commands.

Command	Description
debug ephone alarm	Sets SkinnyStation alarm messages debugging for the Cisco IP phone.
debug ephone error	Sets error debugging for the Cisco IP phone.
debug ephone keepalive	Sets keepalive debugging for the Cisco IP phone.
debug ephone loopback	Sets MWI debugging for the Cisco IP phone.
debug ephone pak	Provides voice packet level debugging and prints the contents of one voice packet in every 1024 voice packets.
debug ephone raw	Provides raw low-level protocol debugging display for all SCCP messages.
debug ephone register	Sets registration debugging for the Cisco IP phone.
debug ephone state	Sets state debugging for the Cisco IP phone.
debug ephone statistics	Sets statistics debugging for the Cisco IP phone.
show debugging	Displays information about the types of debugging that are enabled for your router.

debug ephone error

To set error debugging for the Cisco IP phone, use the **debug ephone error** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ephone error [**mac-address** *mac-address*]

no debug ephone error [**mac-address** *mac-address*]

Syntax Description

mac-address	(Optional) Defines the MAC address of the Cisco IP phone.
<i>mac-address</i>	(Optional) Specifies the MAC address of the Cisco IP phone.

Defaults

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(5)YD	This command was introduced on the following platforms: Cisco 2600 series and Cisco 3600 series multiservice routers, and Cisco IAD2420 series Integrated Access Devices (IADs).
12.2(2)XT	This command was implemented on the Cisco 1750 and Cisco 1751 multiservice routers.
12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T and implemented on the Cisco 3725 and Cisco 3745 routers.
12.2(8)T1	This command was implemented on the Cisco 2600-XM and Cisco 2691 routers.
12.2(11)T	This command was implemented on the Cisco 1760 routers.

Usage Guidelines

The **debug ephone error** command cancels debugging at the detail and state level.

If the **mac-address** keyword is not used, the **debug ephone error** command debugs all Cisco IP phones that are registered to the router. You can remove debugging for the Cisco IP phones that you do not want to debug by using the **mac-address** keyword with the **no** form of this command.

You can enable or disable debugging on any number of Cisco IP phones. To see the Cisco IP phones that have debugging enabled, enter the **show ephone** command and look at the debug field in the output. When debugging is enabled for a Cisco IP phone, the debug output is displayed for the directory numbers associated with the Cisco IP phone.

Examples

The following is sample output of error debugging for the Cisco IP phone with MAC address 0030.94c3.8724:

```
Router# debug ephone error mac-address 0030.94c3.8724
```

■ debug ephone error

```
EPHONE error debugging is enabled
```

```
socket [2] send ERROR 11
Skinny Socket [2] retry failure
```

Related Commands

Command	Description
debug ephone alarm	Sets SkinnyStation alarm messages debugging for the Cisco IP phone.
debug ephone detail	Sets detail debugging for the Cisco IP phone.
debug ephone keepalive	Sets keepalive debugging for the Cisco IP phone.
debug ephone loopback	Sets MWI debugging for the Cisco IP phone.
debug ephone pak	Provides voice packet level debugging and prints the contents of one voice packet in every 1024 voice packets.
debug ephone raw	Provides raw low-level protocol debugging display for all SCCP messages.
debug ephone register	Sets registration debugging for the Cisco IP phone.
debug ephone state	Sets state debugging for the Cisco IP phone.
debug ephone statistics	Sets statistics debugging for the Cisco IP phone.
show debugging	Displays information about the types of debugging that are enabled for your router.

debug ephone extension-assigner

To display status messages produced by the extension assigner application, use the **debug ephone extension-assigner** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ephone extension-assigner

no debug ephone extension-assigner

Syntax Description This command has no arguments or keywords.

Command Default Debug ephone extension-assigner is disabled.

Command Modes Privileged EXEC

Command History	Cisco IOS Release	Cisco Product	Modification
	12.4(4)XC4	Cisco Unified CME 4.0(3)	This command was introduced.
	12.4(11)XJ	Cisco Unified CME 4.1	This command was introduced.
	12.4(15)T	Cisco Unified CME 4.1	This command was integrated into Cisco IOS Release 12.4(15)T.

Usage Guidelines This command displays status messages produced by the extension assigner application, including messages related to the functions performed by the following Tcl commands:

- phone query—Verifies whether the ephone tag has been assigned a MAC address.
- phone assign—Binds the MAC address from the caller's phone to a preexisting ephone template.
- phone unassign—Removes the MAC address from the ephone tag.

Before using this command, you must load the Tcl script for the extension assigner application.

Examples The following is sample output of extension assigner debugging as the extension assigner application queries phones for their status and issues commands to assign or unassign extension numbers.

```
*Jun 9 19:08:10.627: ephone_query: inCallID=47, tag=4, ephone_tag=4
*Jun 9 19:08:10.627: extAssigner_IsEphoneMacPreset: ephone_tag = 4,
ipKeyswitch.max_ephones = 96
*Jun 9 19:08:10.627: extAssigner_IsEphoneMacPreset: ephone_ptr->mac_addr_str =
000B46BDE075, MAC_EXT_RESERVED_VALUE = 02EAEAEA0000
*Jun 9 19:08:10.627: SkinnyGetActivePhoneIndexFromCallid: callID = 47
*Jun 9 19:08:10.627: SkinnyGetActivePhoneIndexFromCallid: vdbPtr->physical_interface_type
(26); CV_VOICE_EFXS (26)
*Jun 9 19:08:10.627: SkinnyGetActivePhoneIndexFromCallid: vdbPtr->type (6);
CC_IF_TELEPHONY (6)
*Jun 9 19:08:10.627: SkinnyGetActivePhoneIndexFromCallid: htsp->sig_type (26);
CV_VOICE_EFXS (26)
```

debug ephone extension-assigner

```

*Jun 9 19:08:10.627: SkinnyGetActivePhoneIndexFromCallid: dn = 4, chan = 1
*Jun 9 19:08:10.627: ephone_query: EXTASSIGNER_RC_SLOT_ASSIGNED_TO_CALLING_PHONE
*Jun 9 19:08:22.763: ephone_unassign: inCallID=47, tag=4, ephone_tag=4
*Jun 9 19:08:22.763: extAssigner_IsEphoneMacPreset: ephone_tag = 4,
ipKeyswitch.max_ephones = 96

*Jun 9 19:08:22.763: extAssigner_IsEphoneMacPreset: ephone_ptr->mac_addr_str =
000B46BDE075, MAC_EXT_RESERVED_VALUE = 02EAEAEA000
*Jun 9 19:08:22.763: is_ephone_auto_assigned: button-1 dn_tag=4
*Jun 9 19:08:22.763: is_ephone_auto_assigned: NO
*Jun 9 19:08:22.763: SkinnyGetActivePhoneIndexFromCallid: callID = 47
*Jun 9 19:08:22.763: SkinnyGetActivePhoneIndexFromCallid: vdbPtr->physical_interface_type
(26); CV_VOICE_EFXS (26)
*Jun 9 19:08:22.767: SkinnyGetActivePhoneIndexFromCallid: vdbPtr->type (6);
CC_IF_TELEPHONY (6)
*Jun 9 19:08:22.767: SkinnyGetActivePhoneIndexFromCallid: http->sig_type (26);
CV_VOICE_EFXS (26)
*Jun 9 19:08:22.767: SkinnyGetActivePhoneIndexFromCallid: dn = 4, chan = 1
*Jun 9 19:08:29.795: ephone-4[8]:fStationOnHookMessage: Extension Assigner request
restart, cmd=2, new mac=02EAEAEA0004, ephone_tag=4
*Jun 9 19:08:30.063: %IPPHONE-6-UNREGISTER_NORMAL: ephone-4:SEP000B46BDE075 IP:5.5.0.1
Socket:8 DeviceType:Phone has unregistered normally.
*Jun 9 19:08:30.063: ephone-4[8][SEP000B46BDE075]:extAssigner_assign: new
mac=02EAEAEA0004, ephone_tag=4
*Jun 9 19:08:30.063: extAssigner_simple_assign: mac=02EAEAEA0004, tag=4
*Jun 9 19:08:30.063: ephone_updateCNF: update cnf_file ephone_tag=4
*Jun 9 19:08:30.063: extAssigner_assign: restart again (mac=02EAEAEA0004) ephone_tag=4
*Jun 9 19:08:30.131: %IPPHONE-6-REG_ALARM: 23: Name=SEP000B46BDE075 Load=8.0(2.0)
Last=Reset-Restart
*Jun 9 19:08:30.135: %IPPHONE-6-REGISTER_NEW: ephone-7:SEP000B46BDE075 IP:5.5.0.1
Socket:10 DeviceType:Phone has registered.
*Jun 9 19:08:30.503: %IPPHONE-6-UNREGISTER_NORMAL: ephone-7:SEP000B46BDE075 IP:5.5.0.1
Socket:10 DeviceType:Phone has unregistered normally.
*Jun 9 19:08:43.127: %IPPHONE-6-REG_ALARM: 22: Name=SEP000B46BDE075 Load=8.0(2.0)
Last=Reset-Reset
*Jun 9 19:08:43.131: %IPPHONE-6-REGISTER: ephone-7:SEP000B46BDE075 IP:5.5.0.1 Socket:13
DeviceType:Phone has registered.

```

Related Commands

Command	Description
debug ephone state	Sets state debugging for Cisco IP phones.
debug voip application script	Displays status messages produced by voice over IP application scripts.

debug ephone keepalive

To set keepalive debugging for the Cisco IP phone, use the **debug ephone keepalive** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

```
debug ephone keepalive [mac-address mac-address]
```

```
no debug ephone keepalive [mac-address mac-address]
```

Syntax Description

mac-address	(Optional) Defines the MAC address of the Cisco IP phone.
<i>mac-address</i>	(Optional) Specifies the MAC address of the Cisco IP phone.

Defaults

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(5)YD	This command was introduced on the following platforms: Cisco 2600 series and Cisco 3600 series multiservice routers, and Cisco IAD2420 series Integrated Access Devices (IADs).
12.2(2)XT	This command was implemented on the Cisco 1750 and Cisco 1751 multiservice routers.
12.2(8)T	This command was implemented on the Cisco 3725 and Cisco 3745 routers.
12.2(8)T1	This command was implemented on the Cisco 2600-XM and Cisco 2691 routers.
12.2(11)T	This command was implemented on the Cisco 1760 routers.

Usage Guidelines

The **debug ephone keepalive** command sets keepalive debugging.

If the **mac-address** keyword is not used, the **debug ephone keepalive** command debugs all Cisco IP phones that are registered to the router. You can remove debugging for the Cisco IP phones that you do not want to debug by using the **mac-address** keyword with the **no** form of this command.

You can enable or disable debugging on any number of Cisco IP phones. To see the Cisco IP phones that have debugging enabled, enter the **show ephone** command and look at the debug field in the output. When debugging is enabled for a Cisco IP phone, the debug output is displayed for the directory numbers associated with the Cisco IP phone.

Examples

The following is sample output of the keepalive status for the Cisco IP phone with MAC address 0030.94C3.E1A8:

```
Router# debug ephone keepalive mac-address 0030.94c3.E1A8
```

debug ephone keepalive

EPHONE keepalive debugging is enabled for phone 0030.94C3.E1A8

```
1d05h: ephone-1 Set interface FastEthernet0/0 ETHERNET
1d05h: ephone-1[1]:Keepalive socket[1] SEP003094C3E1A8
1d05h: ephone-1 Set interface FastEthernet0/0 ETHERNET
1d05h: ephone-1[1]:Keepalive socket[1] SEP003094C3E1A8
1d05h: Skinny Checking for stale sockets
1d05h: ephone-1 Set interface FastEthernet0/0 ETHERNET
1d05h: ephone-1[1]:Keepalive socket[1] SEP003094C3E1A8
1d05h: ephone-1 Set interface FastEthernet0/0 ETHERNET
1d05h: ephone-1[1]:Keepalive socket[1] SEP003094C3E1A8
1d05h: Skinny active socket list (3/96): 1 2 4
```

Related Commands

Command	Description
debug ephone alarm	Sets SkinnyStation alarm messages debugging for the Cisco IP phone.
debug ephone detail	Sets detail debugging for the Cisco IP phone.
debug ephone error	Sets error debugging for the Cisco IP phone.
debug ephone loopback	Sets MWI debugging for the Cisco IP phone.
debug ephone pak	Provides voice packet level debugging and prints the contents of one voice packet in every 1024 voice packets.
debug ephone raw	Provides raw low-level protocol debugging display for all SCCP messages.
debug ephone register	Sets registration debugging for the Cisco IP phone.
debug ephone state	Sets state debugging for the Cisco IP phone.
debug ephone statistics	Sets statistics debugging for the Cisco IP phone.
show debugging	Displays information about the types of debugging that are enabled for your router.

debug ephone loopback

To set debugging for loopback calls, use the **debug ephone loopback** command in privileged EXEC mode. To disable debugging, use the **no** form of this command.

debug ephone loopback [**mac-address** *mac-address*]

no debug ephone loopback [**mac-address** *mac-address*]

Syntax Description

mac-address <i>mac-address</i>	(Optional) Specifies the MAC address of a Cisco IP phone for debugging.
---------------------------------------	-------------------------------------------------------------------------

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(2)XT	This command was introduced for Cisco IOS Telephony Services (now known as Cisco CallManager Express) Version 2.0 on the Cisco 1750, Cisco 1751, Cisco 2600 series, Cisco 3600 series, and Cisco IAD2420 series.
12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T and implemented on the Cisco 3725 and Cisco 3745.
12.2(8)T1	This command was implemented on the Cisco 2600-XM and Cisco 2691.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.

Usage Guidelines

The **debug ephone loopback** command sets debugging for incoming and outgoing calls on all loopback-dn pairs or on the single loopback-dn pair that is associated with the IP phone that has the MAC address specified in this command.

If you enable the **debug ephone loopback** command and the **debug ephone pak** command at the same time, the output displays packet debug output for the voice packets that are passing through the loopback-dn pair.

You can enable or disable debugging on any number of Cisco IP phones. To see the Cisco IP phones that have debugging enabled, enter the **show ephone** command and look at the debug field in the output. When debugging is enabled for a Cisco IP phone, the debug output is displayed for the directory numbers associated with that Cisco IP phone.

Examples

The following example contains two excerpts of output for a call that is routed through a loopback. The first excerpt is output from the **show running-config** command and displays the loopback configuration used for this example. The second excerpt is output from the **debug ephone loopback** command.

```
Router# show running-config
```

```
.  
.
.
```

debug ephone loopback

```

ephone-dn 14
  number 1514
!
!
ephone-dn 42
  number 17181..
  loopback-dn 43 forward 4
  no huntstop
!
!
ephone-dn 43
  number 19115..
  loopback-dn 42 forward 4
!
.
.
.

```

A loopback call is started. An incoming call to 1911514 (ephone-dn 43) uses the loopback pair of ephone-dns to become an outgoing call to extension 1514. The number in the outgoing call has only four digits because the **loopback-dn** command specifies forwarding of four digits. The outgoing call uses ephone-dn 42, which is also specified in the **loopback-dn** command under ephone-dn 43. When the extension at 1514 rings, the following debug output is displayed:

```
Router# debug ephone loopback
```

```

Mar 7 00:57:25.376:Pass processed call info to special DN 43 chan 1
Mar 7 00:57:25.376:SkinnySetCallInfoLoopback DN 43 state IDLE to DN 42 state IDLE
Mar 7 00:57:25.376:Called Number = 1911514 Called Name =
Mar 7 00:57:25.376:Calling Number = 8101 Calling Name =
  orig Called Number =
Copy Caller-ID info from Loopback DN 43 to DN 42
Mar 7 00:57:25.376:DN 43 Forward 1514
Mar 7 00:57:25.376:PredictTarget match 1514 DN 14 is idle
Mar 7 00:57:25.380:SkinnyUpdateLoopbackState DN 43 state RINGING calledDn -1
Mar 7 00:57:25.380:Loopback DN 42 state IDLE
Mar 7 00:57:25.380:Loopback DN 43 calledDN -1 callingDn -1 G711Ulaw64k
Mar 7 00:57:25.380:SkinnyUpdateLoopbackState DN 43 to DN 42 signal OFFHOOK
Mar 7 00:57:25.380:SetDnCodec Loopback DN 43 codec 4:G711Ulaw64k vad 0 size 160
Mar 7 00:57:25.380:SkinnyDnToneLoopback DN 42 state SIEZE to DN 43 state RINGING
Mar 7 00:57:25.380:TONE ON DtInsideDialTone
Mar 7 00:57:25.380:SkinnyDnToneLoopback called number = 1911514
Mar 7 00:57:25.380:DN 43 Forward 1514
Mar 7 00:57:25.380:DN 42 from 43 Dial 1514
Mar 7 00:57:25.384:SkinnyDnToneLoopback DN 42 state ALERTING to DN 43 state RINGING
Mar 7 00:57:25.384:TONE OFF
Mar 7 00:57:25.384:SkinnyDnToneLoopback DN 42 state ALERTING to DN 43 state RINGING
Mar 7 00:57:25.384:SkinnyUpdateLoopbackState DN 42 state ALERTING calledDn -1
Mar 7 00:57:25.384:Loopback DN 43 state RINGING
Mar 7 00:57:25.384:Loopback Alerting DN 42 calledDN -1 callingDn -1 G711Ulaw64k
Mar 7 00:57:25.388:ephone-5[7]:DisplayCallInfo incoming call
Mar 7 00:57:25.388:SkinnyDnToneLoopback DN 42 state ALERTING to DN 43 state RINGING
Mar 7 00:57:25.388:TONE ON DtAlertingTone
Mar 7 00:57:25.388:SkinnyDnToneLoopback DN 42 to DN 43 deferred alerting by
DtAlertingTone
Mar 7 00:57:25.388:EFXS_STATE_ONHOOK_RINGING already done for DN 43 chan 1
Mar 7 00:57:25.388:Set prog_ind 0 for DN 42 chan 1
.
.
.

```

When extension 1514 answers the call, the following debug output is displayed:

```

.
.
Mar 7 00:57:32.158:SkinnyDnToneLoopback DN 42 state ALERTING to DN 43 state RINGING
Mar 7 00:57:32.158:TONE OFF
Mar 7 00:57:32.158:dn_support_g729 true DN 42 chan 1 (loopback)
Mar 7 00:57:32.158:SetDnCodec Loopback DN 43 codec 4:G711Ulaw64k vad 0 size 160
Mar 7 00:57:32.158:SkinnyUpdateLoopbackState DN 42 state CALL_START calledDn 14
Mar 7 00:57:32.158:Loopback DN 43 state RINGING
Mar 7 00:57:32.158:SkinnyUpdateLoopbackState DN 42 to DN 43 deferred alerting by
CALL_START already sent
Mar 7 00:57:32.158:SetDnCodec reassert defer_start for DN 14 chan 1
Mar 7 00:57:32.158:Delay media until loopback DN 43 is ready
Mar 7 00:57:32.158:SkinnyUpdateLoopbackCodec check for DN 14 chan 1 from DN 42 loopback
DN 43
Mar 7 00:57:32.158:SkinnyUpdateLoopbackCodec DN chain is 14 1, other=42, lb=43, far=-1 1,
final=43 1
Mar 7 00:57:32.158:SkinnyUpdateLoopbackCodec DN 14 chan 1 DN 43 chan 1 codec 4 match
Mar 7 00:57:32.162:SkinnyUpdateLoopbackState DN 42 state CONNECTED calledDn 14
Mar 7 00:57:32.162:Loopback DN 43 state RINGING
Mar 7 00:57:32.162:SkinnyUpdateLoopbackState DN 42 to DN 43 signal ANSWER
Mar 7 00:57:32.162:Loopback DN 42 calledDN 14 callingDn -1 G711Ulaw64k
Mar 7 00:57:32.162:Loopback DN 43 calledDN -1 callingDn -1 incoming G711Ulaw64k
Mar 7 00:57:32.162:ephone-5[7][SEP000DBBEF37D]:refreshDisplayLine for line 1 DN 14 chan
1
Mar 7 00:57:32.162:dn_support_g729 true DN 43 chan 1 (loopback)
Mar 7 00:57:32.162:SetDnCodec Loopback DN 42 codec 4:G711Ulaw64k vad 0 size 160
Mar 7 00:57:32.162:SkinnyUpdateLoopbackState DN 43 state CALL_START calledDn -1
Mar 7 00:57:32.162:Loopback DN 42 state CONNECTED
Mar 7 00:57:32.162:SkinnyUpdateLoopbackState DN 43 has defer_dn 14 chan 1 set
Mar 7 00:57:32.162:SkinnyUpdateLoopbackState DN 43 has defer_dn 14 chan 1:
-invoke SkinnyOpenReceive
Mar 7 00:57:32.162:SkinnyUpdateLoopbackCodec check for DN 14 chan 1 from DN 42 loopback
DN 43
Mar 7 00:57:32.162:SkinnyUpdateLoopbackCodec DN chain is 14 1, other=42, lb=43, far=-1 1,
final=43 1
Mar 7 00:57:32.162:SkinnyUpdateLoopbackCodec DN 14 chan 1 DN 43 chan 1 codec 4 match
Mar 7 00:57:32.162:SkinnyUpdateLoopbackState DN 43 state CALL_START calledDn -1
Mar 7 00:57:32.162:Loopback DN 42 state CONNECTED
Mar 7 00:57:32.454:SkinnyGetDnAddrInfo DN 43 LOOPBACK
update media address to 10.0.0.6 25390 from DN 14
Mar 7 00:57:33.166:ephone-5[7]:DisplayCallInfo incoming call
.
.

```

When the called extension, 1514, goes back on-hook, the following debug output is displayed:

```

.
.
Mar 7 00:57:39.224:Loopback DN 42 disc reason 16 normal state CONNECTED
Mar 7 00:57:39.224:SkinnyUpdateLoopbackState DN 42 state CALL_END calledDn -1
Mar 7 00:57:39.224:Loopback DN 43 state CONNECTED
Mar 7 00:57:39.224:SkinnyUpdateLoopbackState DN 42 to DN 43 signal ONHOOK
Mar 7 00:57:39.236:SkinnyDnToneLoopback DN 42 state IDLE to DN 43 state IDLE
Mar 7 00:57:39.236:TONE OFF
Mar 7 00:57:39.236:SkinnyDnToneLoopback DN 43 state IDLE to DN 42 state IDLE
Mar 7 00:57:39.236:TONE OFF

```

Table 5 describes the significant fields shown in the display.

Table 5 *debug ephone loopback Field Descriptions*

Field	Description
Called Number	Original called number as presented to the incoming side of the loopback-dn.
Forward	Outgoing number that is expected to be dialed by the outgoing side of the loopback-dn pair.
PredictTarget Match	Extension (ephone-dn) that is anticipated by the loopback-dn to be the far-end termination for the call.
signal OFFHOOK	Indicates that the outgoing side of the loopback-dn pair is going off-hook prior to placing the outbound call leg.
Dial	Outbound side of the loopback-dn that is actually dialing the outbound call leg.
deferred alerting	Indicates that the alerting, or ringing, tone is returning to the original inbound call leg in response to the far-end ephone-dn state.
DN chain	Chain of ephone-dns that has been detected, starting from the far-end that terminates the call. Each entry in the chain indicates an ephone-dn tag and channel number. Entries appear in the following order, from left to right: <ul style="list-style-type: none"> Ephone-dn tag and channel of the far-end call terminator (in this example, ephone-dn 14 is extension 1514). other—Ephone-dn tag of the outgoing side of the loopback. lb—Ephone-dn tag of the incoming side of the loopback. far—Ephone-dn tag and channel of the far-end call originator, or -1 for a nonlocal number. final—Ephone-dn tag for the originator of the call on the incoming side of the loopback. If the originator is not a local ephone-dn, this is set to -1. This number represents the final ephone-dn tag in the chain, looking toward the originator.
codec match	Indicates that there is no codec conflict between the two calls on either side of the loopback-dn.
GetDnAddrInfo	IP address of the IP phone at the final destination extension (ephone-dn), after resolving the chain of ephone-dns involved.
disc_reason	Disconnect cause code, in decimal. These are normal CC_CAUSE code values that are also used in call control API debugging. Common cause codes include the following: <ul style="list-style-type: none"> 16—Normal disconnect. 17—User busy. 19—No answer. 28—Invalid number.

Related Commands	Command	Description
	debug ephone pak	Provides voice packet level debugging.
	loopback-dn	Configures loopback-dn virtual loopback voice ports used to establish demarcation points for VoIP voice calls and supplementary services.
	show ephone	Displays information about registered Cisco IP phones.
	show ephone-dn loopback	Displays information for ephone-dns that have been set up for loopback calls.

debug ephone lpcor

To display debugging information for calls using the logical partitioning class of restriction (LPCOR) feature, use the **debug ephone lpcor** command in privileged EXEC mode. To disable debugging, use the **no** form of this command.

```
debug ephone lpcor [mac-address mac-address]
```

```
no debug ephone lpcor [mac-address mac-address]
```

Syntax Description

mac-address mac-address (Optional) Specifies the MAC address of a specific IP phone.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
15.0(1)XA	This command was introduced.

Usage Guidelines

Use this command for troubleshooting LPCOR calls to phones in a Cisco Unified CME system.

If the **mac-address** keyword is not used, this command debugs all phones that are registered to the Cisco Unified CME router. You can disable debugging for specific phones by using the **mac-address** keyword with the **no** form of this command.

Examples

The following is sample output from the **debug ephone lpcor** command for a call between ephone-1 and ephone-2 that was blocked by LPCOR policy validation:

```
Router# debug ephone lpcor

*Jun 24 11:23:45.599: ephone-1[0/3][SEP003094C25F38]:ephone_get_lpcor_index: dir 0
*Jun 24 11:23:46.603: ephone-2[1/2][SEP0021A02DB62A]:ephone_get_lpcor_index: dir 1
```

Related Commands

Command	Description
debug voip application lpcor	Enables debugging of the LPCOR application system.
debug voip lpcor	Displays debugging information for the LPCOR feature.
lpcor incoming	Associates an incoming call with a LPCOR resource-group policy.
lpcor outgoing	Associates an outgoing call with a LPCOR resource-group policy.
show ephone	Displays information about phones registered to Cisco Unified CME.
show voice lpcor policy	Displays the LPCOR policy for the specified resource group.

debug ephone message

To enable message tracing between ephones, use the **debug ephone message** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ephone message [detail]

no debug ephone message

Syntax Description	detail
	(Optional) Displays signaling connection control protocol (SCCP) messages sent and received between ephones in the Cisco Unified CallManager Express (Cisco Unified CME) system.

Command Modes	Privileged EXEC
---------------	-----------------

Command History	Cisco IOS Release	Modification
	12.4(4)XC	This command was introduced.
	12.4(9)T	This command was integrated into Cisco IOS Release 12.4(9)T.

Usage Guidelines	<p>The debug ephone message command enables message tracing between ephones.</p> <p>The debug ephone command debugs all ephones associated with a Cisco Unified CME router.</p> <p>You can enable or disable debugging on any number of ephones. To see the ephones that have debugging enabled, enter the show ephone command and look at the debug field in the output. When debugging is enabled for a ephone, the debug output is displayed for the directory numbers associated with the ephone.</p>
------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Examples	The following is sample output for the debug ephone message command for ephones:
----------	-----------------------------------------------------------------------------------------

```
Router# debug ephone message

EPHONE skinny message debugging is enabled
*Jul 17 12:12:54.883: Received message from phone 7, SkinnyMessageID = StationKe
epAliveMessageID
*Jul 17 12:12:54.883: Sending message to phone 7, SkinnyMessageID = StationKe
epAliveAckMessageID
```

The following command disables ephone message debugging:

```
Router# no debug ephone message

EPHONE skinny message debugging is disabled
```

Related Commands

Command	Description
debug ephone alarm	Sets SkinnyStation alarm messages debugging for the ephone.
debug ephone detail	Sets detail debugging for the ephone.
debug ephone error	Sets error debugging for the ephone.
debug ephone mwi	Sets MWI debugging for the ephone.
debug ephone pak	Provides voice packet level debugging and displays the contents of one voice packet in every 1024 voice packets.
debug ephone raw	Provides raw low-level protocol debugging display for all SCCP messages.
debug ephone register	Sets registration debugging for the ephone.
debug ephone state	Sets state debugging for the ephone.
debug ephone statistics	Sets statistics debugging for the ephone.
debug ephone video	Sets video debugging for the ephone.
show debugging	Displays information about the types of debugging that are enabled for your router.
show ephone	Displays information about ephones.

debug ephone mlpp

To display debugging information for Multilevel Precedence and Preemption (MLPP) calls to phones in a Cisco Unified CME system, use the **debug ephone mlpp** command in privileged EXEC mode. To disable debugging, use the **no** form of this command.

debug ephone mlpp [**mac-address** *mac-address*]

no debug ephone mlpp [**mac-address** *mac-address*]

Syntax Description

mac-address *mac-address* (Optional) Specifies the MAC address of a specific IP phone.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.4(22)YB	This command was introduced.
12.4(24)T	This command was integrated into Cisco IOS Release 12.4(24)T .

Usage Guidelines

Use this command to troubleshoot calls that use the MLPP service.

Examples

The following is sample output from the **debug ephone mlpp** command. This example shows output for the following call scenario:

- Ephone 1 is connected to ephone 3 (nonMLPP call).
- Ephone 4 makes an MLPP call to ephone 3. Preemption tone is played to both ephone 1 and 3.
- Ephone 3 is disconnected after the preemption tone timeout and precedence ringing.
- Ephone 3 answers the MLPP call and is connected to ephone 4.

Router# **debug ephone mlpp**

```
Sep 5 14:23:00.499: ephone-4 [3/3] [SEP001AE2BC3EE7]:indication=1
Sep 5 14:23:00.499: ephone-4 [3/3] [SEP001AE2BC3EE7]:max_precedence=0
Sep 5 14:23:02.299: ephone-4 [3/3] [SEP001AE2BC3EE7]:mlpp_ephone_display_update callID=294
Sep 5 14:23:02.299: ephone-4 [3/3] [SEP001AE2BC3EE7]:indication=1
Sep 5 14:23:02.299: ephone-4 [3/3] [SEP001AE2BC3EE7]:mlpp_precedence=4, domain=0
Sep 5 14:23:02.303: ephone-3 [2/1] [SEP001B54BA0D64]:preemption=1
Sep 5 14:23:02.303: ephone-3 [2/1] [SEP001B54BA0D64]:preemption=1
Sep 5 14:23:02.303: mlpp_ephone_find_call: preempt_htsp=1774234732,
preempt_htsp->mlpp_preemptor_cid=294
Sep 5 14:23:02.303: //294/A6B5C03A8141/VOIP-MLPP/voice_mlpp_get_preemptInfo:
mlpp_ephone_find_call is successful
Sep 5 14:23:02.303: ephone-4 [3/3] [SEP001AE2BC3EE7]:indication=1
Sep 5 14:23:02.303: ephone-4 [3/3] [SEP001AE2BC3EE7]:mlpp_precedence=4, domain=0
Sep 5 14:23:02.303: ephone-4 [3/3] [SEP001AE2BC3EE7]:indication=1
Sep 5 14:23:02.303: ephone-4 [3/3] [SEP001AE2BC3EE7]:mlpp_precedence=4, domain=0
Sep 5 14:23:02.303: ephone-6 [5/6] [SEP0018187F49FD]:indication=1
Sep 5 14:23:02.303: ephone-6 [5/6] [SEP0018187F49FD]:mlpp_precedence=4, domain=0
```

■ debug ephone mlpp

```

Sep 5 14:23:02.303: ephone-4[3/3][SEP001AE2BC3EE7]:indication=1
Sep 5 14:23:02.307: ephone-1[0/2][SEP0014A9818797]:indication=1
Sep 5 14:23:02.307: ephone-3[2/1][SEP001B54BA0D64]:indication=1
Sep 5 14:23:02.307: ephone-1[0/2][SEP0014A9818797]:indication=1DtPreemptionTone
Sep 5 14:23:02.307: ephone-3[2/1][SEP001B54BA0D64]:indication=1DtPreemptionTone
Sep 5 14:23:07.307: ephone-3[2/1][SEP001B54BA0D64]:indication=1
Sep 5 14:23:07.307: ephone-1[0/2][SEP0014A9818797]:indication=1
Sep 5 14:23:07.319: ephone-3[2/1][SEP001B54BA0D64]:indication=1
Sep 5 14:23:07.319: ephone-3[2/1][SEP001B54BA0D64]:indication=1
Sep 5 14:23:07.319: ephone-3[2/1][SEP001B54BA0D64]:mlpp precedence=4, domain=0
Sep 5 14:23:07.319: ephone-3[2/1][SEP001B54BA0D64]:indication=1
Sep 5 14:23:07.319: ephone-3[2/1][SEP001B54BA0D64]: MLPP Precedence Ring 6 instead
Sep 5 14:23:10.623: ephone-3[2/1][SEP001B54BA0D64]:indication=1
Sep 5 14:23:10.623: ephone-3[2/1][SEP001B54BA0D64]:indication=1
Sep 5 14:23:10.623: ephone-3[2/1][SEP001B54BA0D64]:mlpp precedence=4, domain=0
Sep 5 14:23:10.623: ephone-3[2/1][SEP001B54BA0D64]:indication=1
Sep 5 14:23:10.623: ephone-3[2/1][SEP001B54BA0D64]:mlpp precedence=4, domain=0
Sep 5 14:23:10.623: ephone-4[3/3][SEP001AE2BC3EE7]:indication=1
Sep 5 14:23:10.623: ephone-4[3/3][SEP001AE2BC3EE7]:mlpp precedence=4, domain=0
Sep 5 14:23:10.623: ephone-6[5/6][SEP0018187F49FD]:indication=1
Sep 5 14:23:10.623: ephone-6[5/6][SEP0018187F49FD]:mlpp precedence=4, domain=0

```

Related Commands

Command	Description
debug voice mlpp	Displays debugging information for MLPP service.
mlpp indication	Enables MLPP indication on an SCCP phone or analog FXS port.
mlpp max-precedence	Sets the maximum precedence (priority) level that a phone user can specify when making an MLPP call.
mlpp preemption	Enables preemption capability on an SCCP phone or analog FXS port.

debug ephone moh

To set debugging for music on hold (MOH), use the **debug ephone moh** command in privileged EXEC mode. To disable debugging, use the **no** form of this command.

debug ephone moh [**mac-address** *mac-address*]

no debug ephone moh [**mac-address** *mac-address*]

Syntax	Description
mac-address <i>mac-address</i>	(Optional) Specifies the MAC address of a Cisco IP phone for debugging.

Command Modes	Description
Privileged EXEC	

Command History	Release	Modification
	12.2(2)XT	This command was introduced for Cisco IOS Telephony Services (now known as Cisco CallManager Express) Version 2.0 and Cisco Survivable Remote Site Telephony (SRST) Version 2.0 on the Cisco 1750, Cisco 1751, Cisco 2600 series, Cisco 3600 series, and Cisco IAD2420 series.
	12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T and implemented on the Cisco 3725 and Cisco 3745.
	12.2(8)T1	This command was implemented on the Cisco 2600-XM and Cisco 2691.
	12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.

Usage Guidelines Always use the **no moh** command before modifying or replacing the MOH file in Flash memory.

When a configuration using the **multicast moh** command is used and the **debug ephone moh** command is enabled, if you delete or modify the MOH file in the router's Flash memory, the debug output can be excessive and can flood the console. The multicast MOH configuration should be removed before using the **no moh** command when the **debug ephone moh** command is enabled.

Examples The following sample output shows MOH activity prior to the first MOH session. Note that if you enable multicast MOH, that counts as the first session.

```
Router# debug ephone moh

Mar  7 00:52:33.817:MOH AU file
Mar  7 00:52:33.817:skinny_open_moh_play set type to 3
Mar  7 00:52:33.825: 2E73 6E64 0000 0018 0007 3CCA 0000 0001
Mar  7 00:52:33.825: 0000 1F40 0000 0001 FFFF FFFF FFFF FFFF
Mar  7 00:52:33.825: FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
Mar  7 00:52:33.825: FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
Mar  7 00:52:33.825: FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
```

```

Mar 7 00:52:33.825: FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
Mar 7 00:52:33.825: FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
Mar 7 00:52:33.825: FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
Mar 7 00:52:33.825: FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
Mar 7 00:52:33.825: FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
Mar 7 00:52:33.825: FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
Mar 7 00:52:33.825: FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
Mar 7 00:52:33.825: FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
Mar 7 00:52:33.825: FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
Mar 7 00:52:33.825: FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
Mar 7 00:52:33.825: FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
Mar 7 00:52:33.825:
Mar 7 00:52:33.825:AU file processing Found .snd
Mar 7 00:52:33.825:AU file data start at 24 end at 474338
Mar 7 00:52:33.825:AU file codec Media_Payload_G711Ulaw64k
Mar 7 00:52:33.825:MOH read file header type AU start 24 end 474338
Mar 7 00:52:33.825:MOH pre-read block 0 at write-offset 0 from 24
Mar 7 00:52:33.833:MOH pre-read block 1 at write-offset 8000 from 8024
Mar 7 00:52:33.845:Starting read server with play-offset 0 write-offset 16000

```

Table 6 describes the significant fields shown in the display.

Table 6 *debug ephone moh Field Descriptions*

Field	Description
type	0—invalid 1—raw file 2—wave format file (.wav) 3—AU format (.au) 4—live feed
AU file processing Found .snd	A .snd header was located in the AU file.
AU file data start at, end at	Data start and end file offset within the MOH file, as indicated by the file header.
read file header type	File format found (AU, WAVE, or RAW).
pre-read block, write-offset	Location in the internal MOH buffer to which data is being written, and location from which that data was read in the file.
play-offset, write-offset	Indicates the relative positioning of MOH file read-ahead buffering. Data is normally written from a Flash file into the internal circular buffer, ahead of the location from which data is being played or output.

Related Commands

Command	Description
moh (telephony-service)	Generates an audio stream from a file for MOH in a Cisco CME system.
multicast moh	Uses the MOH audio stream as a multicast source in a Cisco CME system.

debug ephone mwi

To set message waiting indication (MWI) debugging for the Cisco IOS Telephony Service router, use the **debug ephone mwi** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ephone mwi

no debug ephone mwi

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(2)XT	This command was introduced on the following platforms: Cisco 1750, Cisco 1751, Cisco 2600 series and Cisco 3600 series multiservice routers; and Cisco IAD2420 series Integrated Access Devices (IADs).
12.2(8)T	This command was implemented on the Cisco 3725 and Cisco 3745 routers.
12.2(8)T1	This command was implemented on the Cisco 2600-XM and Cisco 2691 routers.
12.2(11)T	This command was implemented on the Cisco 1760 routers.

Usage Guidelines

The **debug ephone mwi** command sets message waiting indication debugging for the Cisco IOS Telephony Service router. Because the MWI protocol activity is not specific to any individual Cisco IP phone, setting the MAC address keyword qualifier for this command is not useful.



Note

Unlike the other related **debug ephone** commands, the **mac-address** keyword does not help debug a particular Cisco IP phone.

Examples

The following is sample output of the message waiting indication status for the Cisco IOS Telephony Service router:

```
Router# debug ephone mwi
```

Related Commands	Command	Description
	debug ephone alarm	Sets SkinnyStation alarm messages debugging for the Cisco IP phone.
	debug ephone detail	Sets detail debugging for the Cisco IP phone.
	debug ephone error	Sets error debugging for the Cisco IP phone.
	debug ephone keepalive	Sets keepalive debugging for the Cisco IP phone.
	debug ephone pak	Provides voice packet level debugging and prints the contents of one voice packet in every 1024 voice packets.
	debug ephone raw	Provides raw low-level protocol debugging display for all SCCP messages.
	debug ephone register	Sets registration debugging for the Cisco IP phone.
	debug ephone state	Sets state debugging for the Cisco IP phone.
	debug ephone statistics	Sets statistics debugging for the Cisco IP phone.
	show debugging	Displays information about the types of debugging that are enabled for your router.

debug ephone pak

To provide voice packet level debugging and to print the contents of one voice packet in every 1024 voice packets, use the **debug ephone pak** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ephone pak [**mac-address** *mac-address*]

no debug ephone pak [**mac-address** *mac-address*]

Syntax Description

mac-address	(Optional) Defines the MAC address of the Cisco IP phone.
<i>mac-address</i>	(Optional) Specifies the MAC address of the Cisco IP phone.

Defaults

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(5)YD	This command was introduced on the following platforms: Cisco 2600 series and Cisco 3600 series multiservice routers, and Cisco IAD2420 series Integrated Access Devices (IADs).
12.2(2)XT	This command was implemented on the Cisco 1750 and Cisco 1751 multiservice routers.
12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T and implemented on the Cisco 3725 and Cisco 3745 routers.
12.2(8)T1	This command was implemented on the Cisco 2600-XM and Cisco 2691 routers.
12.2(11)T	This command was implemented on the Cisco 1760 routers.

Usage Guidelines

The **debug ephone pak** command provides voice packet level debugging and prints the contents of one voice packet in every 1024 voice packets.

If the **mac-address** keyword is not used, the **debug ephone pak** command debugs all Cisco IP phones that are registered to the router. You can remove debugging for the Cisco IP phones that you do not want to debug by using the **mac-address** keyword with the **no** form of this command.

You can enable or disable debugging on any number of Cisco IP phones. To see the Cisco IP phones that have debugging enabled, enter the **show ephone** command and look at the debug field in the output. When debugging is enabled for a Cisco IP phone, the debug output is displayed for the directory numbers associated with the Cisco IP phone.

Examples

The following is sample output of packet debugging for the Cisco IP phone with MAC address 0030.94c3.8724:

```

Router# debug ephone pak mac-address 0030.94c3.8724

EPHONE packet debugging is enabled for phone 0030.94c3.8724

01:29:14: ***ph_xmit_ephone DN 3 tx_pkts 5770 dest=10.2.1.1 orig len=32
  pakcopy=0 discards 27 ip_enctype 0 0 last discard: unsupported payload type
01:29:14: to_skinny_duration 130210 offset -30 last -40 seq 0 adj 0
01:29:14: IP: 45B8 003C 0866 0000 3F11 3F90 2800 0001 0A02 0101
01:29:14: TTL 63 TOS B8 prec 5
01:29:14: UDP: 07D0 6266 0028 0000
01:29:14: sport 2000 dport 25190 length 40 checksum 0
01:29:14: RTP: 8012 16AF 9170 6409 0E9F 0001
01:29:14: is_rtp:1 is_frfl1:0 vlen:0 delta_t:160 vofr1:0 vofr2:0
scodec:11 rtp_bits:8012 rtp_codec:18 last_bad_payload 19
01:29:14: vencap FAILED
01:29:14: PROCESS SWITCH
01:29:15: %SYS-5-CONFIG_I: Configured from console by console
01:29:34: ***SkinnyPktIp DN 3 10.2.1.1 to 40.0.0.1 pkts 4880 FAST sw
01:29:34: from_skinny_duration 150910
01:29:34: nw 3BBC2A8 addr 3BBC2A4 mac 3BBC2A4 dg 3BBC2C4 dgs 2A
01:29:34: MAC: 1841 0800
01:29:34: IP: 45B8 0046 682E 0000 3E11 E0BD 0A02 0101 2800 0001
01:29:34: TTL 62 TOS B8 prec 5
01:29:34: UDP: 6266 07D0 0032 0000
01:29:34: sport 25190 dport 2000 length 50 checksum 0
01:29:34: RTP: 8012 55FF 0057 8870 3AF4 C394
01:29:34: RTP: rtp_bits 8012 seq 55FF ts 578870 ssrc 3AF4C394
01:29:34: PAYLOAD:
01:29:34: 1409 37C9 54DE 449C 3B42 0446 3AAB 182E
01:29:34: 56BC 5184 58E5 56D3 13BE 44A7 B8C4
01:29:34:
01:29:37: ***ph_xmit_ephone DN 3 tx_pkts 6790 dest=10.2.1.1 orig len=32
  pakcopy=0 discards 31 ip_enctype 0 0 last discard: unsupported payload type
01:29:37: to_skinny_duration 153870 offset -150 last -40 seq 0 adj 0
01:29:37: IP: 45B8 003C 0875 0000 3F11 3F81 2800 0001 0A02 0101
01:29:37: TTL 63 TOS B8 prec 5
01:29:37: UDP: 07D0 6266 0028 0000
01:29:37: sport 2000 dport 25190 length 40 checksum 0
01:29:37: RTP: 8012 1AAF 9173 4769 0E9F 0001
01:29:37: is_rtp:1 is_frfl1:0 vlen:0 delta_t:160 vofr1:0 vofr2:0

```

Related Commands

Command	Description
debug ephone alarm	Sets SkinnyStation alarm messages debugging for the Cisco IP phone.
debug ephone detail	Sets detail debugging for the Cisco IP phone.
debug ephone error	Sets error debugging for the Cisco IP phone.
debug ephone keepalive	Sets keepalive debugging for the Cisco IP phone.
debug ephone loopback	Sets MWI debugging for the Cisco IP phone.
debug ephone raw	Provides raw low-level protocol debugging display for all SCCP messages.
debug ephone register	Sets registration debugging for the Cisco IP phone.
debug ephone state	Sets state debugging for the Cisco IP phone.
debug ephone statistics	Sets statistics debugging for the Cisco IP phone.
show debugging	Displays information about the types of debugging that are enabled for your router.

debug ephone qov

To display quality of voice (QOV) statistics for calls when preset limits are exceeded, use the **debug ephone qov** command in privileged EXEC mode. To disable debugging, use the **no** form of this command.

```
debug ephone qov [mac-address mac-address]
```

```
no debug ephone qov [mac-address mac-address]
```

Syntax Description

mac-address mac-address (Optional) Specifies the MAC address of a Cisco IP phone for debugging.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.2(15)ZJ2	This command was introduced for Cisco CallManager Express 3.0 and Cisco Survivable Remote Site Telephony (SRST) Version 3.0.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.

Usage Guidelines

Once enabled, the **debug ephone qov** command produces output only when the QOV statistics reported by phones exceed preset limits. Phones are polled every few seconds for QOV statistics on VoIP calls only, not on local PSTN calls. An output report is produced when limits are surpassed for either or both of the following:

- Lost packets—A report is triggered when two adjacent QOV samples show an increase of four or more lost packets between samples. The report is triggered by an increase of lost packets in a short period of time, not by the total number of lost packets.
- Jitter and latency—A report is triggered when either jitter or latency exceeds 100 milliseconds.

To receive a QOV report at the end of each call regardless of whether the QOV limits have been exceeded, enable the **debug ephone alarm** command in addition to the **debug ephone qov** command.

The **debug ephone statistics** command displays the raw statistics that are polled from phones and used to generate QOV reports.

Examples

The following sample output describes QOV statistics for a call on ephone 5:

```
Router# debug ephone qov

Mar  7 00:54:57.329:ephone-5[7]:QOV DN 14 chan 1 (1514) ref 4 called=1514 calling=8101
Mar  7 00:54:57.329:ephone-5[7][SEP000DBDBEF37D]:Lost 91 Jitter 0 Latency 0
Mar  7 00:54:57.329:ephone-5[7][SEP000DBDBEF37D]:previous Lost 0 Jitter 0 Latency 0
Mar  7 00:54:57.329:ephone-5[7][SEP000DBDBEF37D]:Router sent 1153 pkts, current phone got
1141
received by all (shared) phones 0
Mar  7 00:54:57.329:ephone-5[7]:worst jitter 0 worst latency 0
Mar  7 00:54:57.329:ephone-5[7]:Current phone sent 1233 packets
```

Mar 7 00:54:57.329:ephone-5[7]:Signal Level to phone 3408 (-15 dB) peak 3516 (-15 dB)

Table 7 describes the significant fields shown in the display.

Table 7 *debug ephone qov Field Descriptions*

Field	Description
Lost	Number of lost packets reported by the IP phone.
Jitter, Latency	The most recent jitter and latency parameters reported by the IP phone.
previous Lost, Jitter, Latency	Values from the previous QOV statistics report that were used as the comparison points against which the current statistics triggered generation of the current report.
Router sent pkts	Number of packets sent by the router to the IP phone. This number is the total for the entire call, even if the call is moved from one phone to another during a call, which can happen with shared lines.
current phone got	Number of packets received by the phone currently terminating the call. This number is the total for the entire call, even if the call is moved from one phone to another during a call, which can happen with shared lines.
worst jitter, worst latency	Highest value reported by the phone during the call.
Current phone sent packets	Number of packets that the current phone claims it sent during the call.
Signal Level to phone	Signal level seen in G.711 voice packet data prior to the sending of the most recent voice packet to the phone. The first number is the raw sample value, converted from G.711 to 16-bit linear format and left-justified. The number in parentheses is the value in decibels (dB), assuming that 32,767 is about +3 dB. Note This value is meaningful only if the call uses a G.711 codec.

Related Commands

Command	Description
debug ephone alarm	Displays alarm messages for IP phones.
debug ephone statistics	Displays call statistics for IP phones.

debug ephone raw

To provide raw low-level protocol debugging display for all Skinny Client Control Protocol (SCCP) messages, use the **debug ephone raw** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ephone raw [*mac-address mac-address*]

no debug ephone raw [*mac-address mac-address*]

Syntax Description

mac-address	(Optional) Defines the MAC address of the Cisco IP phone.
<i>mac-address</i>	(Optional) Specifies the MAC address of the Cisco IP phone.

Defaults

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(5)YD	This command was introduced on the following platforms: Cisco 2600 series and Cisco 3600 series multiservice routers, and Cisco IAD2420 series Integrated Access Devices (IADs).
12.2(2)XT	This command was implemented on the Cisco 1750 and Cisco 1751 multiservice routers.
12.2(8)T	This command was implemented on the Cisco 3725 and Cisco 3745 routers.
12.2(8)T1	This command was implemented on the Cisco 2600-XM and Cisco 2691 routers.
12.2(11)T	This command was implemented on the Cisco 1760 routers.

Usage Guidelines

The **debug ephone raw** command provides raw low-level protocol debug display for all SCCP messages. The debug display provides byte level display of Skinny TCP socket messages.

If the **mac-address** keyword is not used, the **debug ephone raw** command debugs all Cisco IP phones that are registered to the router. You can remove debugging for the Cisco IP phones that you do not want to debug by using the **mac-address** keyword with the **no** form of this command.

You can enable or disable debugging on any number of Cisco IP phones. To see the Cisco IP phones that have debugging enabled, enter the **show ephone** command and look at the debug field in the output. When debugging is enabled for a Cisco IP phone, the debug output is displayed for the directory numbers associated with the Cisco IP phone.

Examples

The following is sample output of raw protocol debugging for the Cisco IP phone with MAC address 0030.94c3.E1A8:

```
Router# debug ephone raw mac-address 0030.94c3.E1A8
```

```

EPHONE raw protocol debugging is enabled for phone 0030.94C3.E1A8

1d05h: skinny socket received 4 bytes on socket [1]
0 0 0 0
1d05h:
1d05h: SkinnyMessageID = 0
1d05h: skinny send 4 bytes
4 0 0 0 0 0 0 0 0 1 0 0
1d05h: socket [1] sent 12 bytes OK (incl hdr) for ephone-(1)

1d06h: skinny socket received 4 bytes on socket [1]
0 0 0 0
1d06h:
1d06h: SkinnyMessageID = 0
1d06h: skinny send 4 bytes
4 0 0 0 0 0 0 0 0 1 0 0
1d06h: socket [1] sent 12 bytes OK (incl hdr) for ephone-(1)

```

Related Commands

Command	Description
debug ephone alarm	Sets SkinnyStation alarm messages debugging for the Cisco IP phone.
debug ephone detail	Sets detail debugging for the Cisco IP phone.
debug ephone error	Sets error debugging for the Cisco IP phone.
debug ephone keepalive	Sets keepalive debugging for the Cisco IP phone.
debug ephone loopback	Sets MWI debugging for the Cisco IP phone.
debug ephone pak	Provides voice packet level debugging and prints the contents of one voice packet in every 1024 voice packets.
debug ephone register	Sets registration debugging for the Cisco IP phone.
debug ephone state	Sets state debugging for the Cisco IP phone.
debug ephone statistics	Sets statistics debugging for the Cisco IP phone.
show debugging	Displays information about the types of debugging that are enabled for your router.

debug ephone register

To set registration debugging for the Cisco IP phone, use the **debug ephone register** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ephone register [**mac-address** *mac-address*]

no debug ephone register [**mac-address** *mac-address*]

Syntax Description

mac-address	(Optional) Defines the MAC address of the Cisco IP phone.
<i>mac-address</i>	(Optional) Specifies the MAC address of the Cisco IP phone.

Defaults

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(5)YD	This command was introduced on the following platforms: Cisco 2600 series and Cisco 3600 series multiservice routers, and Cisco IAD2420 series Integrated Access Devices (IADs).
12.2(2)XT	This command was implemented on the Cisco 1750 and Cisco 1751 multiservice routers.
12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T and implemented on the Cisco 3725 and Cisco 3745 routers.
12.2(8)T1	This command was implemented on the Cisco 2600-XM and Cisco 2691 routers.
12.2(11)T	This command was implemented on the Cisco 1760 routers.

Usage Guidelines

The **debug ephone register** command sets registration debugging for the Cisco IP phones.

If the **mac-address** keyword is not used, the **debug ephone register** command debugs all Cisco IP phones that are registered to the router. You can remove debugging for the Cisco IP phones that you do not want to debug by using the **mac-address** keyword with the **no** form of this command.

You can enable or disable debugging on any number of Cisco IP phones. To see the Cisco IP phones that have debugging enabled, enter the **show ephone** command and look at the debug field in the output. When debugging is enabled for a Cisco IP phone, the debug output is displayed for the directory numbers associated with the Cisco IP phone.

Examples

The following is sample output of registration debugging for the Cisco IP phone with MAC address 0030.94c3.8724:

```
Router# debug ephone register mac-address 0030.94c3.8724
```

```

Ephone registration debugging is enabled

1d06h: New Skinny socket accepted [1] (2 active)
1d06h: sin_family 2, sin_port 50778, in_addr 10.1.0.21
1d06h: skinny_add_socket 1 10.1.0.21 50778
1d06h: ephone-(1)[1] StationRegisterMessage (2/3/12) from 10.1.0.21
1d06h: ephone-(1)[1] Register StationIdentifier DeviceName SEP003094C3E1A8
1d06h: ephone-(1)[1] StationIdentifier Instance 1 deviceType 7
1d06h: ephone-1[-1]:stationIpAddr 10.1.0.21
1d06h: ephone-1[-1]:maxStreams 0
1d06h: ephone-(1) Allow any Skinny Server IP address 10.1.0.6
.
.
.
1d06h: ephone-1[1]:RegisterAck sent to ephone 1: keepalive period 30
.

```

Related Commands

Command	Description
debug ephone alarm	Sets SkinnyStation alarm messages debugging for the Cisco IP phone.
debug ephone detail	Sets detail debugging for the Cisco IP phone.
debug ephone error	Sets error debugging for the Cisco IP phone.
debug ephone keepalive	Sets keepalive debugging for the Cisco IP phone.
debug ephone loopback	Sets MWI debugging for the Cisco IP phone.
debug ephone pak	Provides voice packet level debugging and prints the contents of one voice packet in every 1024 voice packets.
debug ephone raw	Provides raw low-level protocol debugging display for all SCCP messages.
debug ephone state	Sets state debugging for the Cisco IP phone.
debug ephone statistics	Sets statistics debugging for the Cisco IP phone.
show debugging	Displays information about the types of debugging that are enabled for your router.

debug ephone sccp-state

To set debugging for the SCCP call state, use the **debug ephone sccp-state** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ephone sccp-state [**mac-address** *mac-address*]

no debug ephone sccp-state [**mac-address** *mac-address*]

Syntax Description	mac-address (Optional) Specifies the MAC address of a phone. <i>mac-address</i>
---------------------------	-------------------------------------------------------------------------------------------

Command Default	Debugging is not enabled for SCCP state.
------------------------	------------------------------------------

Command Modes	Privileged EXEC
----------------------	-----------------

Command History	Cisco IOS Release	Modification
	12.4(4)XC	This command was introduced.
	12.4(9)T	This command was integrated into Cisco IOS Release 12.4(9)T.

Usage Guidelines	<p>This command is used with Cisco Unified CallManager Express (Cisco Unified CME).</p> <p>This command outputs only the debug messages that correspond to SCCP messages sent to IP phones to indicate the SCCP phone call state, such as RingIn, OffHook, Connected, and OnHook. These debug messages are also included in the output for the debug ephone detail command among other information.</p>
-------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Examples	<p>The following example sets SCCP state debugging for one Cisco Unified CME phone with the MAC address of 678B.AEF9.DAB5.</p>
-----------------	--------------------------------------------------------------------------------------------------------------------------------

```
Router# debug ephone sccp-state mac-address 678B.AEF9.DAB5

EPHONE SCCP state message debugging is enabled
  for ephones 000B.BEF9.DFB5

*Mar  8 06:38:45.863: %ISDN-6-CONNECT: Interface Serial2/0/0:22 is now connected to
4085254871 unknown
*Mar  8 06:38:50.487: ephone-2[13]:SetCallState line 4 DN 60(60) chan 1 ref 100 TsRingIn
*Mar  8 06:38:52.399: ephone-2[13]:SetCallState line 4 DN 60(-1) chan 1 ref 100 TsOffHook
*Mar  8 06:38:52.399: ephone-2[13]:SetCallState line 4 DN 60(-1) chan 1 ref 100
TsConnected
*Mar  8 06:38:58.415: %ISDN-6-CONNECT: Interface Serial2/0/0:22 is now connected to
4085254871 unknown
*Mar  8 06:38:59.963: ephone-2[13]:SetCallState line 4 DN 60(-1) chan 1 ref 100 TsOnHook
*Mar  8 06:38:59.975: %ISDN-6-DISCONNECT: Interface Serial2/0/0:22 disconnected from
4085254871 , call lasted 7 seconds
```

Related Commands

Command	Description
debug ephone detail	Sets detail debugging for one or all Cisco Unified IP phones.

debug ephone state

To set state debugging for the Cisco IP phone, use the **debug ephone state** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ephone state [**mac-address** *mac-address*]

no debug ephone state [**mac-address** *mac-address*]

Syntax Description

mac-address	(Optional) Defines the MAC address of the Cisco IP phone.
<i>mac-address</i>	(Optional) Specifies the MAC address of the Cisco IP phone.

Defaults

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(5)YD	This command was introduced on the following platforms: Cisco 2600 series and Cisco 3600 series multiservice routers, and Cisco IAD2420 series Integrated Access Devices (IADs).
12.2(2)XT	This command was implemented on the Cisco 1750 and Cisco 1751 multiservice routers.
12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T and implemented on the Cisco 3725 and Cisco 3745 routers.
12.2(8)T1	This command was implemented on the Cisco 2600-XM and Cisco 2691 routers.
12.2(11)T	This command was implemented on Cisco 1760 routers.

Usage Guidelines

The **debug ephone state** command sets state debugging for the Cisco IP phones.

If the **mac-address** keyword is not used, the **debug ephone state** command debugs all Cisco IP phones that are registered to the router. You can remove debugging for the Cisco IP phones that you do not want to debug by using the **mac-address** keyword with the **no** form of this command.

You can enable or disable debugging on any number of Cisco IP phones. To see the Cisco IP phones that have debugging enabled, enter the **show ephone** command and look at the debug field in the output.

When debugging is enabled for a Cisco IP phone, the debug output is displayed for the directory numbers associated with the Cisco IP phone.

Examples

The following is sample output of state debugging for the Cisco IP phone with MAC address 0030.94c3.E1A8:

```
Router# debug ephone state mac-address 0030.94c3.E1A8
```

```

EPHONE state debugging is enabled for phone 0030.94C3.E1A8

1d06h: ephone-1[1]:OFFHOOK
1d06h: ephone-1[1]:SIEZE on activeline 0
1d06h: ephone-1[1]:SetCallState line 1 DN 1 TsOffHook
1d06h: ephone-1[1]:Skinny-to-Skinny call DN 1 to DN 2 instance 1
1d06h: ephone-1[1]:SetCallState line 1 DN 1 TsRingOut
1d06h: ephone-1[1]:Call Info DN 1 line 1 ref 158 called 5002 calling 5001
1d06h: ephone-1[1]: Jane calling
1d06h: ephone-1[1]: Jill
1d06h: ephone-1[1]:SetCallState line 3 DN 2 TsRingIn
1d06h: ephone-1[1]:Call Info DN 2 line 3 ref 159 called 5002 calling 5001
1d06h: ephone-1[1]: Jane calling
1d06h: ephone-1[1]: Jill
1d06h: ephone-1[1]:SetCallState line 3 DN 2 TsCallRemoteMultiline
1d06h: ephone-1[1]:SetCallState line 1 DN 1 TsConnected
1d06h: ephone-1[1]:OpenReceive DN 1 codec 4:G711Ulaw64k duration 10 ms bytes 80
1d06h: ephone-1[1]:OpenReceiveChannelAck 1.2.172.21 port=24010
1d06h: ephone-1[1]:StartMedia 1.2.172.22 port=24612
1d06h: DN 1 codec 4:G711Ulaw64k duration 10 ms bytes 80
1d06h: ephone-1[1]:CloseReceive
1d06h: ephone-1[1]:StopMedia
1d06h: ephone-1[1]:SetCallState line 3 DN 2 TsOnHook
1d06h: ephone-1[1]:SetCallState line 1 DN 1 TsOnHook
1d06h: ephone-1[1]:SpeakerPhoneOnHook
1d06h: ephone-1[1]:ONHOOK
1d06h: ephone-1[1]:SpeakerPhoneOnHook
1d06h: SkinnyReportDnState DN 1 ONHOOK

```

Related Commands

Command	Description
debug ephone alarm	Sets SkinnyStation alarm messages debugging for the Cisco IP phone.
debug ephone detail	Sets detail debugging for the Cisco IP phone.
debug ephone error	Sets error debugging for the Cisco IP phone.
debug ephone keepalive	Sets keepalive debugging for the Cisco IP phone.
debug ephone loopback	Sets MWI debugging for the Cisco IP phone.
debug ephone pak	Provides voice packet level debugging and prints the contents of one voice packet in every 1024 voice packets.
debug ephone raw	Provides raw low-level protocol debugging display for all SCCP messages.
debug ephone register	Sets registration debugging for the Cisco IP phone.
debug ephone	Sets statistics debugging for the Cisco IP phone.
show debugging	Displays information about the types of debugging that are enabled for your router.

debug ephone statistics

To set call statistics debugging for the Cisco IP phone, use the **debug ephone statistics** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ephone statistics [**mac-address** *mac-address*]

no debug ephone statistics [**mac-address** *mac-address*]

Syntax Description

mac-address	(Optional) Defines the MAC address of the Cisco IP phone.
<i>mac-address</i>	(Optional) Specifies the MAC address of the Cisco IP phone.

Defaults

No default behavior or values

Command Modes

Privileged EXEC

Command History

Release	Modification
12.1(5)YD	This command was introduced on the following platforms: Cisco 2600 series and Cisco 3600 series multiservice routers, and Cisco IAD2420 series Integrated Access Devices (IADs).
12.2(2)XT	This command was implemented on the Cisco 1750 and Cisco 1751 multiservice routers.
12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T and implemented on the Cisco 3725 and Cisco 3745 routers.
12.2(8)T1	This command was implemented on the Cisco 2600-XM and Cisco 2691 routers.
12.2(11)T	This command was implemented on the Cisco 1760 routers.

Usage Guidelines

The **debug ephone statistics** command provides a debug monitor display of the periodic messages from the Cisco IP phone to the router. These include transmit-and-receive packet counts and an estimate of drop packets. The call statistics can also be displayed for live calls using the **show ephone** command.

If the **mac-address** keyword is not used, the **debug ephone statistics** command debugs all Cisco IP phones that are registered to the router. You can remove debugging for the Cisco IP phones that you do not want to debug by using the **mac-address** keyword with the **no** form of this command.

You can enable or disable debugging on any number of Cisco IP phones. To see the Cisco IP phones that have debugging enabled, enter the **show ephone** command and look at the debug field in the output. When debugging is enabled for a Cisco IP phone, the debug output is displayed for the directory numbers associated with the Cisco IP phone.

Examples

The following is sample output of statistics debugging for the Cisco IP phone with MAC address 0030.94C3.E1A8:

```

Router# debug ephone statistics mac-address 0030.94C3.E1A8

EPHONE statistics debugging is enabled for phone 0030.94C3.E1A8

1d06h: Clear Call Stats for DN 1 call ref 162
1d06h: Clear Call Stats for DN 1 call ref 162
1d06h: Clear Call Stats for DN 1 call ref 162
1d06h: Clear Call Stats for DN 2 call ref 163
1d06h: ephone-1[1]:GetCallStats line 1 ref 162 DN 1: 5001
1d06h: ephone-1[1]:Call Stats for line 1 DN 1 5001 ref 162
1d06h: ephone-1[1]:TX Pkts 0 bytes 0 RX Pkts 0 bytes 0
1d06h: ephone-1[1]:Pkts lost 4504384 jitter 0 latency 0
1d06h: ephone-1[1]:Src 0.0.0.0 0 Dst 0.0.0.0 0 bytes 80 vad 0 G711Ulaw64k
1d06h: ephone-1[1]:GetCallStats line 1 ref 162 DN 1: 5001
1d06h: STATS: DN 1 Packets Sent 0
1d06h: STATS: DN 2 Packets Sent 0
1d06h: ephone-1[1]:Call Stats found DN -1 from Call Ref 162
1d06h: ephone-1[1]:Call Stats for line 0 DN -1 5001 ref 162
1d06h: ephone-1[1]:TX Pkts 275 bytes 25300 RX Pkts 275 bytes 25300
1d06h: ephone-1[1]:Pkts lost 0 jitter 0 latency 0

```

Related Commands

Command	Description
debug ephone alarm	Sets SkinnyStation alarm messages debugging for the Cisco IP phone.
debug ephone detail	Sets detail debugging for the Cisco IP phone.
debug ephone error	Sets error debugging for the Cisco IP phone.
debug ephone keepalive	Sets keepalive debugging for the Cisco IP phone.
debug ephone loopback	Sets MWI debugging for the Cisco IP phone.
debug ephone pak	Provides voice packet level debugging and prints the contents of one voice packet in every 1024 voice packets.
debug ephone raw	Provides raw low-level protocol debugging display for all SCCP messages.
debug ephone register	Sets registration debugging for the Cisco IP phone.
debug ephone state	Sets state debugging for the Cisco IP phone.
show debugging	Displays information about the types of debugging that are enabled for your router.

debug ephone video

To set video debugging for ephones, use the **debug ephone video** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ephone video

no debug ephone video

Syntax Description This command has no arguments or keywords.

Command Default Debugging is disabled for ephone video.

Command Modes Privileged EXEC

Command History	Cisco IOS Release	Modification
	12.4(4)XC	This command was introduced.
	12.4(9)T	This command was integrated into Cisco IOS Release 12.4(9)T.

Usage Guidelines The **debug ephone video** command sets ephone video traces, which provide information about different video states for the call, including video capabilities selection, start, and stop.

The **debug ephone** command debugs all ephones that are registered to the Cisco Unified CallManager Express (Cisco Unified CME) system.

You can enable or disable debugging on any number of ephones. To see the ephones that have debugging enabled, enter the **show ephone** command and look at the debug field in the output. When debugging is enabled for a ephone, the debug output is displayed for the directory numbers associated with the ephone.

Examples The following is sample output for the **debug ephone video** command for ephones:

```
Router# debug ephone video

*Mar 13 16:10:02.703: SkinnyVideoCodecMatch_Caps2Caps: match capability: tx_idxcap = 4,
tx_idxpref = 3,
*Mar 13 16:10:02.703: rx_idxcap = 0, rx_idxpref = 0, videoBitRate = 7040
tx_mpi = 1
*Mar 13 16:10:04.711: ephone-19[1][SEPFFFA00000019]:checkToOpenMultiMedia: dn=19, chan=1
*Mar 13 16:10:04.711: ephone-19[1]:skinnyDP[19].s2s = 0
*Mar 13 16:10:04.711: ephone-19[1]:s2s is not set - hence not video capable
*Mar 13 16:10:04.719: ephone-19[1][SEPFFFA00000019]:SkinnyStartMultiMediaTransmission:
chan 1 dn 19
*Mar 13 16:10:04.723: ephone-19[1]:Accept OLC and open multimedia channel
*Mar 13 16:10:04.723: ephone-19[1][SEPFFFA00000019]:SkinnyOpenMultiMediaReceiveChannel: dn
19 chan 1
```

```

*Mar 13 16:10:04.967: ephone-19[1][SEPFFFA00000019]:fStationOpenReceiveChannelAckMessage:
MEDIA_DN 19 MEDIA_CHAN 1
*Mar 13 16:10:04.967: ephone-19[1]:fStationOpenMultiMediaReceiveChannelAckMessage:
*Mar 13 16:10:04.967: ephone-19[1]:Other_dn == -1
sk3745-2#
*Mar 13 16:10:14.787: ephone-19[1]:SkinnyStopMedia: Stop Multimedia
*Mar 13 16:10:14.787: ephone-19[1][SEPFFFA00000019]:SkinnyCloseMultiMediaReceiveChannel:
passThruPartyID = 0, callReference = 23
*Mar 13 16:10:14.787: ephone-19[1]:SkinnyStopMultiMediaTransmission: line 1 chan 1 dn 19

```

Related Commands

Command	Description
debug ephone alarm	Sets SkinnyStation alarm messages debugging for the ephone.
debug ephone detail	Sets detail debugging for the ephone.
debug ephone error	Sets error debugging for the ephone.
debug ephone message	Sets message debugging for the ephone.
debug ephone mwi	Sets MWI debugging for the ephone.
debug ephone pak	Provides voice packet level debugging and displays the contents of one voice packet in every 1024 voice packets.
debug ephone raw	Provides raw low-level protocol debugging display for all SCCP messages.
debug ephone register	Sets registration debugging for the ephone.
debug ephone state	Sets state debugging for the ephone.
debug ephone statistics	Sets statistics debugging for the ephone.
show debugging	Displays information about the types of debugging that are enabled for your router.
show ephone	Displays information about registered ephones.

debug ephone vm-integration

To display pattern manipulation information used for integration with voice-mail applications, use the **debug ephone vm-integration** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

```
debug ephone vm-integration [mac-address mac-address]
```

```
no debug ephone vm-integration [mac-address mac-address]
```

Syntax Description

mac-address mac-address (Optional) Specifies the MAC address of a Cisco IP phone for debugging.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.3(7)T	This command was introduced.

Usage Guidelines

This command displays the voice-mail integration patterns that were created using the **pattern** commands in vm-integration configuration mode. The patterns are used to forward calls to a voice-mail number that is set with the **voicemail** command.

If you do not specify the **mac-address** keyword, the **debug ephone vm-integration** command debugs all Cisco IP phones that are registered to the router. To remove debugging for Cisco IP phones, enter the **no** form of this command with the **mac-address** keyword.

Examples

The following sample output shows information for the vm-integration tokens that have been defined:

```
Router# debug ephone vm-integration

*Jul 23 15:38:03.294:ephone-3[3]:StimulusMessage 15 (1) From ephone 2
*Jul 23 15:38:03.294:ephone-3[3]:Voicemail access number pattern check
*Jul 23 15:38:03.294:SkinnyGetCallState for DN 3 chan 1 IDLE
*Jul 23 15:38:03.294:called DN -1 chan 1, calling DN -1 chan 1 phone -1 s2s:0
*Jul 23 15:38:03.294:dn number for dn 3 is 19003
*Jul 23 15:38:03.294:Updated number for token 1 is 19003
*Jul 23 15:38:03.294:CDN number for dn 3 is
*Jul 23 15:38:03.294:Updated number for token 2 is
*Jul 23 15:38:03.294:Updated number for token 0 is
*Jul 23 15:38:03.294:Update is 219003*
*Jul 23 15:38:03.294:New Voicemail number is 19101219003*
```

[Table 8](#) describes the significant fields shown in the display.

Table 8 *debug ephone vm-integration Field Descriptions*

Field	Description
token 0	First token that was defined in the pattern.
token 1	Second token that was defined in the pattern.
token 2	Third token that was defined in the pattern.

Related Commands

Command	Description
pattern direct	Configures the DTMF digit pattern forwarding necessary to activate the voice-mail system when a user presses the Messages button on a phone.
pattern ext-to-ext busy	Configures the DTMF digit pattern forwarding necessary to activate the voice-mail system once an internal extension reaches a busy extension and the call is forwarded to voice mail.
pattern ext-to-ext no-answer	Configures the DTMF digit pattern forwarding necessary to activate the voice-mail system once an internal extension fails to connect to an extension and the call is forwarded to voice mail.
pattern trunk-to-ext busy	Configures the DTMF digit pattern forwarding necessary to activate the voice-mail system once an external trunk call reaches a busy extension and the call is forwarded to voice mail.
pattern trunk-to-ext no-answer	Configures the DTMF digit pattern forwarding necessary to activate the voice-mail system when an external trunk call reaches an unanswered extension and the call is forwarded to voice mail.
vm-integration	Enters voice-mail integration configuration mode and enables voice-mail integration with DTMF and analog voice-mail systems.
voicemail	Defines the telephone number that is speed-dialed when the Messages button on a Cisco IP phone is pressed.

debug ephone whisper-intercom

To display debugging messages for the Whisper Intercom feature, use the **debug ephone whisper-intercom** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ephone whisper-intercom

no debug ephone whisper-intercom

Syntax Description This command has no arguments or keywords.

Command Default Debugging for Whisper Intercom is disabled.

Command Modes Privileged EXEC (#)

Command History	Release	Modification
	12.4(22)YB	This command was introduced.
	12.4(24)T	This command was integrated into Cisco IOS Release 12.4(24)

Usage Guidelines This command displays debugging information about the Whisper Intercom feature configured on a directory number of a SCCP phone.

Examples The following example displays output from the **debug ephone whisper-intercom** command:

```
Router# debug ephone whisper-intercom

ephone-1[0] Mac:1111.C1C1.0001 TCP socket:[8] activeLine:0 whisperLine:2 REGISTERED in
SCCP ver 12/12 max_streams=3
mediaActive:0 whisper_mediaActive:0 startMedia:1 offhook:1 ringing:0 reset:0 reset_sent:0
paging 0 debug:0 caps:5
IP:10.6.2.185 9237 7970 keepalive 16 max_line 8
button 1: dn 1 number 2001 CH1 IDLE CH2 IDLE
button 2: dn 161 number 6001 auto dial 6002 CH1 WHISPER
Preferred Codec: g711ulaw
Active Call on DN 161 chan 1 :6001 0.0.0.0 0 to 10.6.2.185 9280 via 10.6.2.185
G711Ulaw64k 160 bytes no vad
Tx Pkts 0 bytes 0 Rx Pkts 0 bytes 0 Lost 0
Jitter 0 Latency 0 callingDn -1 calledDn 162

ephone-2[1] Mac:1111.C1C1.0002 TCP socket:[7] activeLine:0 whisperLine:2 REGISTERED in
SCCP ver 12/12 max_streams=3
mediaActive:0 whisper_mediaActive:1 startMedia:0 offhook:1 ringing:0 reset:0 reset_sent:0
paging 0 debug:0 caps:5
IP:10.6.2.185 9240 7970 keepalive 16 max_line 8
button 1: dn 2 number 2002 CH1 IDLE CH2 IDLE
```

```

button 2: dn 162 number 6002 auto dial 6001 CH1 WHISPER
Preferred Codec: g711ulaw
Active Call on DN 162 chan 1 :6002 10.6.2.185 9280 to 10.6.2.254 2000 via 10.6.2.185
G711Ulaw64k 160 bytes no vad
Tx Pkts 0 bytes 0 Rx Pkts 0 bytes 0 Lost 0
Jitter 0 Latency 0 callingDn 161 calledDn -1

```

Related Commands

Command	Description
show ephone-dn whisper	Displays information about whisper intercom ephone-dns that have been created in Cisco Unified CME.
whisper-intercom	Enables the Whisper Intercom feature on a directory number.



debug mwi relay errors

To debug message waiting indication (MWI) relay errors, use the **debug mwi relay errors** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mwi relay errors

no debug mwi relay errors

Syntax Description This command has no arguments or keywords.

Defaults No default behavior or values

Command Modes Privileged EXEC

Command History	Release	Modification
	12.2(2)XT	This command was introduced on the following platforms: Cisco 1750, Cisco 1751, Cisco 2600 series and Cisco 3600 series multiservice routers; and Cisco IAD2420 series Integrated Access Devices (IADs).
	12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T and implemented on the Cisco 3725 and Cisco 3745 routers.
	12.2(8)T1	This command was implemented on the Cisco 2600-XM and Cisco 2691 routers.
	12.2(11)T	This command was implemented on the Cisco 1760 routers.

Usage Guidelines The **debug mwi relay errors** command provides a debug monitor display of any error messages, when MWI Relay Server (Cisco IOS Telephony Server) is trying to do MWI Relay to extensions on remote Cisco IOS Telephony Service (ITS).

Examples The following examples show errors when MWI Relay Server tries to do an MWI Relay to extension 7004, but location of 7004 is not known to the MWI Relay Server:

```
Router# debug mwi relay errors

mwi-relay error info debugging is on
01:46:48: MWI-APP: mwi_notify_status: No ClientID (7004) registered
```

Related Commands	Command	Description
	debug ephone mwi	Sets MWI debugging for the Cisco IOS Telephony Service router.
	debug mwi relay events	Sets MWI relay events debugging for the Cisco IOS Telephony Service router.

debug mwi relay events

To set message waiting indication (MWI) relay events debugging, use the **debug mwi relay events** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug mwi relay events

no debug mwi relay events

Syntax Description This command has no arguments or keywords.

Defaults No default behavior or values

Command Modes Privileged EXEC

Command History

Release	Modification
12.2(2)XT	This command was introduced on the following platforms: Cisco 1750, Cisco 1751, Cisco 2600 series and Cisco 3600 series multiservice routers; and Cisco IAD2420 series Integrated Access Devices (IADs).
12.2(8)T	This command was integrated into Cisco IOS Release 12.2(8)T and implemented on the Cisco 3725 and Cisco 3745 routers.
12.2(8)T1	This command was implemented on the Cisco 2600-XM and Cisco 2691 routers.
12.2(11)T	This command was implemented on the Cisco 1760 routers.

Usage Guidelines

The **debug mwi relay events** command provides a debug monitor display of events, when MWI Relay Server (Cisco IOS Telephony Server) is trying to do MWI Relay to extensions on remote Cisco IOS Telephony Services (ITS).

Examples

The following debugging messages are shown when the MWI Relay server tries to send MWI Information to remote client 7001 and the location of 7001 is known by the MWI Relay Server:

```
Router# debug mwi relay events

mwi-relay events info debugging is on

01:45:34: mwi_notify_status: Queued event for mwi_app_queue
01:45:34: MWI-APP: mwi_app_process_event:
01:45:34: MWI-APP: mwi_app_process_event: MWI Event for ClientID(7001)@(1.8.17.22)
```

Related Commands	Command	Description
	debug ephone mwi	Sets MWI debugging for the Cisco IOS Telephony Service router.
	debug mwi relay errors	Sets MWI relay errors debugging for the Cisco IOS Telephony Service router.

debug shared-line

To display debugging information about SIP shared lines, use the **debug shared-line** command in privileged EXEC mode. To disable debugging messages, use the **no** form of this command.

```
debug shared-line {all | errors | events | info}
```

```
no debug shared-line {all | errors | events | info}
```

Syntax Description

all	Displays all shared-line debugging messages.
errors	Displays shared-line error messages.
events	Displays shared-line event messages.
info	Displays general information about shared lines.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.4(22)YB	This command was introduced.
12.4(24)T	This command was integrated into Cisco IOS Release 12.4(24)T.

Examples

The following example shows output from the **debug shared-line all** command:

```
Router# debug shared-line all

Aug 21 21:56:56.949: //Shared-Line/EVENT/shrl_validate_newcall_outgoing:Outgoing call
validation request from AFW for user = 20143, usrContainer = 4A7CFBDC
.Aug 21 21:56:56.949: //Shared-Line/INFO/shrl_find_ccb_by_dn:Searching Shared-Line table
for dn '20143'
.Aug 21 21:56:56.949: //Shared-Line/INFO/shrl_find_ccb_by_dn:Entry not found for dn
'20143'
.Aug 21 21:56:56.949: //Shared-Line/INFO/shrl_find_ccb_by_demote_dn:Demoted dn: 20143
.Aug 21 21:56:56.949: //Shared-Line/INFO/shrl_validate_newcall_outgoing:User '20143'
doesn't exist in Shared-Line table
.Aug 21 21:56:56.957: //Shared-Line/EVENT/shrl_validate_newcall_incoming:Incoming call
validation request from AFW for user = 20141
.Aug 21 21:56:56.957: //Shared-Line/INFO/shrl_find_ccb_by_dn:Searching Shared-Line table
for dn '20141'
.Aug 21 21:56:56.957: //Shared-Line/INFO/shrl_find_ccb_by_dn:Entry found [ccb = 4742EAD4]
for dn '20141'
.Aug 21 21:56:56.957: //Shared-Line/INFO/shrl_validate_newcall_incoming:User '20141'
found: ccb = 4742EAD4, mem_count = 2
.Aug 21 21:56:56.957: //Shared-Line/EVENT/shrl_validate_newcall_incoming:Obtained call
instance inst: 0 for incoming call, incoming leg (peer_callid): 5399)
.Aug 21 21:56:56.957: //Shared-Line/INFO/shrl_update_barge_calltype:Updating shared-line
call -1 with calltype = 1
.Aug 21 21:56:56.961: //Shared-Line/INFO/shrl_find_ccb_by_dn:Searching Shared-Line table
for dn '20141'
.Aug 21 21:56:56.961: //Shared-Line/INFO/shrl_find_ccb_by_dn:Entry found [ccb = 4742EAD4]
for dn '20141'
```

```

.Aug 21 21:56:56.961: //Shared-Line/INFO/shrl_find_ccb_by_dn:Searching Shared-Line table
for dn '20141'
.Aug 21 21:56:56.961: //Shared-Line/INFO/shrl_find_ccb_by_dn:Entry found [ccb = 4742EAD4]
for dn '20141'
.Aug 21 21:57:01.689: %IPPHONE-6-REG_ALARM: 24: Name=SEP00141C48E126 Load=8.0(5.0)
Last=Phone-Reg-Rej
.Aug 21 21:57:04.261: //Shared-Line/EVENT/shrl_app_event_notify_handler:Event notification
received: event = 9, callID = 5401, dn = 20141
.Aug 21 21:57:04.261: //Shared-Line/INFO/shrl_find_ccb_by_dn:Searching Shared-Line table
for dn '20141'
.Aug 21 21:57:04.261: //Shared-Line/INFO/shrl_find_ccb_by_dn:Entry found [ccb = 4742EAD4]
for dn '20141'
.Aug 21 21:57:04.261: //Shared-Line/EVENT/shrl_process_connect:called with state = 3,
callID = 5401, peer callID = 5399, dn = 20141, usrContainer = 4A7CACA4
.Aug 21 21:57:04.261: //Shared-Line/INFO/shrl_connect_upd_callinfo:Parsed To:
20141@15.6.0.2, to-tag: 2ed5b927-6ad6
.Aug 21 21:57:04.261: //Shared-Line/INFO/shrl_connect_upd_callinfo:Parsed Contact:
20141@15.6.0.2 for sipCallId: E8583537-6F0211DD-96A69BA1-1228BEFB@15.10.0.1
.Aug 21 21:57:04.261: //Shared-Line/EVENT/shrl_connect_upd_callinfo:Obtained call instance
inst: 0
.Aug 21 21:57:04.261: //Shared-Line/INFO/shrl_connect_upd_callinfo:CONNECT from shared
line for incoming shared-line call.
.Aug 21 21:57:04.261: //Shared-Line/INFO/shrl_find_peer_by_ipaddr:Trying to match peer for
member 20141@15.6.0.2
.Aug 21 21:57:04.261: //Shared-Line/INFO/shrl_find_peer_by_ipaddr:Matching peer [40002]
session target parsed = 15.6.0.2
.Aug 21 21:57:04.261: //Shared-Line/INFO/shrl_connect_upd_callinfo:Matching member found:
20141@15.6.0.2
.Aug 21 21:57:04.261: //Shared-Line/INFO/shrl_update_remote_name:Updating shared-line call
dialog info 5401

.Aug 21 21:57:04.261: //Shared-Line/INFO/shrl_process_connect:Updated callinfo for callid:
5401, member: '20141@15.6.0.2', peer-tag: 40002
.Aug 21 21:57:04.261: //Shared-Line/INFO/shrl_process_connect:Notify remote users about
CALL-CONNECT.
.Aug 21 21:57:04.261: //Shared-Line/EVENT/shrl_send_dialog_notify:Sending NOTIFY to remote
user: 20141@15.6.0.1
.Aug 21 21:57:04.261: //Shared-Line/INFO/shrl_send_dialog_notify:Sending NOTIFY to remote
user: 20141@15.6.0.1 about state 3 on incoming call from 20141@15.6.0.2 privacy OFF
.Aug 21 21:57:04.261: //Shared-Line/INFO/shrl_send_dialog_notify:Dialog msg: dir: 1,
orient: 2, local_tag: 2ed5b927-6ad6, remote_tag: 89DCF0-139B, local_uri: 20141@15.6.0.2,
remote_uri: 20143@15.10.0.1
.Aug 21 21:57:04.261: //Shared-Line/INFO/shrl_send_dialog_notify:Dialog notify sent
successfully
.Aug 21 21:57:04.261: //Shared-Line/INFO/shrl_process_connect:Shared-Line '20141':
Successfully sent notify for callid: 5401
.Aug 21 21:57:04.265: //Shared-Line/INFO/shrl_find_ccb_by_dn:Searching Shared-Line table
for dn '20141'
.Aug 21 21:57:04.265: //Shared-Line/INFO/shrl_find_ccb_by_dn:Entry found [ccb = 4742EAD4]
for dn '20141'
.Aug 21 21:57:04.265: //Shared-Line/INFO/shrl_find_ccb_by_dn:Searching Shared-Line table
for dn '20143'
.Aug 21 21:57:04.265: //Shared-Line/INFO/shrl_find_ccb_by_dn:Entry not found for dn
'20143'
.Aug 21 21:57:04.269: //Shared-Line/INFO/shrl_find_ccb_by_demote_dn:Demoted dn: 20143
.Aug 21 21:57:04.269: //Shared-Line/INFO/shrl_update_totag:Shared-Line not enabled for
'20143'
.Aug 21 21:57:04.269: //Shared-Line/EVENT/shrl_app_event_notify_handler:Event notification
received: event = 21, callID = 5401, dn = 20141
.Aug 21 21:57:04.269: //Shared-Line/INFO/shrl_find_ccb_by_dn:Searching Shared-Line table
for dn '20141'
.Aug 21 21:57:04.269: //Shared-Line/INFO/shrl_find_ccb_by_dn:Entry found [ccb = 4742EAD4]
for dn '20141'

```

```
.Aug 21 21:57:04.269: //Shared-Line/EVENT/shrl_process_callerid_update:called with state =
7, callID = 5401, peer callID = 5399, dn = 20141
.Aug 21 21:57:04.269: //Shared-Line/INFO/shrl_process_callerid_update:Updated callinfo for
callid: 5401, member: '20141@15.6.0.2', peer-tag: 40002
.Aug 21 21:57:04.269: //Shared-Line/EVENT/shrl_is_outbound:Check for shared line call type
callid 5401for user = 20141
.Aug 21 21:57:04.269: //Shared-Line/INFO/shrl_find_ccb_by_dn:Searching Shared-Line table
for dn '20141'
.Aug 21 21:57:04.269: //Shared-Line/INFO/shrl_find_ccb_by_dn:Entry found [ccb = 4742EAD4]
for dn '20141'
.Aug 21 21:57:04.269: //Shared-Line/EVENT/shrl_barge_type:Check for shared line call type
callid 5401for user = 20141
.Aug 21 21:57:04.269: //Shared-Line/INFO/shrl_find_ccb_by_dn:Searching Shared-Line table
for dn '20141'
.Aug 21 21:57:04.269: //Shared-Line/INFO/shrl_find_ccb_by_dn:Entry found [ccb = 4742EAD4]
for dn '20141'
.Aug 21 21:57:04.273: //Shared-Line/INFO/shrl_find_ccb_by_dn:Searching Shared-Line table
for dn '20141'
.Aug 21 21:57:04.273: //Shared-Line/INFO/shrl_find_ccb_by_dn:Entry found [ccb = 4742EAD4]
for dn '20141'
.Aug 21 21:57:04.281: //Shared-Line/EVENT/shrl_notify_done_handler:NOTIFY_DONE received
for subID: 5 respCode: 17
.Aug 21 21:57:04.281: //Shared-Line/INFO/shrl_find_ccb_by_subid:Search ccb for subid: 5
.Aug 21 21:57:04.281: //Shared-Line/INFO/shrl_find_ccb_by_subid:Found the entry ccb:
4742EAD4 member: 20141@15.6.0.1
.Aug 21 21:57:04.281: //Shared-Line/INFO/shrl_free_spi_respinfo:Free ASNL resp info for
subID = 5
```

Related Commands

Command	Description
shared-line	Creates a directory number to be shared by multiple SIP phones.
show shared-line	Displays information about active calls using SIP shared lines.



debug voice register errors

To display debug information on voice register module errors during registration in a Cisco Unified CallManager Express (Cisco Unified CME) or Cisco Unified Session Initiation Protocol (SIP) Survivable Remote Site Telephony (SRST) environment, use the **debug voice register errors** command in privileged EXEC mode. To disable debugging, use the **no** form of the command.

debug voice register errors

no debug voice register errors

Syntax Description This command has no arguments or keywords

Command Default Disabled

Command Modes Privileged EXEC mode

Command History	Cisco IOS Release	Modification
	12.2(15)ZJ	This command was introduced for Cisco SIP SRST 3.0
	12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T for Cisco SIP SRST 3.0.
	12.4(4)T	This command was added to Cisco Unified CME 3.4 and Cisco SIP SRST 3.4.

Usage Guidelines Registration errors include failure to match pools or any internal errors that happen during registration.

Examples

Cisco Unified CME

The following is sample output for this command for a registration request with authentication enabled:

```
...
*May 6 18:07:26.971: VOICE_REG_POOL: Register request for (4901) from (10.5.49.83)
*May 6 18:07:26.971: VOICE_REG_POOL: key(9499C07A000036A3) added to nonce table
*May 6 18:07:26.975: VOICE_REG_POOL: Contact doesn't match any pools
*May 6 18:07:26.975: //4/89D7750A8005/SIP/Error/ccsip_spi_register_incoming_registration:
Registration Authorization failed with authorization header=
...
```

If there are no voice register pools configured for a particular registration request, the message “Contact doesn’t match any pools” is displayed.

When authentication is enabled and if the phone requesting registration cannot be authenticated, the message “Registration Authorization failed with authorization header” is displayed.

Cisco Unified SIP SRST

The following is sample output from this command:

```
Router# debug voice register errors
```

```
*Apr 22 11:52:54.523 PDT: VOICE_REG_POOL: Contact doesn't match any pools
*Apr 22 11:52:54.539 PDT: VOICE_REG_POOL: Register request for (33015) from (10.2.152.39)
*Apr 22 11:52:54.539 PDT: VOICE_REG_POOL: Contact doesn't match any pools.
*Apr 22 11:52:54.559 PDT: VOICE_REG_POOL: Register request for (33017) from (10.2.152.39)
*Apr 22 11:53:04.559 PDT: VOICE_REG_POOL: Maximum registration threshold for pool(3) hit
```

If there are no voice register pools configured for a particular registration request, the message “Contact doesn’t match any pools” is displayed.

If the **max registrations** command is configured, when registration requests reach the maximum limit, the “Maximum registration threshold for pool (x) hit” message is displayed for the particular pool.

Table 9 describes the significant fields shown in the display.

Table 9 *debug voice register errors Field Descriptions*

Field	Description
Contact (doesn’t match any pools)	Contact refers to the location of the SIP devices and the IP address.
key (MAC address)	Unique MAC address of a locally available individual SIP phone used to support a degree of authentication in Cisco Unified CME.
Register request for (telephone number) from (IP address).	The unique key for each registration is the telephone number.
Registration Authorization (failed with authorization header)	Registration Authorization message is displayed when authenticate command is configured in Cisco Unified CME.

Related Commands

Command	Description
debug voice register events	Displays debug information on voice register module events during SIP phone registrations in a Cisco Unified CME or Cisco Unified SIP SRST environment.

debug voice register events

To display debug information on voice register module events during Session Initiation Protocol (SIP) phone registrations in a Cisco Unified CallManager Express (Cisco Unified CME) or Cisco Unified SIP Survivable Remote Site Telephony (SRST) environment, use the **debug voice register events** command in privileged EXEC mode. To disable debugging, use the **no** form of this command.

debug voice register events

no debug voice register events

Syntax Description This command has no arguments or keywords

Command Default Disabled

Command Modes Privileged EXEC mode

Command History	Cisco IOS Release	Modification
	12.2(15)ZJ	This command was introduced for Cisco SIP SRST 3.0
	12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T for Cisco SIP SRST 3.0.
	12.4(4)T	This command was added to Cisco CME 3.4 and Cisco SIP SRST 3.4.

Usage Guidelines Using the **debug voice register events** command should suffice to view registration activity. Registration activity includes matching of pools, registration creation, and automatic creation of dial peers. For more details and error conditions, you can use the **debug voice register errors** command.

Cisco Unified CME

The following example shows output from this command:

```
*May 6 18:07:27.223: VOICE_REG_POOL: Register request for (4901) from (1.5.49.83)
*May 6 18:07:27.223: VOICE_REG_POOL: Contact matches pool 1 number list 1
*May 6 18:07:27.223: VOICE_REG_POOL: key(4901) contact(10.5.49.83) add to contact table
*May 6 18:07:27.223: VOICE_REG_POOL: No entry for (4901) found in contact table
*May 6 18:07:27.223: VOICE_REG_POOL: key(4901) contact(10.5.49.83) added to contact
tableVOICE_REG_POOL pool->tag(1), dn->tag(1), submask(1)
*May 6 18:07:27.223: VOICE_REG_POOL: Creating param container for dial-peer 40001.
*May 6 18:07:27.223: VOICE_REG_POOL: Created dial-peer entry of type 0
*May 6 18:07:27.223: VOICE_REG_POOL: Registration successful for 4901, registration id is
2
...
```

The phone number 4901 associated with voice register pool 1, voice register dn 1, registered successfully. A dynamic normal (type 0) VoIP dial peer has been created for entry 4901. The dial peer can be verified using the **show voice register dial-peers** and **show sip-ua status registrar** commands.

Cisco Unified SIP SRST

The following is sample output from this command:

```
Router# debug voice register events

Apr 22 10:50:21.731 PDT: VOICE_REG_POOL: Contact matches pool 1
Apr 22 10:50:21.731 PDT: VOICE_REG_POOL: key(91011) contact(192.168.0.2) add to contact
table
Apr 22 10:50:21.731 PDT: VOICE_REG_POOL: key(91011) exists in contact table
Apr 22 10:50:21.731 PDT: VOICE_REG_POOL: contact(192.168.0.2) exists in contact table, ref
updated
Apr 22 10:50:21.731 PDT: VOICE_REG_POOL: Created dial-peer entry of type 1
Apr 22 10:50:21.731 PDT: VOICE_REG_POOL: Registration successful for 91011, registration
id is 257
```

The phone number 91011 registered successfully, and *type 1* is reported in the debug, which means that there is a preexisting VoIP dial peer.

```
Apr 22 10:50:38.119 PDT: VOICE_REG_POOL: Register request for (91021) from (192.168.0.3)
Apr 22 10:50:38.119 PDT: VOICE_REG_POOL: Contact matches pool 2
Apr 22 10:50:38.123 PDT: VOICE_REG_POOL: key(91021) contact(192.168.0.3) add to contact
table
Apr 22 10:50:38.123 PDT: VOICE_REG_POOL: key(91021) exists in contact table
Apr 22 10:50:38.123 PDT: VOICE_REG_POOL: contact(192.168.0.3) exists in contact table, ref
updated
Apr 22 10:50:38.123 PDT: VOICE_REG_POOL: Created dial-peer entry of type 1
Apr 22 10:50:38.123 PDT: VOICE_REG_POOL: Registration successful for 91021, registration
id is 258
```

A dynamic VoIP dial peer has been created for entry 91021. The dial peer can be verified using the **show voice register dial-peers** and **show sip-ua status registrar** commands.

```
Apr 22 10:51:08.971 PDT: VOICE_REG_POOL: Register request for (95021) from (10.2.161.50)
Apr 22 10:51:08.971 PDT: VOICE_REG_POOL: Contact matches pool 3
Apr 22 10:51:08.971 PDT: VOICE_REG_POOL: key(95021) contact(10.2.161.50) add to contact
table
Apr 22 10:51:08.971 PDT: VOICE_REG_POOL: No entry for (95021) found in contact table
Apr 22 10:51:08.975 PDT: VOICE_REG_POOL: key(95021) contact(10.2.161.50) added to contact
table
Apr 22 10:51:08.979 PDT: VOICE_REG_POOL: Created dial-peer entry of type 0
Apr 22 10:51:08.979 PDT: VOICE_REG_POOL: Registration successful for 95021, registration
id is 259
Apr 22 10:51:09.019 PDT: VOICE_REG_POOL: Register request for (95012) from (10.2.161.50)
Apr 22 10:51:09.019 PDT: VOICE_REG_POOL: Contact matches pool 3
Apr 22 10:51:09.019 PDT: VOICE_REG_POOL: key(95012) contact(10.2.161.50) add to contact
table
Apr 22 10:51:09.019 PDT: VOICE_REG_POOL: No entry for (95012) found in contact table
Apr 22 10:51:09.023 PDT: VOICE_REG_POOL: key(95012) contact(10.2.161.50) added to contact
table
Apr 22 10:51:09.027 PDT: VOICE_REG_POOL: Created dial-peer entry of type 0
Apr 22 10:51:09.027 PDT: VOICE_REG_POOL: Registration successful for 95012, registration
id is 260
Apr 22 10:51:09.071 PDT: VOICE_REG_POOL: Register request for (95011) from (10.2.161.50)
Apr 22 10:51:09.071 PDT: VOICE_REG_POOL: Contact matches pool 3
Apr 22 10:51:09.071 PDT: VOICE_REG_POOL: key(95011) contact(10.2.161.50) add to contact
table
Apr 22 10:51:09.071 PDT: VOICE_REG_POOL: No entry for (95011) found in contact table
Apr 22 10:51:09.075 PDT: VOICE_REG_POOL: key(95011) contact(10.2.161.50) added to contact
table
Apr 22 10:51:09.079 PDT: VOICE_REG_POOL: Created dial-peer entry of type 0
Apr 22 10:51:09.079 PDT: VOICE_REG_POOL: Registration successful for 95011, registration
id is 261
Apr 22 10:51:09.123 PDT: VOICE_REG_POOL: Register request for (95500) from (10.2.161.50)
Apr 22 10:51:09.123 PDT: VOICE_REG_POOL: Contact matches pool 3
```

```

Apr 22 10:51:09.123 PDT: VOICE_REG_POOL: key(95500) contact(10.2.161.50) add to contact
table
Apr 22 10:51:09.123 PDT: VOICE_REG_POOL: No entry for (95500) found in contact table
Apr 22 10:51:09.127 PDT: VOICE_REG_POOL: key(95500) contact(10.2.161.50) added to contact
table
Apr 22 10:51:09.131 PDT: VOICE_REG_POOL: Created dial-peer entry of type 0
Apr 22 10:51:09.131 PDT: VOICE_REG_POOL: Registration successful for 95500, registration
id is 262
*Apr 22 11:52:54.523 PDT: VOICE_REG_POOL: Contact doesn't match any pools
*Apr 22 11:52:54.539 PDT: VOICE_REG_POOL: Register request for (33015) from (10.2.152.39)
*Apr 22 11:52:54.539 PDT: VOICE_REG_POOL: Contact doesn't match any pools
*Apr 22 11:52:54.559 PDT: VOICE_REG_POOL: Register request for (33017) from (10.2.152.39)

```

Table 10 describes the significant fields shown in the display.

Table 10 *debug voice register events Field Descriptions*

Field	Description
Contact	Indicates the location of the SIP devices and may indicate the IP address.
contact table	The table that maintains the location of the SIP devices.
key	The phone number is used as the unique key to maintain registrations of SIP devices.
multiple contact	More than one registration matches the same phone number.
no entry	The incoming registration was not found.
type 0	Normal dial peer.
type 1	Existing normal dial peer.
type 2	Proxy dial peer.
type 3	Existing proxy dial peer.
type 4	Dial-plan dial peer.
type 5	Existing dial-plan dial peer.
type 6	Alias dial peer.
type 7	Existing alias dial peer.
un-registration successful	The incoming un-register was successful.
Register request/registration id <i>number</i>	The internal unique number for each registration; useful for debugging particular registrations.

Related Commands

Command	Description
debug voice register errors	Displays debug information on voice register module errors during registration in a Cisco Unified CME or Cisco Unified SIP SRST environment.
show sip-ua status registrar	Displays all the SIP endpoints that are currently registered with the contact address.
show voice register dial-peers	Displays details of Cisco Unified SIP SRST configuration and of all dynamically created VoIP dial peers.

