



## Cisco Unified CME Commands: D

---

- [date-format \(telephony-service\)](#), on page 3
- [date-format \(voice register global\)](#), on page 4
- [debug callmonitor](#), on page 5
- [debug capf-server](#), on page 8
- [debug cch323 video](#), on page 10
- [debug credentials](#), on page 12
- [debug cti](#), on page 14
- [debug ctl-client](#), on page 16
- [debug ephone alarm](#), on page 17
- [debug ephone blf](#), on page 19
- [debug ephone ccm-compatible](#), on page 21
- [debug ephone detail](#), on page 23
- [debug ephone error](#), on page 26
- [debug ephone extension-assigner](#), on page 28
- [debug ephone hfs](#), on page 30
- [debug ephone keepalive](#), on page 32
- [debug ephone loopback](#), on page 34
- [debug ephone lpcor](#), on page 39
- [debug ephone message](#), on page 40
- [debug ephone mlpp](#), on page 42
- [debug ephone moh](#), on page 44
- [debug ephone mwi](#), on page 46
- [debug ephone paging](#), on page 48
- [debug ephone pak](#), on page 50
- [debug ephone qov](#), on page 52
- [debug ephone raw](#), on page 54
- [debug ephone register](#), on page 56
- [debug ephone sccp-state](#), on page 58
- [debug ephone shared-line-mixed](#), on page 59
- [debug ephone state](#), on page 62
- [debug ephone statistics](#), on page 64
- [debug ephone video](#), on page 66
- [debug ephone vm-integration](#), on page 68

- [debug ephone whisper-intercom](#), on page 70
- [debug mwi relay errors](#), on page 72
- [debug mwi relay events](#), on page 73
- [debug shared-line](#), on page 74
- [debug voice register errors](#), on page 77
- [debug voice register events](#), on page 79
- [default \(voice hunt-group\)](#), on page 83
- [description \(ephone\)](#), on page 84
- [description \(ephone-dn and ephone-dn-template\)](#), on page 85
- [description \(ephone-hunt\)](#), on page 87
- [description \(voice hunt-group\)](#), on page 88
- [description \(voice moh-group\)](#), on page 89
- [description \(voice register pool\)](#), on page 90
- [description \(voice register pool-type\)](#)[description \(voice register pool-type\)](#), on page 91
- [device-id \(ephone-type\)](#), on page 92
- [device-name](#), on page 94
- [device-security-mode](#), on page 95
- [device-type](#), on page 97
- [dial-peer no-match isdn disconnect-cause](#), on page 99
- [dialplan](#), on page 100
- [dialplan-pattern](#), on page 102
- [dialplan-pattern \(call-manager-fallback\)](#), on page 106
- [dialplan-pattern \(voice register\)](#), on page 109
- [digit collect kpml](#), on page 112
- [direct-inward-dial isdn](#), on page 113
- [directory](#), on page 115
- [directory entry](#), on page 116
- [display-logout](#), on page 118
- [dnd \(voice register pool\)](#), on page 119
- [dnd feature-ring](#), on page 120
- [dnd-control \(voice register template\)](#), on page 122
- [dn-webedit](#), on page 123
- [dst \(voice register global\)](#), on page 124
- [dst auto-adjust \(voice register global\)](#), on page 126
- [dtmf-relay \(voice register pool\)](#), on page 127

## date-format (telephony-service)

To set the date display format on the Cisco IP phones in a Cisco CallManager Express (Cisco CME) system, use the **dateformat** command in telephony-service configuration mode. To display the date in the default format, use the **no** form of this command.

**date-format** {dd-mm-yy | mm-dd-yy | yy-dd-mm | yy-mm-dd}  
**no date-format**

<b>Syntax Description</b>	<b>dd-mm-yy mm-dd-yy yy-dd-mm yy-mm-dd</b> Format in which dates are displayed on the IP phone: <ul style="list-style-type: none"> <li>• <b>dd</b>—Two-digit day.</li> <li>• <b>mm</b>—Two-digit month.</li> <li>• <b>yy</b>—Two-digit year.</li> </ul>
---------------------------	---

**Command Default** Default is **mm-dd-yy**.

**Command Modes** Telephony-service configuration (config-telephony)

<b>Command History</b>	Cisco IOS Release	Cisco Product	Modification
	12.2(2)XT	Cisco ITS 2.0	This command was introduced
	12.2(8)T	Cisco ITS 2.0	This command was integrated into Cisco IOS Release 12.2(8)T.

### Examples

The following example sets the date format to day, month, and year, so that December 17, 2004 is represented as 17-12-04.

```
Router(config)# telephony-service
Router(config-telephony)# date-format dd-mm-yy
```

## date-format (voice register global)

To set the date display format on SIP phones directly connected in Cisco Unified CME, use the **dateformat** command in voice register global configuration mode. To display the date in the default format, use the **no** form of this command.

**date-format** {dd-mm-yy | mm-dd-yy | yy-dd-mm | yy-mm-dd}  
**no date-format**

Syntax Description	
<b>d/m/y m/d/y y-d-m y/d/m y/m/d yy-m-d</b>	Format in which dates are displayed on the SIP IP phone: <ul style="list-style-type: none"> <li>• <b>d</b>—Two-digit date of the month</li> <li>• <b>m</b>—Two-digit month</li> <li>• <b>y</b>—Two-digit year</li> <li>• <b>yy</b>—Four-digit year</li> </ul>

**Command Default** Date is displayed as m/d/y.

**Command Modes** Voice register global configuration (config-register-global)

Command History	Cisco IOS Release	Cisco Product	Modification
	12.4(4)T	Cisco CME 3.4	This command was introduced.

### Examples

The following example shows how to set the date format so that a date such as December 3, 2007 is represented as 2007-12-03. By using the default configuration, this same date appears as 12/03/07.

```
Router(config)# voice register global
Router(config-register-global)# date-format yy-m-d
```

Related Commands	Command	Description
	<b>dst auto-adjust (voice register global)</b>	Enables automatic adjustment of daylight saving time on SIP phones.
	<b>time-format (voice register global)</b>	Selects a 12-hour clock or a 24-hour clock for the time display format on SIP IP phones in Cisco Unified CME.

# 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

<b>all</b>	All call-monitor debugging traces.
<b>core</b>	Core information debugging traces.
<b>detail</b>	Detailed debugging traces.
<b>errors</b>	Call-monitor error debugging traces.
<b>events</b>	Call-monitor event debugging traces.
<b>hwconf</b>	Debugging traces related to hardware configuration.
<b>info</b>	Call-monitor information debugging traces.
<b>xml</b>	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
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_gcinfo_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
<b>callmonitor</b>	Enable call monitoring messaging functionality on a SIP endpoint in a VoIP network.
<b>gcid</b>	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

<b>all</b>	Collect all CAPF information available.
<b>error</b>	Collect only information about CAPF errors.
<b>events</b>	Collect only information about CAPF status events.
<b>messages</b>	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 De
viceType:Phone has registered.
001912: .Jul 21 18:18:05.171: %IPPHONE-6-UNREGISTER_NORMAL: ephone-1:SEP000E325C9A43
IP:1.1.1.127 So
cket: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
<b>debug ephone video</b>	Sets video debugging for the Cisco Unified IP phone.
<b>show call active video</b>	Displays call information for SCCP video calls in progress.
<b>show call history video</b>	Displays call history information for SCCP video calls.
<b>show debugging</b>	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
```

The below table describes the significant fields shown in the display.

**Table 1: 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
<b>credentials</b>	Enters credentials configuration mode to configure a Cisco Unified CME CTL provider certificate or a Cisco Unified SRST router certificate.
<b>ctl-service admin</b>	Specifies a user name and password to authenticate the CTL client during the CTL protocol.
<b>ip source-address (credentials)</b>	Enables the Cisco Unified CME or SRST router to receive messages through the specified IP address and port.
<b>show credentials</b>	Displays the credentials settings on a Cisco Unified CME or SRST router.
<b>show debugging</b>	Displays information about the types of debugging that are enabled for your router.
<b>trustpoint (credentials)</b>	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

<b>all</b>	All CTI debugging traces.
<b>callcontrol</b>	CTI call control debugging traces.
<b>core</b>	Basic call debugging traces.
<b>dmgr</b>	CTI device manager debugging traces.
<b>lm</b>	CTI line monitoring debugging traces.
<b>protoif</b>	CTI protocol interface debugging traces.
<b>session</b>	CTI session debugging traces.
<b>xml</b>	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.
15.1(1)T	This command was integrated into Cisco IOS Release 15.1(1)T.

## 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
<b>show debugging</b>	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**

<b>Syntax Description</b>	This command has no arguments or keywords.
<b>Command Default</b>	Collection of CTL client debug information is disabled.
<b>Command Modes</b>	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

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

## Command Default

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**

<b>Command</b>	<b>Description</b>
<b>debug ephone detail</b>	Sets detail debugging for the Cisco IP phone.
<b>debug ephone error</b>	Sets error debugging for the Cisco IP phone.
<b>debug ephone keepalive</b>	Sets keepalive debugging for the Cisco IP phone.
<b>debug ephone loopback</b>	Sets MWI debugging for the Cisco IP phone.
<b>debug ephone pak</b>	Provides voice packet level debugging and prints the contents of one voice packet in every 1024 voice packets.
<b>debug ephone raw</b>	Provides raw low-level protocol debugging display for all SCCP messages.
<b>debug ephone register</b>	Sets registration debugging for the Cisco IP phone.
<b>debug ephone state</b>	Sets state debugging for the Cisco IP phone.
<b>debug ephone statistics</b>	Sets statistics debugging for the Cisco IP phone.
<b>show debugging</b>	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*]

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

<b>Command Modes</b>	Privileged EXEC
----------------------	-----------------

<b>Command History</b>	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.

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

<b>Examples</b>	The following is sample output from the <b>debug ephone blf</b> 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
<b>blf-speed-dial</b>	Enables BLF monitoring for a speed-dial number on a phone registered to Cisco Unified CME.
<b>presence call-list</b>	Enables BLF monitoring for call lists and directories on phones registered to a Cisco Unified CME router.
<b>show presence global</b>	Displays configuration information about the presence service.
<b>show presence subscription</b>	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*]

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

**Command Modes** Privileged EXEC

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	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
```

**debug ephone ccm-compatible**

```
*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
<b>debug ephone state</b>	Displays call state information.
<b>show debugging</b>	Displays information about the types of debugging that are enabled for your router.
<b>show ephone</b>	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	<b>mac-address</b>	(Optional) Defines the MAC address of the Cisco IP phone.
	<i>mac-address</i>	(Optional) Specifies the MAC address of the Cisco IP phone.

**Command Default** 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
.
.
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
<b>debug ephone alarm</b>	Sets SkinnyStation alarm messages debugging for the Cisco IP phone.
<b>debug ephone error</b>	Sets error debugging for the Cisco IP phone.
<b>debug ephone keepalive</b>	Sets keepalive debugging for the Cisco IP phone.
<b>debug ephone loopback</b>	Sets MWI debugging for the Cisco IP phone.
<b>debug ephone pak</b>	Provides voice packet level debugging and prints the contents of one voice packet in every 1024 voice packets.
<b>debug ephone raw</b>	Provides raw low-level protocol debugging display for all SCCP messages.
<b>debug ephone register</b>	Sets registration debugging for the Cisco IP phone.
<b>debug ephone state</b>	Sets state debugging for the Cisco IP phone.
<b>debug ephone statistics</b>	Sets statistics debugging for the Cisco IP phone.
<b>show debugging</b>	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

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

## Command Default

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
EPHONE error debugging is enabled
socket [2] send ERROR 11
Skinny Socket [2] retry failure
```

**Related Commands**

Command	Description
<b>debug ephone alarm</b>	Sets SkinnyStation alarm messages debugging for the Cisco IP phone.
<b>debug ephone detail</b>	Sets detail debugging for the Cisco IP phone.
<b>debug ephone keepalive</b>	Sets keepalive debugging for the Cisco IP phone.
<b>debug ephone loopback</b>	Sets MWI debugging for the Cisco IP phone.
<b>debug ephone pak</b>	Provides voice packet level debugging and prints the contents of one voice packet in every 1024 voice packets.
<b>debug ephone raw</b>	Provides raw low-level protocol debugging display for all SCCP messages.
<b>debug ephone register</b>	Sets registration debugging for the Cisco IP phone.
<b>debug ephone state</b>	Sets state debugging for the Cisco IP phone.
<b>debug ephone statistics</b>	Sets statistics debugging for the Cisco IP phone.
<b>show debugging</b>	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: http->sig_type (26); CV_VOICE_EFXS
(26)
*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.

```

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

# debug ephone hfs

To collect and display debugging information on the download of IP phone configuration and firmware files using the HTTP File-Fetch Server (HFS) service in a Cisco Unified CME system, use the **debug ephone hfs** command in privileged EXEC mode. To disable collection of debug information, use the **no** form of this command.

**[no] debug ephone hfs**

<b>Syntax Description</b>	This command has no arguments or keywords.
<b>Command Default</b>	There are no debug logs on the console or buffer.
<b>Command Modes</b>	Privileged EXEC (#)

Command History	Release	Modification
	15.2(1)T	This command was introduced.

**Usage Guidelines** Use the **debug ephone hfs** command to troubleshoot an attempt to download Cisco Unified SIP IP phone configuration and firmware files using the HFS service.

## Examples

The following sample display shows a successful file fetch:

```
Router# debug ephone hfs
Jan  5 01:29:00.829: ephone_hfs_util_urlhook:URL Context --->
    svr_port=6970
    rem_port=63881
    is_ssl=0
    req_method=1
    url=/softkeyDefault.xml
Jan  5 01:29:00.833: ephone_hfs_util_urlhook:Found the binding, fn[softkeyDefault.xml],
path[system:/ephone/sipphone/softkeyDefault.xml]
Jan  5 01:29:00.833: ephone_hfs_util_get_action:Get HTTP-url[/softkeyDefault.xml],
fetch_path[system:/ephone/sipphone/softkeyDefault.xml], fetch_from_home[0]
Jan  5 01:29:00.853: HFS SUCCESS !!! fn=system:/ephone/sipphone/softkeyDefault.xml size=4376
upload-time(s.ms)=0.016
```

The following sample display shows an unsuccessful file fetch, where the file is not found:

```
Router# debug ephone hfs
Jan  5 01:43:16.561: ephone_hfs_util_urlhook:URL Context --->
    svr_port=6970
    rem_port=63890
    is_ssl=0
    req_method=1
    url=/softkeyDefault2.xml
Jan  5 01:43:16.561: ephone_hfs_util_urlhook:File not found
```

The table describes the significant fields shown in the display.

**Table 2: debug ephone hfs Field Descriptions**

Field	Description
svr_port	Cisco Unified CME port where the request is sent by the remote Cisco Unified SIP IP phone.
rem_port	Remote port of the Cisco Unified SIP IP phone. The request originates from this port.
is_ssl	Indicates if a secure HTTP connection is established using the Secure Sockets Layer (SSL) method.
req_method	Indicates the type of HTTP request message. A value of 1 is equivalent to HTTP-GET while a value of 2 is equivalent to HTTP-POST.
url	Location of the file to be downloaded.

**Related Commands**

Command	Description
<b>hfs enable</b>	Enables the HFS download service on an IP Phone in a Cisco Unified CME system.

# 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

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

## Command Default

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
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
<b>debug ephone alarm</b>	Sets SkinnyStation alarm messages debugging for the Cisco IP phone.
<b>debug ephone detail</b>	Sets detail debugging for the Cisco IP phone.
<b>debug ephone error</b>	Sets error debugging for the Cisco IP phone.
<b>debug ephone loopback</b>	Sets MWI debugging for the Cisco IP phone.
<b>debug ephone pak</b>	Provides voice packet level debugging and prints the contents of one voice packet in every 1024 voice packets.
<b>debug ephone raw</b>	Provides raw low-level protocol debugging display for all SCCP messages.
<b>debug ephone register</b>	Sets registration debugging for the Cisco IP phone.
<b>debug ephone state</b>	Sets state debugging for the Cisco IP phone.
<b>debug ephone statistics</b>	Sets statistics debugging for the Cisco IP phone.
<b>show debugging</b>	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

<b>mac-address</b> <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
.
.
.
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:TONE OFF
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

```

## debug ephone loopback

```

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][SEP000DBDBEF37D]: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

```

The below table describes the significant fields shown in the display.

**Table 3: debug ephone loopback Field Descriptions**

Field	Description
Called Number	Original called number as presented to the incoming side of the loopback-dn.

Field	Description
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"> <li>• Ephone-dn tag and channel of the far-end call terminator (in this example, ephone-dn 14 is extension 1514).</li> <li>• other—Ephone-dn tag of the outgoing side of the loopback.</li> <li>• lb—Ephone-dn tag of the incoming side of the loopback.</li> <li>• far—Ephone-dn tag and channel of the far-end call originator, or -1 for a nonlocal number.</li> <li>• 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.</li> </ul>
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"> <li>• 16—Normal disconnect.</li> <li>• 17—User busy.</li> <li>• 19—No answer.</li> <li>• 28—Invalid number.</li> </ul>

## Related Commands

Command	Description
<b>debug ephone pak</b>	Provides voice packet level debugging.
<b>loopback-dn</b>	Configures loopback-dn virtual loopback voice ports used to establish demarcation points for VoIP voice calls and supplementary services.

Command	Description
<b>show ephone</b>	Displays information about registered Cisco IP phones.
<b>show ephone-dn loopback</b>	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*]

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

<b>Command Modes</b>	Privileged EXEC (#)
----------------------	---------------------

<b>Command History</b>	Release	Modification
	15.0(1)XA	This command was introduced.
	15.1(1)T	This command was integrated into Cisco IOS Release 15.1(1)T.

<b>Usage Guidelines</b>	Use this command for troubleshooting LPCOR calls to phones in a Cisco Unified CME system.
	If the <b>mac-address</b> 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 <b>mac-address</b> keyword with the <b>no</b> form of this command.

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

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

<b>detail</b>	(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

The **debug ephone message** command enables message tracing between ephones.

The debug ephone command debugs all ephones associated with a Cisco Unified CME router.

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 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
<b>debug ephone alarm</b>	Sets SkinnyStation alarm messages debugging for the ephone.
<b>debug ephone detail</b>	Sets detail debugging for the ephone.
<b>debug ephone error</b>	Sets error debugging for the ephone.
<b>debug ephone mwi</b>	Sets MWI debugging for the ephone.



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

<b>mac-address</b> <i>mac-address</i>	(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
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
<b>debug voice mlpp</b>	Displays debugging information for MLPP service.
<b>mlpp indication</b>	Enables MLPP indication on an SCCP phone or analog FXS port.
<b>mlpp max-precedence</b>	Sets the maximum precedence (priority) level that a phone user can specify when making an MLPP call.
<b>mlpp preemption</b>	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

<b>mac-address</b> <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 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: 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

```

The below table describes the significant fields shown in the display.

**Table 4: 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
<b>moh (telephony-service)</b>	Generates an audio stream from a file for MOH in a Cisco CME system.
<b>multicast moh</b>	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.

**Command Default** 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
	<b>debug ephone alarm</b>	Sets SkinnyStation alarm messages debugging for the Cisco IP phone.
	<b>debug ephone detail</b>	Sets detail debugging for the Cisco IP phone.
	<b>debug ephone error</b>	Sets error debugging for the Cisco IP phone.

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

# debug ephone paging

To collect debugging information on paging for both Cisco Unified SIP IP and Cisco Unified SCCP IP phones, use the **debug ephone paging** command in privileged EXEC mode. To disable debugging, use the **no** form of this command.

**[no] debug ephone paging**

## Syntax Description

This command has no arguments or keywords.

## Command Modes

Privileged EXEC (#)

## Command History

Release	Modification
15.2(2)T	This command was introduced.

## Examples

The following example shows debug messages from the **debug ephone paging** command:

```
*Dec 7 21:53:42.519: Paging-dn 250 sccp count=1 sip count=2
*Dec 7 21:53:42.527: SkinnyBuildPagingList for DN 250
*Dec 7 21:53:42.527: SkinnySetPagingList added DN 251 to list for DN 250
*Dec 7 21:53:42.527: SkinnySetPagingList added DN 252 to list for DN 250
*Dec 7 21:53:42.527: Paging Group List: 251 252 0 0 0 0 0 0 0
*Dec 7 21:53:42.527: SkinnySetupPagingDnMulticast 239.1.1.0 20480 for DN 250
*Dec 7 21:53:42.527: Found paging DN 250 on ephone-2
*Dec 7 21:53:42.527: Added interface GigabitEthernet0/0 to multicast list for DN 250
*Dec 7 21:53:42.527: SkinnyStartPagingPhone 1 for DN 250 with multicast
*Dec 7 21:53:42.527: Found paging DN 250 on pool 1[40001] is_paging=FALSE
*Dec 7 21:53:42.527: SipPagingPhoneReq for pool 1[40001] with multicast start
*Dec 7 21:53:42.527: Found paging DN 250 on pool 2[40003] is_paging=FALSE
*Dec 7 21:53:42.527: SipPagingPhoneReq for pool 2[40003] with multicast start
*Dec 7 21:53:42.531: SkinnyBuildPagingList DN 250 for 1 targets
*Dec 7 21:53:42.531: SkinnyStartPagingMedia for 1 targets for DN 250
*Dec 7 21:53:57.471: SkinnyStopPagingPhone 1 for DN 250 with multicast
*Dec 7 21:53:57.471: SipPagingPhoneReq for pool 1[40001] with multicast stop
*Dec 7 21:53:57.471: SipPagingPhoneReq for pool 2[40003] with multicast stop
```

The following example shows another set of debug messages from the **debug ephone paging** command:

```
*Oct 27 22:39:32.543: Paging-dn 251 sccp count 1 sip count 1
*Oct 27 22:39:32.551: SkinnyBuildPagingList for DN 251
*Oct 27 22:39:32.551: SkinnySetupPagingDnMulticast 239.1.1.1 20480 for DN 251
*Oct 27 22:39:32.551: Found paging DN 251 on ephone-2
*Oct 27 22:39:32.551: Added interface GigabitEthernet0/0 to multicast list for DN 251
*Oct 27 22:39:32.551: SkinnyStartPagingPhone for DN 251 with multicast
*Oct 27 22:39:32.551: Found paging DN 251 on pool 3[40007]
*Oct 27 22:39:32.551: SipPagingPhoneReq for pool 3[40007] with multicast start
*Oct 27 22:39:32.551: SkinnyBuildPagingList DN 251 for 1 targets
*Oct 27 22:39:32.551: SkinnyStartPagingMedia for 1 targets for DN 251
*Oct 27 22:39:38.331: SkinnyStopPagingPhone for DN 251 with multicast
*Oct 27 22:39:38.331: SipPagingPhoneReq for pool 3[40007] with multicast stop
```



**Related Commands**

Command	Description
<b>paging-dn</b>	Creates a paging extension to receive audio pages on a Cisco Unified IP phone in a Cisco Unified CME system.
<b>paging-dn (voice register)</b>	Registers a Cisco Unified SIP IP phone to an ephone-dn paging directory number.

# 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

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

## Command Default

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_encntype 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
pакcopy=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
<b>debug ephone alarm</b>	Sets SkinnyStation alarm messages debugging for the Cisco IP phone.
<b>debug ephone detail</b>	Sets detail debugging for the Cisco IP phone.
<b>debug ephone error</b>	Sets error debugging for the Cisco IP phone.
<b>debug ephone keepalive</b>	Sets keepalive debugging for the Cisco IP phone.
<b>debug ephone loopback</b>	Sets MWI debugging for the Cisco IP phone.
<b>debug ephone raw</b>	Provides raw low-level protocol debugging display for all SCCP messages.
<b>debug ephone register</b>	Sets registration debugging for the Cisco IP phone.
<b>debug ephone state</b>	Sets state debugging for the Cisco IP phone.
<b>debug ephone statistics</b>	Sets statistics debugging for the Cisco IP phone.
<b>show debugging</b>	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

<b>mac-address</b> <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(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)
```

The below table describes the significant fields shown in the display.

Table 5: 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	<p>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.</p> <p><b>Note</b> This value is meaningful only if the call uses a G.711 codec.</p>

## Related Commands

Command	Description
<b>debug ephone alarm</b>	Displays alarm messages for IP phones.
<b>debug ephone statistics</b>	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

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

## Command Default

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
<b>debug ephone alarm</b>	Sets SkinnyStation alarm messages debugging for the Cisco IP phone.
<b>debug ephone detail</b>	Sets detail debugging for the Cisco IP phone.
<b>debug ephone error</b>	Sets error debugging for the Cisco IP phone.
<b>debug ephone keepalive</b>	Sets keepalive debugging for the Cisco IP phone.
<b>debug ephone loopback</b>	Sets MWI debugging for the Cisco IP phone.
<b>debug ephone pak</b>	Provides voice packet level debugging and prints the contents of one voice packet in every 1024 voice packets.
<b>debug ephone register</b>	Sets registration debugging for the Cisco IP phone.
<b>debug ephone state</b>	Sets state debugging for the Cisco IP phone.
<b>debug ephone statistics</b>	Sets statistics debugging for the Cisco IP phone.
<b>show debugging</b>	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

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

## Command Default

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
<b>debug ephone alarm</b>	Sets SkinnyStation alarm messages debugging for the Cisco IP phone.
<b>debug ephone detail</b>	Sets detail debugging for the Cisco IP phone.
<b>debug ephone error</b>	Sets error debugging for the Cisco IP phone.
<b>debug ephone keepalive</b>	Sets keepalive debugging for the Cisco IP phone.
<b>debug ephone loopback</b>	Sets MWI debugging for the Cisco IP phone.
<b>debug ephone pak</b>	Provides voice packet level debugging and prints the contents of one voice packet in every 1024 voice packets.
<b>debug ephone raw</b>	Provides raw low-level protocol debugging display for all SCCP messages.
<b>debug ephone state</b>	Sets state debugging for the Cisco IP phone.
<b>debug ephone statistics</b>	Sets statistics debugging for the Cisco IP phone.
<b>show debugging</b>	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*]

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

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

**Command Modes** Privileged EXEC

<b>Command History</b>	<b>Cisco IOS Release</b>	<b>Modification</b>
	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 (Cisco Unified CME).

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.

## Examples

The following example sets SCCP state debugging for one Cisco Unified CME phone with the MAC address of 678B.AEF9.DAB5.

```
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
```

<b>Related Commands</b>	<b>Command</b>	<b>Description</b>
	<b>debug ephone detail</b>	Sets detail debugging for one or all Cisco Unified IP phones.

# debug ephone shared-line-mixed

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

**[no] debug ephone shared-line-mixed {all | errors | events | info}**

## Syntax Description

<b>all</b>	Displays all mixed shared-line debugging messages.
<b>errors</b>	Displays mixed shared-line error messages.
<b>events</b>	Displays mixed shared-line event messages.
<b>info</b>	Displays general information about mixed shared lines.

## Command Modes

Privileged EXEC (#)

## Command History

Release	Modification
15.2(2)T	This command was introduced.

## Usage Guidelines

Use the **debug ephone shared-line-mixed** command to show the debugging messages for Cisco Unified SCCP IP phone users in the SCCP layer of a mixed shared line.

## Examples

The following is a sample output from the **debug ephone shared-line-mixed** command for an outgoing call:

```
Router# debug ephone shared-line-mixed
Mar  9 20:16:37.571: skinny_notify_shrl_state_change: shrl event 1 sccp_id 0 peer_tag 20014
  callid 53 incoming 0
Mar  9 20:16:37.571: skinny_shrl_get_call_state: dn 14, chan 1 call state 0
Mar  9 20:16:37.571: skinny_shrl_reserve_idle_chan: reserve dn 14, chan 1
Mar  9 20:16:37.571: skinny_notify_shrl_state_change: dn = 14, chan = 1 event = 1
Mar  9 20:16:37.583: skinny_process_shrl_event: event type 1 callid 53 dn 14 chan 1
Mar  9 20:16:37.583: skinny_process_shrl_callproc: dn 14, chan 1, callid 53
Mar  9 20:16:37.583: skinny_update_shrl_call_state: dn 14, chan 1, call state 13
Router#
Router#
Mar  9 20:16:45.151: skinny_notify_shrl_state_change: shrl event 2 sccp_id 112 peer_tag
20014 callid 53 incoming 0
Mar  9 20:16:45.151: skinny_notify_shrl_state_change: dn = 14, chan = 1 event = 2
Mar  9 20:16:45.155: skinny_process_shrl_event: event type 2 callid 53 dn 14 chan 1
Mar  9 20:16:45.155: skinny_update_shrl_remote: incoming 0, remote_number 2509, remote_name
2509
Router#
Router#
Mar  9 20:16:57.775: skinny_notify_shrl_state_change: shrl event 3 sccp_id 112 peer_tag
20014 callid 53 incoming 0
Mar  9 20:16:57.779: skinny_notify_shrl_state_change: dn = 14, chan = 1 event = 3
Mar  9 20:16:57.779: skinny_process_shrl_event: event type 4 callid 53 dn 14 chan 1
Mar  9 20:16:57.779: skinny_update_shrl_call_state: dn 14, chan 1, call state 2
```

The following is a sample output from the **debug ephone shared-line-mixed** command for an incoming call with hold and resume:

```
Router# debug ephone shared-line-mixed
Mar  9 20:17:16.943: skinny_update_shrl_dn_chan: dn 14, chan 1
Mar  9 20:17:19.143: skinny_notify_shrl_state_change: shrl event 2 sccp_id 112 peer_tag
20014 callid 57 incoming 1
Mar  9 20:17:19.143: skinny_notify_shrl_state_change: dn = 14, chan = 1 event = 2
Mar  9 20:17:19.147: skinny_process_shrl_event: event type 2 callid 57 dn 14 chan 1
Mar  9 20:17:19.147: skinny_update_shrl_remote: incoming 1, remote_number 2509, remote_name
2509
Mar  9 20:17:19.155: skinny_shrl_get_call_state: dn 14, chan 1 call state 2
Mar  9 20:17:19.155: skinny_set_shrl_remote_connect: dn 14, chan 1
Mar  9 20:17:19.159: skinny_process_shrl_event: event type 3 callid 0 dn 14 chan 1
Mar  9 20:17:19.159: skinny_update_shrl_call_state: dn 14, chan 1, call state 13
Router#
Mar  9 20:17:24.347: skinny_notify_shrl_state_change: shrl event 4 sccp_id 112 peer_tag
20014 callid 57 incoming 0
Mar  9 20:17:24.347: skinny_notify_shrl_state_change: dn = 14, chan = 1 event = 4
Mar  9 20:17:24.347: skinny_process_shrl_event: event type 5 callid 57 dn 14 chan 1
Mar  9 20:17:24.347: skinny_update_shrl_call_state: dn 14, chan 1, call state 8
Mar  9 20:17:28.307: skinny_shrl_resume_non_active_line: ref 5 line 4
Mar  9 20:17:28.307: skinny_update_shrl_call_state: dn 14, chan 1, call state 2
Mar  9 20:17:28.319: skinny_shrl_resume_non_active_line: fake redial to 2509
Mar  9 20:17:29.127: skinny_shrl_check_remote_resume: resume callid 62 holder callid 57
Mar  9 20:17:29.127: skinny_shrl_check_remote_resume: resume callid 62 holder callid 57
Mar  9 20:17:29.127: skinny_shrl_get_privacy: dn 14, chan 1 phone 2 privacy 0
Mar  9 20:17:29.135: skinny_notify_shrl_state_change: shrl event 3 sccp_id 112 peer_tag
20014 callid 57 incoming 0
Mar  9 20:17:29.135: skinny_notify_shrl_state_change: dn = 14, chan = 1 event = 3
Mar  9 20:17:29.135: skinny_shrl_set_resume_info: dn 14, chan 1
Mar  9 20:17:29.135: skinny_update_shrl_dn_chan: dn 14, chan 1
Mar  9 20:17:29.155: skinny_process_shrl_event: event type 4 callid 57 dn 14 chan 1
Router#
Mar  9 20:17:42.407: skinny_notify_shrl_hold_or_resume_request: dn 14, chan 1, hold 1
Mar  9 20:17:42.411: skinny_shrl_get_privacy: dn 14, chan 1 phone 2 privacy 0
Router#
Mar  9 20:17:46.979: skinny_notify_shrl_state_change: shrl event 1 sccp_id 112 peer_tag
20014 callid 64 incoming 0
Mar  9 20:17:46.979: skinny_notify_shrl_state_change: dn = 14, chan = 1 event = 1
Mar  9 20:17:46.983: skinny_shrl_get_privacy: dn 14, chan 1 phone 2 privacy 0
Mar  9 20:17:46.987: skinny_notify_shrl_state_change: shrl event 2 sccp_id 112 peer_tag
20014 callid 64 incoming 0
Mar  9 20:17:46.987: skinny_notify_shrl_state_change: dn = 14, chan = 1 event = 2
Mar  9 20:17:46.987: skinny_process_shrl_event: event type 1 callid 64 dn 14 chan 1
Mar  9 20:17:46.987: skinny_process_shrl_event: event type 2 callid 64 dn 14 chan 1
Mar  9 20:17:46.999: skinny_set_shrl_remote_connect: dn 14, chan 1
Mar  9 20:17:46.999: skinny_set_shrl_remote_connect: dn 14, chan 1
Mar  9 20:17:47.007: skinny_process_shrl_event: event type 3 callid 0 dn 14 chan 1
Mar  9 20:17:47.007: skinny_update_shrl_call_state: dn 14, chan 1, call state 13
Mar  9 20:17:47.007: skinny_process_shrl_event: event type 3 callid 0 dn 14 chan 1
Router#
Mar  9 20:17:53.795: skinny_notify_shrl_state_change: shrl event 3 sccp_id 112 peer_tag
20014 callid 64 incoming 0
Mar  9 20:17:53.795: skinny_notify_shrl_state_change: dn = 14, chan = 1 event = 3
Mar  9 20:17:53.795: skinny_process_shrl_event: event type 4 callid 64 dn 14 chan 1
Mar  9 20:17:53.795: skinny_update_shrl_call_state: dn 14, chan 1, call state 2
```

## Related Commands

Command	Description
<b>shared-line</b>	Creates a directory number to be shared by multiple Cisco Unified SIP IP phones.

Command	Description
<b>shared-line sip</b>	Adds an ephone-dn as a member of a shared directory number in the database of the Shared-Line Service Module for a mixed shared line between Cisco Unified SIP IP phones and Cisco Unified SCCP IP phones.
<b>show shared-line</b>	Displays information about active calls using SIP shared lines.

# 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

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

## Command Default

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
<b>debug ephone alarm</b>	Sets SkinnyStation alarm messages debugging for the Cisco IP phone.
<b>debug ephone detail</b>	Sets detail debugging for the Cisco IP phone.
<b>debug ephone error</b>	Sets error debugging for the Cisco IP phone.
<b>debug ephone keepalive</b>	Sets keepalive debugging for the Cisco IP phone.
<b>debug ephone loopback</b>	Sets MWI debugging for the Cisco IP phone.
<b>debug ephone pak</b>	Provides voice packet level debugging and prints the contents of one voice packet in every 1024 voice packets.
<b>debug ephone raw</b>	Provides raw low-level protocol debugging display for all SCCP messages.
<b>debug ephone register</b>	Sets registration debugging for the Cisco IP phone.
<b>debug ephone</b>	Sets statistics debugging for the Cisco IP phone.
<b>show debugging</b>	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

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

## Command Default

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
<b>debug ephone alarm</b>	Sets SkinnyStation alarm messages debugging for the Cisco IP phone.
<b>debug ephone detail</b>	Sets detail debugging for the Cisco IP phone.
<b>debug ephone error</b>	Sets error debugging for the Cisco IP phone.
<b>debug ephone keepalive</b>	Sets keepalive debugging for the Cisco IP phone.
<b>debug ephone loopback</b>	Sets MWI debugging for the Cisco IP phone.
<b>debug ephone pak</b>	Provides voice packet level debugging and prints the contents of one voice packet in every 1024 voice packets.
<b>debug ephone raw</b>	Provides raw low-level protocol debugging display for all SCCP messages.
<b>debug ephone register</b>	Sets registration debugging for the Cisco IP phone.
<b>debug ephone state</b>	Sets state debugging for the Cisco IP phone.
<b>show debugging</b>	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**

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

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

<b>Command Modes</b>	Privileged EXEC
----------------------	-----------------

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	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*
```

The below table describes the significant fields shown in the display.

**Table 6: 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**

<b>Command</b>	<b>Description</b>
<b>pattern direct</b>	Configures the DTMF digit pattern forwarding necessary to activate the voice-mail system when a user presses the Messages button on a phone.
<b>pattern ext-to-ext busy</b>	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.
<b>pattern ext-to-ext no-answer</b>	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.
<b>pattern trunk-to-ext busy</b>	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.
<b>pattern trunk-to-ext no-answer</b>	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.
<b>vm-integration</b>	Enters voice-mail integration configuration mode and enables voice-mail integration with DTMF and analog voice-mail systems.
<b>voicemail</b>	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
<b>show ephone-dn whisper</b>	Displays information about whisper intercom ephone-dns that have been created in Cisco Unified CME.
<b>whisper-intercom</b>	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.

**Command Default** No default behavior or values

**Command Modes** Privileged EXEC

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
```

Command	Description
<b>debug ephone mwi</b>	Sets MWI debugging for the Cisco IOS Telephony Service router.
<b>debug mwi relay events</b>	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.

**Command Default** No default behavior or values

**Command Modes** Privileged EXEC

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)
```

Command	Description
<b>debug ephone mwi</b>	Sets MWI debugging for the Cisco IOS Telephony Service router.
<b>debug mwi relay errors</b>	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

<b>all</b>	Displays all shared-line debugging messages.
<b>errors</b>	Displays shared-line error messages.
<b>events</b>	Displays shared-line event messages.
<b>info</b>	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
```

**debug shared-line**

```
.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
<b>shared-line</b>	Creates a directory number to be shared by multiple SIP phones.
<b>show shared-line</b>	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.

The below table describes the significant fields shown in the display.

**Table 7: 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 ( <i>MAC address</i> )	Unique MAC address of a locally available individual SIP phone used to support a degree of authentication in Cisco Unified CME.
Register request for ( <i>telephone number</i> ) from ( <i>IP address</i> ).	The unique key for each registration is the telephone number.
Registration Authorization (failed with authorization header)	Registration Authorization message is displayed when <b>authenticate</b> command is configured in Cisco Unified CME.

#### Related Commands

Command	Description
<b>debug voice register events</b>	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**

<b>Syntax Description</b>	This command has no arguments or keywords
<b>Command Default</b>	Disabled
<b>Command Modes</b>	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)

```

The below table describes the significant fields shown in the display.

**Table 8: 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 unregister 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
<b>debug voice register errors</b>	Displays debug information on voice register module errors during registration in a Cisco Unified CME or Cisco Unified SIP SRST environment.
<b>show sip-ua status registrar</b>	Displays all the SIP endpoints that are currently registered with the contact address.

Command	Description
<b>show voice register dial-peers</b>	Displays details of Cisco Unified SIP SRST configuration and of all dynamically created VoIP dial peers.

# default (voice hunt-group)

To set a command to its defaults values, use the **default** command in voice hunt-group configuration mode.

**default** *default-value*

## Syntax Description

<i>default-value</i>	One of the voice hunt group configuration commands. Valid choices are as follows: <ul style="list-style-type: none"> <li>• hops (Peer or longest-idle voice hunt group only)</li> <li>• preference</li> <li>• timeout</li> </ul>
----------------------	--

## Command Default

There are no default behaviors or values.

## Command Modes

Voice hunt-group configuration (config-voi-hunt-group)

## Command History

Cisco IOS Release	Cisco Product	Modification
12.4(4)T	Cisco CME 3.4	This command was introduced.

## Usage Guidelines

Use this command to configure the default value for a voice hunt group command.

The default command instructs the voice hunt group to use the default value of the specified command whenever the hunt group is called. This has the same effect as using the no form of the specified command, but the default command clearly specifies which commands are using their default values.

To use the default values for more than one command, enter each command on a separate line.

## Examples

The following example shows how to set the default values for two separate voice hunt-group commands:

```
Router(config)# voice hunt-group 4
peer
Router(config-voi-hunt-group)# default hops
Router(config-voi-hunt-group)# default timeout
```

## Related Commands

Command	Description
<b>voice hunt-group</b>	Defines a hunt group for SIP phones in Cisco Unified CME.

# description (ephone)

To provide ephone descriptions for network management systems using an eXtensible Markup Language (XML) query, use the **description** command in ephone configuration mode. To remove a description, use the **no** form of this command.

**description** *string*  
**no application**

## Syntax Description

<i>string</i>	Allows for a maximum of 128 characters, including spaces. There are no character restrictions.
---------------	--

## Command Default

No ephone description is configured.

## Command Modes

Ephone configuration (config-ephone)

## Command History

Cisco IOS Release	Cisco Product	Modification
12.3(11)XL	Cisco CME 3.2.1	This command was introduced
12.3(14)T	Cisco CME 3.3	This command was integrated into Cisco IOS Release 12.3(14)T.

## Usage Guidelines

The descriptions configured with this command will appear neither on phone displays nor in show command output. Instead, they are sent to network management systems, such as CiscoView. Network management systems obtain **description** command data by sending an XML ISgetDevice request to a Cisco CME system. Cisco CME responds by sending ISDevDesc field data to the network management system, which uses the data to perform such tasks as printing descriptions on screen.

## Examples

The following example provides a description for ephone 1:

```
Router(config)# ephone 1
Router(config-ephone) descri
ption S/N:SK09456FPH3, Location:SJ21- 2nd Floor E5-9, User: Smith, John
```

## description (ephone-dn and ephone-dn-template)

To display a custom text-string description in the header bar of all supported Cisco Unified IP phones, use the **description** command in ephone-dn or ephone-dn-template configuration mode. To return to the default, use the **no** form of this command.

**description** *string*  
**no description**

### Syntax Description

<i>string</i>	Alphanumeric characters to be displayed in the header bar of the phone display. If spaces appear in the string, enclose the string in quotation marks. The maximum string length is 40 characters.
<b>Note</b>	Display behavior depends on phone firmware version.

### Command Default

The extension number of the first line on the phone appears in the header bar.

### Command Modes

Ephone-dn configuration (config-ephone)  
 Ephone-dn-template configuration (config-ephone-dn)

### Command History

Cisco IOS Release	Cisco Product	Modification
12.2(11)T	Cisco ITS 2.0.1	This command was introduced.
12.2(11)YT	Cisco ITS 2.1	The number of characters in the string was modified.
12.2(15)T	Cisco ITS 2.1	This command was integrated into Cisco IOS Release 12.2(15)T.
12.4(4)XC	Cisco Unified CME 4.0	This command was made available in ephone-dn-template configuration mode.
12.4(9)T	Cisco Unified CME 4.0	This command in ephone-dn-template configuration mode was integrated into Cisco IOS 12.4(9)T.

### Usage Guidelines

Use this command under the ephone-dn that is associated with the first line button on a Cisco Unified IP phone. This command is typically used to display the entire E.164 telephone number associated with the first line button in the header bar rather than just the extension number, which is the default.

This command is supported by the following IP phones:

- Cisco Unified IP Phone 7940 and 7940G
- Cisco Unified IP Phones 7960 and 7960G
- Cisco Unified IP Phone 7970
- Cisco Unified IP Phone 7971

For Cisco Unified IP Phone 7940s and 7940Gs or Cisco Unified IP Phone 7960s and 7960Gs, the *string* is truncated to 14 characters if the text string is greater than 14 characters.

For Cisco Unified IP Phone 797x, all characters in the *string* appear alternately with time and date, each for 5 seconds.

If you use an ephone-dn template to apply a command to an ephone-dn and you also use the same command in ephone-dn configuration mode for the same ephone-dn, the value that you set in ephone-dn configuration mode has priority.

## Examples

The following example shows how to define a header bar display for a phone on which the first line button is the extension number 50155:

```
Router(config)# ephone-dn 4

Router(config-ephone-dn)# number 50155
Router(config-ephone-dn)# description
888-555-0155
```

The following example shows how to use an ephone-dn template to define a header bar display for a phone on which the first line button is the extension number 50155:

```
Router(config)# ephone-dn-template 3
Router(config-ephone-dn-template)# description
"888 555-0155"
Router(config-ephone-dn-template)# exit
Router(config)# ephone-dn 4

Router(config-ephone-dn)# number 50155
Router(config-ephone-dn)# ephone-dn-template 3
```

## Related Commands

Command	Description
<b>number</b>	Configures a valid number for a Cisco Unified IP phone.

## description (ephone-hunt)

To create a label for an ephone hunt group, use the **description** command in ephone-hunt configuration mode. To return this value to the default, use the **no** form of this command.

**description** *string*  
**no description**

### Syntax Description

<i>string</i>	Character string that identifies a hunt group.
---------------	--

### Command Default

No description exists for the ephone hunt group.

### Command Modes

Ephone-hunt configuration (config-ephone-hunt)

### Command History

Cisco IOS Release	Cisco Product	Modification
12.4(4)XC	Cisco Unified CME 4.0	This command was introduced.
12.4(9)T	Cisco Unified CME 4.0	This command was integrated into Cisco IOS 12.4(9)T.

### Usage Guidelines

This command creates a label to identify the ephone-hunt group. This label helps make the configuration more readable.

### Examples

The following example shows how to identify a hunt group for technical support agents.

```
ephone-hunt 3 peer
pilot 4200
list 1001, 1002, 1003
description Tech Support Hunt Group
hops 3
timeout 7, 10, 15
max-timeout 25
final 4500
```

## description (voice hunt-group)

To specify a description for a voice hunt group, use the **description** command in voice hunt-group configuration mode. To remove the description, use the **no** form of this command.

**description** *description*

**no description** *description*

### Syntax Description

<i>description</i>	Specific description of the hunt group.
--------------------	---

### Command Default

No description for the hunt group.

### Command Modes

Voice hunt-group configuration (config-voice-hunt-group)

### Command History

Release	Modification
15.3(2)T	This command was introduced.

### Examples

The following example shows how to specify a description for voice hunt-group 12 using the **description** command and presents the description in the output of the **do show run** command:

```
Router(config)# voice hunt-group 12
Router (config-voice-hunt-group)# description ?
  LINE  description for this hunt group
Router (config-voice-hunt-group)# description specific huntgroup description
Router (config-voice-hunt-group)# do show run | sec voice hunt-group
voice hunt-group 12 parallel
  timeout 0
description specific huntgroup description
```

Command	Description
<b>voice hunt-group</b>	Enters voice hunt-group configuration mode to create a hunt group for phones in a Cisco Unified CME system.



# description (voice moh-group)

To display a brief description specific to a MOH group, use the **description** command in voice moh-group configuration mode. To remove the description, use the **no** form of this command.

**description** *string*  
**no description**

## Syntax Description

<i>string</i>	An alphanumeric string to add a brief description specific to a MOH group. Maximum length: 80 characters including spaces.
---------------	--

## Command Default

No MOH group description is configured.

## Command Modes

Voice moh-group configuration (config-voice-moh-group)

## Command History

Cisco IOS Release	Cisco Product	Modification
15.0(1)XA	Cisco Unified CME 8.0 Cisco Unified SRST 8.0	This command was introduced
15.1(1)T	Cisco Unified CME 8.0 Cisco Unified SRST 8.0	This command was integrated into Cisco IOS Release 15.1(1)T.

## Usage Guidelines

This command allows you to type a brief text describing a specific voice-moh-group. You can use maximum 80 characters, including spaces to describe a MOH group.

## Examples

The following example provides a description for voice-moh-group1:

```
Router(config)#
Router(config-voice-moh-group)#
Router(config-voice-moh-group) description this is a moh group for sales
```

## Related Commands

Command	Description
<b>voice-moh-group</b>	Enter voice-moh-group configuration mode.
<b>moh</b>	Enables music on hold from a flash audio feed
<b>multicast moh</b>	Enables multicast of the music-on-hold audio stream.
<b>extension-range</b>	Specifies the extension range for a clients calling a voice-moh-group.

# description (voice register pool)

To display a custom description in the header bar of Cisco IP Phone 7940 and 7940G or a Cisco IP Phone 7960 and 7960G, use the **description** command in voice register pool configuration mode. To return to the default, use the **no** form of this command.

**description** *string*  
**no description**

## Syntax Description

<i>string</i>	Allows for a maximum of 128 characters, including spaces. There are no character restrictions.
---------------	--

## Command Default

The extension number of the first line on the phone appears in the header bar.

## Command Modes

Voice register pool configuration (config-register-pool)

## Command History

Cisco IOS Release	Cisco product	Modification
12.4(4)T	Cisco CME 3.4	This command was introduced.

## Usage Guidelines

Use this command to display a customized description in the header bar of a SIP phone instead of the extension number, which is the default. For example, you can display the entire E.164 telephone number associated with the first phone line.

String is truncated to 14 characters in the display of the Cisco IP Phone 7940, Cisco IP Phone 7940G, Cisco IP Phone 7960, and Cisco IP Phone 7960G.

## Examples

The following example shows how to define a header bar display for a SIP phone on which the extension number is 50155:

```
Router(config)# voice register pool 4

Router(config-register-pool)# number 1 50155
Router(config-register-pool)# description
888-555-0155
```

## Related Commands

Command	Description
<b>number (voice register pool)</b>	Configures a valid number for a SIP phone.

# description(voice register pool-type)description(voice register pool-type)

To specify the description string for a new phone model, use the **description** command in voice register pool-type mode. To remove the description string, use the **no** form of this command.

**description** *description*  
**no description** *description*

## Syntax Description

*description string* Specifies description of the phone model.

## Command Default

Description for the phone model is not defined. When the reference-pooltype command is configured, the description of the reference phone is inherited.

## Command Modes

Voice Register Pool-Type Configuration (config-register-pooltype)

## Command History

Cisco IOS Release	Cisco Product	Modification
15.3(3)M	Cisco SIP CME 10.0	This command was introduced.

## Usage Guidelines

Use this command to specify the description string for a new phone model. When you use the **no** form of this command, the inherited properties of the reference phone takes precedence over the default value.

## Example

The following example shows how to specify the description string for a phone model using the **description** command:

```
Router(config)# voice register pool-type 9900
```

```
Router(config-register-pool-type)# description New Cisco SIP Phone 9900
```

## Related Commands

Command	Description
<b>voice register pool-type</b>	Adds a new Cisco Unified SIP IP phone to Cisco Unified CME.

## device-id (ephone-type)

To specify the device ID of a phone type, use the **device-id** command in ephone-type configuration mode. To reset to the default value, use the **no** form of this command.

**device-id** *number*

**no device-id**

### Syntax Description

<i>number</i>	Device ID of the phone type. Range: 1 to 2,147,483,647. Default: 0. See the table below for a list of supported device IDs.
---------------	---

### Command Default

Device ID is 0.

### Command Modes

Ephone-type configuration (config-ephone-type)

### Command History

Cisco IOS Release	Cisco Product	Modification
12.4(15)XZ	Cisco Unified CME 4.3 Cisco Unified SRST 4.3	This command was introduced.
12.4(20)T	Cisco Unified CME 7.0 Cisco Unified SRST 7.0	This command was integrated into Cisco IOS Release 12.4(20)T.

### Usage Guidelines

This command specifies the device ID of the type of phone being added with the ephone-type template. If this command is set to the default value of 0, the ephone-type is invalid.

**Table 9: Supported Values for Ephone-Type Commands**

Supported Device	device-id	device-type	num-buttons	max-presentation
Cisco Unified IP Phone 6901	547	6901	1	1
Cisco Unified IP Phone 6911	548	6911	1	10
Cisco Unified IP Phone 7915 Expansion Module with 12 buttons	227	7915	12	0 (default)
Cisco Unified IP Phone 7915 Expansion Module with 24 buttons	228	7915	24	0
Cisco Unified IP Phone 7916 Expansion Module with 12 buttons	229	7916	12	0
Cisco Unified IP Phone 7916 Expansion Module with 24 buttons	230	7916	24	0
Cisco Unified Wireless IP Phone 7925	484	7925	6	4
Cisco Unified IP Conference Station 7937G	431	7937	1	6
Nokia E61	376	E61	1	1

## Examples

The following example shows the device ID is set to 376 for the Nokia E61 when creating the ephone-type template:

```
Router(config)# ephone-type E61
Router(config-ephone-type)# device-id 376
Router(config-ephone-type)# device-name E61 Mobile Phone
```

## Related Commands

Command	Description
<b>device-name</b>	Assigns a name to a phone type in an ephone-type template.
<b>load</b>	Associates a type of phone with a phone firmware file.
<b>type</b>	Assigns the phone type to a SCCP phone.

# device-name

To assign a name to a phone type in an ephone-type template, use the **device-name** command in ephone-type configuration mode. To remove the name, use the **no** form of this command.

**device-name** *name*

**no device-name**

## Syntax Description

<i>name</i>	String that identifies this phone type. Value is any alphanumeric string up to 32 characters.
-------------	---

## Command Default

No name is assigned to this phone type.

## Command Modes

Ephone-type configuration (config-ephone-type)

## Command History

Cisco IOS Release	Cisco Product	Modification
12.4(15)XZ	Cisco Unified CME 4.3 Cisco Unified SRST 4.3	This command was introduced.
12.4(20)T	Cisco Unified CME 7.0 Cisco Unified SRST 7.0	This command was integrated into Cisco IOS Release 12.4(20)T.

## Usage Guidelines

This command specifies a device name for the type of phone being added with the ephone-type template.

## Examples

The following example shows that the name “E61 Mobile Phone” is assigned to a phone type when creating the ephone-type template:

```
Router(config)# ephone-type E61
Router(config-ephone-type)# device-id 376
Router(config-ephone-type)# device-name E61 Mobile Phone
```

## Related Commands

Command	Description
<b>device-id</b>	Specifies the device ID for a phone type in an ephone-type template.

# device-security-mode

To set the security mode for SCCP signaling for devices communicating with the Cisco Unified CME router globally or per ephone, use the **device-security-mode** command in telephony-service or ephone configuration mode. To return to the default, use the **no** form of this command.

**device-security-mode** {**authenticated** | **none** | **encrypted**}  
**no device-security-mode**

## Syntax Description

<b>authenticated</b>	SCCP signaling between a device and Cisco Unified CME through the secure TLS connection on TCP port 2443.
<b>none</b>	SCCP signaling is not secure.
<b>encrypted</b>	SCCP signaling between a device and Cisco Unified CME through the secure TLS connection on TCP port 2443, and the media uses Secure Real-Time Transport Protocol (SRTP).

## Command Default

Device signaling is not secure.

## Command Modes

Telephony-service configuration (config-telephony)  
 Ephone configuration (config-ephone)

## Command History

Cisco IOS Release	Cisco Product	Modification
12.4(4)XC	Cisco Unified CME 4.0	This command was introduced.
12.4(9)T	Cisco Unified CME 4.0	This command was integrated into Cisco IOS Release 12.4(9)T.
12.4(15)XW	Cisco Unified CME 4.1	The <b>encrypted</b> keyword was added.
12.4(15)XY	Cisco Unified CME 4.2(1)	The <b>encrypted</b> keyword was added.
12.4(15)XZ	Cisco Unified CME 4.3	The <b>encrypted</b> keyword was added.
12.4(20)T	Cisco Unified CME 7.0	This command was integrated into Cisco IOS Release 12.4(20)T.

## Usage Guidelines

Use this command with Cisco Unified CME phone authentication and encryption.

Set the SCCP signaling security mode globally using this command in telephony-service configuration mode or per ephone using this command in ephone configuration mode. If you use both commands, the per-phone setting overrides the global setting.

## Examples

The following example selects secure SCCP signaling for all ephones.

```
Router(config)# telephony-service
Router(config-telephony)# device-security-mode authenticated
```

The following example selects secure SCCP signaling for ephone 28:

```
Router(config)# ephone 28
Router(config-ephone)# button 1:14 2:25
Router(config-ephone)# device-security-mode authenticated
```

The following example selects secure SCCP signaling for all ephones and then disables it for ephone 36:

```
Router(config)# telephony-service
Router(config-telephony)# device-security-mode authentication
Router(config)# ephone 36
Router(config-ephone)# button 1:15 2:16
Router(config-ephone)# device-security-mode none
```

The following example selects encrypted secure SCCP signaling and encryption through SRTP for all ephones:

```
Router(config)# telephony-service
Router(config-telephony)# device-security-mode encrypted
```



# device-type

To specify the phone type, use the **device-type** command in ephone-type configuration mode. To reset to the default value, use the **no** form of this command.

**device-type** *phone-type*  
**no device-type**

## Syntax Description

<i>phone-type</i>	Device type of the phone. See the table for a list of supported device types. Default value is the same value entered with the <b>ephone-type</b> command.
-------------------	--

## Command Default

Device type is the same value that is entered with the **ephone-type** command.

## Command Modes

Ephone-type configuration (config-ephone-type)

## Command History

Cisco IOS Release	Cisco Product	Modification
12.4(15)XZ	Cisco Unified CME 4.3 Cisco Unified SRST 4.3	This command was introduced.
12.4(20)T	Cisco Unified CME 7.0 Cisco Unified SRST 7.0	This command was integrated into Cisco IOS Release 12.4(20)T.

## Usage Guidelines

This command specifies the device type of the phone being added with the ephone-type template. The device type is set to the same value as the **ephone-type** command unless you use this command to change the value.

This command must be set to one of the following supported values.

**Table 10: Supported Values for Ephone-Type Commands**

Supported Device	device-id	device-type	num-buttons	max-presentation
Cisco Unified IP Phone 7915 Expansion Module with 12 buttons	227	7915	12	0 (default)
Cisco Unified IP Phone 7915 Expansion Module with 24 buttons	228	7915	24	0
Cisco Unified IP Phone 7916 Expansion Module with 12 buttons	229	7916	12	0
Cisco Unified IP Phone 7916 Expansion Module with 24 buttons	230	7916	24	0
Cisco Unified IP Conference Station 7937G	431	7937	1	6
Cisco Unified IP Phone 8941	586	8941	4	3
Cisco Unified IP Phone 8945	585	8945	4	3
Nokia E61	376	E61	1	1

## Examples

The following example shows the device type set to 7915 in the ephone-type template for the Cisco Unified IP Phone 7915 Expansion Module with 12 buttons:

```
Router(config)# ephone-type 7915-12 addon
Router(config-ephone-type)# device-id 227
Router(config-ephone-type)# device-name 7915-12
Router(config-ephone-type)# device-type 7915
```

## Related Commands

Command	Description
<b>device-name</b>	Assigns a name to a phone type in an ephone-type template.
<b>ephone-type</b>	Adds a Cisco Unified IP phone type by defining an ephone-type template.
<b>load</b>	Associates a type of phone with a phone firmware file.
<b>type</b>	Assigns the phone type to a SCCP phone.

## dial-peer no-match isdn disconnect-cause

To disconnect the incoming ISDN call when no inbound voice dial peer is matched, use the `dial-peer no-match isdn disconnect-cause` command in global configuration mode. To restore the default incoming call handling behavior, use the `no` form of this command.

**dial-peer no-match isdn disconnect-cause cause-code**  
**no dial-peer no-match isdn disconnect-cause cause-code**

### Syntax Description

<i>cause-code</i>	An ISDN cause code number. Range is from 1 to 188.
-------------------	--

### Command Default

Dial-peer no-match isdn disconnect-cause command is disabled. Incoming ISDN calls are not forced to disconnect if no inbound dial-peer is matched.

### Command Modes

Global configuration

### Command History

Cisco IOS Release	Cisco Product	Modification
15.1(2)T	Cisco Unified CME 8.1	This command was introduced.

### Usage Guidelines

Use this command to disconnect unauthorized ISDN calls when no inbound voice or modem dial peer is matched.

Refer to the ISDN Cause Values table in the Cisco IOS Debug Command Reference, for a list of ISDN cause codes.

### Examples

The following example shows that ISDN cause code 28 has been specified to match inbound voice or modem dial peers:

```
Router# dial-peer no-match disconnect-cause 28
```

### Related Commands

Command	Description
<b>show dial-peer voice</b>	Displays configuration information for dial peers.

# dialplan

To assign a dial plan to a SIP phone, use the **dialplan** command in voice register pool or voice register template configuration mode. To remove the dial plan from the phone, use the **no** form of this command.

**dialplan** *dialplan-tag*  
**no dialplan** *dialplan-tag*

## Syntax Description

<i>dialplan-tag</i>	Number that identifies the dial plan to use for this SIP phone. This is the <i>dialplan-tag</i> argument that was assigned to the dial plan with the <b>voice register dialplan</b> command. Range: 1 to 24.
---------------------	--

## Command Default

No dial plan is assigned to the phone.

## Command Modes

Voice register pool configuration (config-register-pool)  
 Voice register template configuration (config-register-temp)

## Command History

Cisco IOS Release	Cisco Product	Modification
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

You apply a dial plan to a SIP phone with this command after you create the dial plan with the **voice register dialplan** command. When the phone is reset or restarted, the dial plan file specified with this command is loaded to the phone. A phone can use only one dial plan.

A dial plan assigned to a SIP phone has priority over Key Press Markup Language (KPML), which is enabled by default on the phone.

If you use a voice register template to apply a command to a phone and you also use the same command in voice register pool configuration mode for the same phone, the value that you set in voice register pool configuration mode has priority.

After using the **no dialplan** command to remove a dial plan from a phone, use the **restart** command after creating a new configuration profile if the dial plan was defined with the **pattern** command. If the dial plan was defined using a custom XML file with the **filename** command, you must use the **reset** command for the change to take effect.

## Examples

The following example shows that dial plan 5 is assigned to the SIP phone identified by pool 1:

```
Router(config)# voice register pool 1
Router(config-register-pool)# dialplan 5
```

The following example shows that dial plan 5 is assigned to voice register template 10:

```
Router(config)# voice register template 10
Router(config-register-temp)# dialplan 5
```

**Related Commands**

Command	Description
<b>digit collect kpml</b>	Enables KPML digit collection on a SIP phone.
filename	Specifies a custom XML file that contains the dial patterns to use for a SIP dial plan.
<b>pattern</b>	Defines a dial pattern for a SIP dial plan.
<b>show voice register dialplan</b>	Displays all configuration information for a specific SIP dial plan.
<b>show voice register pool</b>	Displays all configuration information associated with a particular voice register pool.
<b>voice register dialplan</b>	Enters voice register dialplan configuration mode to define a dial plan for SIP phones.

# dialplan-pattern

To define a pattern that is used to expand extension numbers in Cisco Unified CME into fully qualified E.164 numbers, use the **dialplan-pattern** command in telephony-service configuration mode. To disable the **dialplan-pattern** command settings, use the **no** form of this command.

**dialplan-pattern** *tag pattern extension-length extension-length* [{**extension-pattern** *extension-pattern* | **no-reg**}] [**demote**]

**no dialplan-pattern** *tag*

## Syntax Description

<i>tag</i>	Identifies this dial-plan pattern. The tag is a number from 1 to 10.
<i>pattern</i>	Dial-plan pattern, such as the area code, the prefix, and the first one or two digits of the extension number, plus wildcard markers or dots (.) for the remainder of the extension number digits.
<b>extension-length</b>	Sets the number of extension digits that will appear as a caller ID.
<i>extension-length</i>	Number of extension digits. The extension length must match the length of extensions for IP phones. Range: 1 to 32.
<b>extension-pattern</b>	(Optional) Sets an extension number's leading digit pattern when it is different from the E.164 telephone number's leading digits as defined in the <i>extension-pattern</i> argument.
<i>extension-pattern</i>	(Optional) Extension number's leading digit pattern. Consists of one or more digits and wildcard markers or dots (.). For example, 5.. would include extension 500 to 599, and 5... would include 5000 to 5999.  The length of the extension pattern must equal the value configured for the <i>extension-length</i> argument.
<b>no-reg</b>	(Optional) Prevents the E.164 numbers in the dial peer from registering with the gatekeeper.
<i>demote</i>	(Optional) Demotes the registered phone if it matches the pattern, extension-length, and extension pattern.

## Command Default

No expansion pattern exists.

## Command Modes

Telephony-service configuration

## Command History

Cisco IOS Release	Cisco Product	Modification
12.1(5)YD	Cisco ITS 1.0	This command was introduced.
12.2(8)T	Cisco ITS 2.0	This command was integrated into Cisco IOS Release 12.2(8)T.
12.2(11)YT	Cisco ITS 2.1	The <b>extension-pattern</b> keyword was added.
12.2(15)T	Cisco ITS 2.1	This command was integrated into Cisco IOS Release 12.2(15)T.

Cisco IOS Release	Cisco Product	Modification
15.1(3)T	Cisco Unified CME 8.5	This command was modified. The demote keyword was added to the dialplan pattern command and the dialplan pattern tag value was increased to 1-10.

### Usage Guidelines

This command creates a pattern for expanding individual abbreviated extension numbers of calling numbers into fully qualified E.164 numbers.

Use this command when configuring a network with multiple Cisco Unified CMEs to ensure that the appropriate calling number, extension or E.164 number, is provided to the target Cisco Unified CME, and appears on the phone display of the called phone. In networks that have a single Cisco Unified CME, this command is not needed.

If multiple dial-plan patterns are defined, the system matches extension numbers against the patterns in sequential order, starting with the lowest numbered dial-plan pattern tag first. Once a pattern matches an extension number, the pattern is used to generate an expanded number. If additional patterns subsequently match the extension number, they are not used.

The **dialplan-pattern** command builds additional dial peers for the expanded numbers it creates. For example, when the ephone-dn with the number 1001 was defined, the following POTS dial peer was automatically created for it:

```
dial-peer voice 20001 pots
 destination-pattern 1001
 voice-port 50/0/2
```

When you define a dial-plan pattern that 1001 will match, such as 40855510..., a second dial peer is created so that calls to both the 1001 and 4085551001 numbers will be completed. In our example, the additional dial peer that is automatically created looks like the following:

```
dial-peer voice 20002 pots
 destination-pattern 4085551001
 voice-port 50/0/2
```

Both numbers are recognized by Cisco Unified CME as being associated with a SCCP phone.

Both dial peers can be seen with the **show telephony-service dial-peer** command.

In networks with multiple routers, you may need to use the **dialplan-pattern** command to expand extensions to E.164 numbers because local extension numbering schemes can overlap each other. Networks with multiple routers have authorities such as gatekeepers that route calls through the network. These authorities require E.164 numbers so that all numbers in the network will be unique. Use the **dialplan-pattern** command to expand extension numbers into unique E.164 numbers for registering with a gatekeeper.

Ephone-dn numbers for the Cisco IP phones must match the number in the *extension-length* argument; otherwise, the extension number cannot be expanded. For example, the following command maps all 3-digit extension numbers to the telephone number 40855501xx, so that extension 111 is expanded but the 4-digit extension 1011 is not.

```
dialplan-pattern 1 40855501.. extension-length 3
```

Using the **dialplan-pattern** command to expand extension numbers can sometimes result in the improper matching of numbers with dial peers. For example, the expanded E.164 number 2035550134 can match dial-peer destination-pattern 203, not 134, which would be the correct destination pattern for the desired extension. If it is necessary for you to use the **dialplan-pattern** command and you know that the expanded

numbers might match destination patterns for other dial peers, you can manually configure the E.164 expanded number for an extension as its secondary number using the **number** command, as shown in the following example:

```
ephone-dn 23
  number 134 secondary 2035550134
```

The pattern created by the **dialplan-pattern** command is also used to enable distinctive ringing for inbound calls. If a calling-party number matches a dial-plan pattern, the call is considered an internal call and has a distinctive ring that identifies the call as internal. Any call with a calling-party number that does not match a dial-plan pattern is considered an external call and has a distinctive ring that is different from the internal ringing.

When the **extension-pattern** keyword and *extension-pattern* argument are used, the leading digits of an extension pattern are stripped and replaced with the corresponding leading digits of the dial plan. For example, the following command maps all 4xx extension numbers to the E.164 number 40855501xx, so that extension 412 corresponds to 4085550112.

```
dialplan-pattern 1 4085550100 extension-length 3 extension-pattern 4..
```

When the **demote** keyword is used, the **dialplan-pattern** command tries to demote the registered phone if it matches the pattern, *extension-length*, and *extension-pattern*.

## Examples

The following example shows how to create dial-plan pattern 1 for extension numbers 5000 to 5099 with a prefix of 408555. If an inbound calling party number (408555044) matches dial-plan pattern 1, the recipient phone will display an extension (5044) as the caller ID and use an internal ringing tone. If an outbound calling party extension number (5044) matches the same dial-plan pattern 1, the calling-party extension will be converted to an E.164 number (408555044). The E.164 calling-party number will appear as the caller ID.

```
Router(config)# telephony-service
Router(config-telephony)# dialplan-pattern 1 4085550.. extension-length 4 extension-pattern
50..
```

In the following example, the **dialplan-pattern** command creates dial-plan pattern 1 for extensions 800 to 899 with the telephone prefix starting with 4085559. As each number in the extension pattern is declared with the **number** command, two POTS dial peers are created. In the example, they are 801 (an internal office number) and 4085579001 (an external number).

```
Router(config)# telephony-service
Router(config-telephony)# dialplan-pattern 1 40855590.. extension-length 3 extension-pattern
8..
```

The following example shows a configuration for two Cisco CME systems. One system uses 50.. and the other uses 60.. for extension numbers. Each is configured with the same two **dialplan-pattern** commands. Calls from the “50..” system to the “60..” system, and vice versa, are treated as internal calls. Calls that go across a H.323 network and calls that go to a PSTN through an ISDN interface on one of the configured Cisco CME routers are represented as E.164.

```
Router(config)# telephony-service
Router(config-telephony)# dialplan-pattern 1 40855550.. extension-length 4 extension-pattern
50..
```



```
Router(config-telephony)# dialplan-pattern 2 51055560.. extension-length 4 extension-pattern 60..
```

**Related Commands**

Command	Description
<b>show telephony-service dial-peer</b>	Displays dial peer information for extensions in a Cisco CME system.

## dialplan-pattern (call-manager-fallback)

To create a global prefix that can be used to expand the extension numbers of inbound and outbound calls into fully qualified E.164 numbers, use the **dialplan-pattern** command in call-manager-fallback configuration mode. To disable the **dialplan-pattern** command settings, use the **no** form of this command.

**dialplan-pattern** *tag pattern extension-length extension-length* [**extension-pattern** *extension-pattern*]  
 [**no-reg**] [**demote**]  
**no dialplan-pattern** *tag* [*pattern extension-length extension-length extension-pattern extension-pattern*]  
 [**no-reg**] [**demote**]

### Syntax Description

<i>tag</i>	Dial-plan string tag used before a ten-digit telephone number. The tag number is from 1 to 10.
<i>pattern</i>	Dial-plan pattern, such as the area code, the prefix, and the first one or two digits of the extension number, plus wildcard markers or dots (.) for the remainder of the extension number digits.
<b>extension-length</b>	Sets the number of extension digits that will appear as a caller ID.
<i>extension-length</i>	The number of extension digits. The extension length must match the setting for IP phones in Cisco Unified CallManager mode. The range is from 1 to 32.
<b>extension-pattern</b>	(Optional) Sets an extension number's leading digit pattern when it is different from the E.164 telephone number's leading digits defined in the <i>pattern</i> variable.
<i>extension-pattern</i>	(Optional) The extension number's leading digit pattern. Consists of one or more digits and wildcard markers or dots (.). For example, 5.. would include extensions 500 to 599; 5... would include extensions 5000 to 5999. The extension pattern must match the setting for IP phones in Cisco Unified CallManager mode.
<b>no-reg</b>	(Optional) Prevents the E.164 numbers in the dial peer from registering with the gatekeeper.
<i>demote</i>	(Optional) Demotes the registered phone if it matches the pattern, extension-length, and extension pattern.

### Command Default

No default behavior or values.

### Command Modes

Call-manager-fallback configuration

### Command History

Cisco IOS Release	Cisco Product	Modification
12.1(5)YD	Cisco SRST 1.0	This command was introduced on the Cisco 2600 series and Cisco 3600 series multiservice routers and on the Cisco IAD2420 series.
12.2(2)XT	Cisco SRST 2.0	This command was implemented on the Cisco 1750 and Cisco 1751 multiservice routers.

Cisco IOS Release	Cisco Product	Modification
12.2(8)T	Cisco SRST 2.0	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	Cisco SRST 2.0	This command was implemented on the Cisco 2600-XM and Cisco 2691 routers.
12.2(11)T	Cisco SRST 2.01	This command was integrated into Cisco IOS Release 12.2(11)T and implemented on the Cisco 1760 routers.
12.2(11)YT	Cisco SRST 2.1	The <b>extension-pattern</b> keyword was added.
15.1(3)T	Cisco Unified SRST 8.5	This command was modified. The demote keyword was added to the dialplan pattern command and the dialplan pattern tag value was increased to 1-10.

### Usage Guidelines

The **dialplan-pattern** command builds additional dial peers. For example, if a hidden POTS dial peer is created, such as the following:

```
Router(config)# dial-peer voice 20001 pots
Router(config-dial-peer)# destination-pattern 1001
Router(config-dial-peer)# voice-port 50/0/2
```

and a dial-plan pattern is created, such as 40855510.., then an additional dial peer will be created that allows calls to both the 1001 and 4085551001 numbers. For example:

```
Router(config)# dial-peer voice 20002 pots
Router(config-dial-peer)# destination-pattern 4085551001
Router(config-dial-peer)# voice-port 50/0/2
```

Both dial peers can be seen with the **show dial-peer voice** command.

The **dialplan-pattern** command also creates a global prefix that can be used by inbound calls (calls to an IP phone in a Cisco Unified SRST system) and outbound calls (calls made from an IP phone in a Cisco Unified SRST system) to expand their extension numbers to fully qualified E.164 numbers.

For inbound calls (calls to an IP phone in a Cisco Unified SRST system) where the calling party number matches the dial-plan pattern, the call is considered a local call and has a distinctive ring that identifies the call as internal. Any calling party number that does not match the dial-plan pattern is considered an external call and has a distinctive ring that is different from the internal ringing.

For outbound calls, the **dialplan-pattern** command converts the calling party's extension number to an E.164 calling party number. Outbound calls that do not use an E.164 number and go through a PRI connection to the PSTN may be rejected by the PRI link as the calling party identifier.

If there are multiple patterns, called-party numbers are checked in numeric order, starting with pattern 1, until a match is found or until the last pattern has been checked. The valid dial-plan pattern with the lowest tag is used as a prefix to all local Cisco IP phones.

When **extension-pattern** *extension-pattern* keyword and argument are used, the leading digits of an extension pattern are stripped and replaced with the corresponding leading digits of the dial plan. For example, the following command maps all extension numbers 4xx to the PSTN number 40855501xx, so that extension 412 corresponds to 4085550112.

**dialplan-pattern (call-manager-fallback)**

```
Router(config)# call-manager-fallback
Router(config-cm-fallback)# dialplan-pattern 1 4085550100 extension-length 3 extension-pattern
4..
```

The number of *extension-pattern* argument characters must match the number set for the *extension-length* argument. For example, if the *extension-length* is 3, the *extension-pattern* can be 8., 1., 51., and so forth.

A dial-plan pattern is required to register the Cisco IP phone lines with a gatekeeper. The **no-reg** keyword provides the option of not registering specific numbers to the gatekeeper so that those numbers can be used for other telephony services.

When the demote keyword is used, the dialplan-pattern command tries to demote the registered phone if it matches the pattern, extension-length, and extension-pattern.

**Examples**

The following example shows how to create dial-plan pattern 1 for extension numbers 5000 to 5099 with a prefix of 408555. If an inbound calling party number (408555044) matches dial-plan pattern 1, the recipient phone will display an extension (5044) as the caller ID and use an internal ringing tone. If an outbound calling party extension number (5044) matches dial-plan pattern 1, the calling party extension will be converted to an E.164 number (4085555044). The E.164 calling party number will appear as the caller ID.

```
Router(config)# call-manager-fallback
Router(config-cm-fallback)# dialplan-pattern 1 40855550.. extension-length 4 extension-pattern
50..
```

In the following example, the **dialplan-pattern** command creates dial-plan pattern 1 for extensions 800 to 899 with the telephone prefix starting with 4085559. As each number in the extension pattern is declared with the **number** command, two POTs dial peers are created. In the example, they are 801 (an internal office number) and 4085559001 (an external number).

```
Router(config)# call-manager-fallback
Router(config-cm-fallback)# dialplan-pattern 1 40855590.. extension-length 3 extension-pattern
8..
```

The following example shows a configuration for two Cisco Unified SRST systems. Each is configured with the same **dialplan-pattern** commands, but one system uses 50.. and the other uses 60.. for extension numbers. Calls from the “50..” system to the “60..” system, and vice versa, are treated as internal calls. Calls that go across an H.323 network and calls that go to a PSTN through an ISDN interface on one of the configured Cisco Unified SRST routers are represented as E.164.

```
Router(config)# call-manager-fallback
Router(config-cm-fallback)# dialplan-pattern 1 40855550.. extension-length 4 extension-pattern
50..
Router(config-cm-fallback)# dialplan-pattern 2 51055560.. extension-length 4 extension-pattern
60..
```

**Related Commands**

Command	Description
<b>call-manager-fallback</b>	Enables Cisco Unified SRST support and enters call-manager-fallback configuration mode.
<b>show dial-peer voice</b>	Displays information for voice dial peers.

## dialplan-pattern (voice register)

To define a pattern that is used to expand extension numbers in Cisco Unified CME into fully qualified E.164 numbers, use the **dialplan-pattern** command in voice register global configuration mode. To disable the **dialplan-pattern** command settings, use the **no** form of this command.

**dialplan-pattern** *tag pattern extension-length extension-length* [{**extension-pattern** *extension-pattern* | **no-reg**}] [**demote**]

**no dialplan-pattern** *tag*

### Syntax Description

<i>tag</i>	Unique number for identifying this dial-plan pattern. Range: 1 to 10.
<i>pattern</i>	Dial-plan pattern to be matched, such as the area code, the prefix, and the first one or two digits of the extension number, plus wildcard markers or dots (.) for the remainder of the extension number digits.
<b>extension-length</b>	Number of extension digits that will appear as a caller ID.
<i>extension-length</i>	Number of digits in an extension.  This variable must match the length of the directory numbers configured for SIP extensions in Cisco Unified CME. Range: 1 to 32.
<b>extension-pattern</b>	(Optional) Leading digit pattern to be configured for an extension when it is different from the leading digit pattern of the E.164 telephone number, as defined in the <i>extension-pattern</i> argument.
<i>extension-pattern</i>	(Optional) Leading digit pattern to be stripped from extension number when expanding an extension to an E.164 telephone number. Consists of one or more digits and wildcard markers or dots (.). For example, 5.. would include extension 500 to 599, and 5... would include 5000 to 5999.  The length of the extension pattern must equal the value configured for the <i>extension-length</i> argument.
<b>no-reg</b>	(Optional) Prevents the E.164 numbers in the dial peer from registering with the gatekeeper.
<i>demote</i>	(Optional) Demotes the registered phone if it matches the pattern, extension-length, and extension pattern.

### Command Default

No expansion pattern exists.

### Command Modes

Voice register global configuration (config-register-global)

### Command History

Cisco IOS Release	Cisco Product	Modification
12.4(4)XC	Cisco Unified CME 4.0	This command was introduced.
12.4(9)T	Cisco Unified CME 4.0	This command was integrated into Cisco IOS 12.4(9)T.

Cisco IOS Release	Cisco Product	Modification
15.1(3)T	Cisco Unified CME 8.5	This command was modified. The demote keyword was added to the dialplan pattern command and the dialplan pattern tag value was increased to 1-10.

### Usage Guidelines

This command creates a pattern for expanding individual abbreviated SIP extension numbers of calling numbers into fully qualified E.164 numbers.

Use this command when configuring a network with multiple Cisco Unified CMEs to ensure that the appropriate calling number, extension or E.164 number, is provided to the target Cisco Unified CME, and appears on the phone display of the called phone. In networks that have a single Cisco Unified CME, this command is not needed.

Up to five dial-plan patterns can be configured. If multiple dial-plan patterns are defined, the system matches extension numbers against the patterns in sequential order, starting with the lowest numbered dial-plan pattern tag first.

Dial peers for directory numbers are automatically created when SIP phones register in Cisco Unified CME. The **dialplan-pattern** command builds a second dial peer for the expanded number because an extension number matches the pattern. Both numbers are recognized by Cisco Unified CME as being associated with a SIP phone.

For example, the following POTS dial peer is automatically created for extension number 1001 when the associated SIP phone registers in Cisco Unified CME:

```
dial-peer voice 20001 pots
 destination-pattern 1001
 voice-port 50/0/2
```

If the extension number (1001) also matches a dial-plan pattern that is configured using the **dialplan-pattern** command, such as 40855510..., a second dial peer is dynamically created so that calls to both the 1001 and 4085551001 numbers can be completed. Based on the dial-plan pattern to be matched, the following additional POTS dial peer is created:

```
dial-peer voice 20002 pots
 destination-pattern 4085551001
 voice-port 50/0/2
```

Using the **no** form of this command will remove the dial peer that was created for the expanded number.

All dial peers can be displayed by using the **show dial-peer voice summary** command. All dial peers for numbers associated to SIP phones only can be displayed by using the **show voice register dial-peers** command. Dial peers created by using the **dialplan-expansion** command cannot be seen in the running configuration.

The value of the extension-length argument must be equal to the length of extension number to be matched, otherwise, the extension number cannot be expanded. For example, the following command maps all 3-digit extension numbers to the telephone number 40855501..., so that extension 111 is expanded but 4-digit extension number 1111 is not.

```
dialplan-pattern 1 40855501.. extension-length 3
```

When the **extension-pattern** keyword and *extension-pattern* argument are configured, the leading digits of the extension pattern variable are stripped away and replaced with the corresponding leading digits of the dial-plan pattern to create the expanded number. For example, the following command maps all 3-digit

extension numbers with the leading digit of “4” to the telephone number 40855501..., so that extension 434 corresponds to 4085550134.

```
dialplan-pattern 1 40855501.. extension-length 3 extension-pattern 4..
```

To apply dialplan-pattern expansion on a per-system basis to individual SIP *redirecting* numbers in a Cisco Unified CME system, including original called and last reroute numbers, use the **call-forward** command.

When the demote keyword is used, the dialplan-pattern command tries to demote the registered phone if it matches the pattern, extension-length, and extension-pattern

## Examples

The following example shows how to create a dialplan-pattern for expanding extension numbers 60xxx to E.164 numbers 510555xxx.

```
Router(config)# voice register global
Router(config-register-global)# dialplan-pattern 1 5105550... extension-length 5
```

The following example is output from the **show dial-peer summary** command displaying information for four dial peers, one each for extensions 60001 and 60002 and, because the dialplan-expansion command was configured to expand 6.... to 408555...., one each for 4085550001 and 4085550002. The latter two dial peers will not appear in the running configuration.

```
Router# show dial-peer summary
```

		AD				PRE	PASS		OUT
TAG	TYPE	MIN	OPER	PREFIX	DEST-PATTERN	FER	THRU	SESS-TARGET	STATT
20010	pots	up	up		60002\$	0			0
20011	pots	up	up		60001\$	0			9
20012	pots	up	up		5105555001\$	0			9
20013	pots	up	up		5105555002\$	0			0

## Related Commands

Command	Description
call-forward (voice register)	Applies dial-plan pattern expansion globally to redirecting number.
show dial-peer summary	Displays all dial peers created in Cisco Unified CME.
show voice register dial-peer	Displays dial-peer information for SIP extensions in Cisco Unified CME.

# digit collect kpml

To enable Key Press Markup Language (KPML) digit collection on a SIP phone, use the **digit collect kpml** command in voice register pool or voice register template configuration mode. To disable KPML, use the **no** form of this command.

**digit collect kpml**  
**no digit collect kpml**

**Syntax Description** This command has no arguments or keywords.

**Command Default** KPML digit collection is enabled.

**Command Modes** Voice register pool configuration (config-register-pool)  
 Voice register template configuration (config-register-temp)

Release	Cisco Product	Modification
12.4(11)XJ	Cisco Unified CME 4.1 Cisco Unified SRST 4.1	This command was introduced.
12.4(15)T	Cisco Unified CME 4.1 Cisco Unified SRST 4.1	This command was integrated into Cisco IOS Release 12.4(15)T.

**Usage Guidelines** KPML is enabled by default for all directory numbers on the phone. A dial plan assigned to a phone has priority over KPML. Use the **no digit collect kpml** command to disable KPML on a phone.

If you use a voice register template to apply a command to a phone and you also use the same command in voice register pool configuration mode for the same phone, the value that you set in voice register pool configuration mode has priority.

KPML is not supported on the Cisco Unified IP Phone 7905, 7912, 7940, or 7960.

**Examples** The following example shows KPML enabled on SIP phone 4:

```
Router(config)# voice register pool 4
Router(config-register-pool)# digit collect kpml
```

Command	Description
<b>dialplan</b>	Assigns a dial plan to a SIP phone.
<b>show voice register pool</b>	Displays all configuration information associated with a SIP phone.
<b>voice register dialplan</b>	Enters voice register dialplan configuration mode to define a dial plan for SIP phones.



# direct-inward-dial isdn

To enable incoming ISDN enbloc dialing calls, use the `direct-inward-dial isdn` command in voice service voip mode. To disable incoming ISDN enbloc dialing calls use the `no` form of the command.

**direct-inward-dial isdn**  
**no direct-inward-dial isdn**

## Syntax Description

This command has no arguments or keywords.

## Command Default

The direct inward dial isdn is command is enabled.

## Command Modes

voice service pots

## Command History

Cisco IOS Release	Cisco Product	Modification
15.1(2)T	Cisco Unified CME 8.1	This command was introduced.

## Usage Guidelines

Use the `direct-inward-dial-isdn` command to enable the direct-inward-dial (DID) call treatment for an incoming ISDN call. When this feature is enabled, the incoming ISDN call is treated as if the digits were received from the DID trunk. The called number is used to select the outgoing dial peer. No dial tone is presented to the caller to collect dialed digits even if “no direct-inward-dial” of the selected inbound dial-peer is defined for an incoming ISDN call.

Use the `no` form of this command to turn off the global direct-inward-dial setting for incoming ISDN calls. When this command line is disabled, the “direct-inward-dial” setting of a selected inbound dial-peer is used to handle the incoming ISDN calls.'

## Examples

The following is a sample output from this command displaying DID enabled for ISDN:

```
!  
voice service voip  
  ip address trusted list  
    ipv4 172.19.245.1  
    ipv4 172.19.247.1  
    ipv4 172.19.243.1  
    ipv4 171.19.245.1  
    ipv4 171.19.10.1  
  allow-connections h323 to h323  
  allow-connections h323 to sip  
  allow-connections sip to h323  
  allow-connections sip to sip  
  supplementary-service media-renegotiate  
  sip  
    registrar server expires max 120 min 120  
!  
!  
dial-peer voice 1 voip  
  destination-pattern 5511...  
  session protocol sipv2  
  session target ipv4:1.3.45.1  
  incoming called-number 5522...  
  direct-inward-dial
```

...

---

**Related Commands**

Command	Description
<b>voice service</b>	Enters voice service configuration mode.

# directory

To define the order in which the names of Cisco IP phone users are displayed in the local directory, use the **directory** command in telephony-service configuration mode. To return to the default, use the **no** form of this command.

**directory** {**first-name-first** | **last-name-first**}  
**no directory** {**first-name-first** | **last-name-first**}

## Syntax Description

<b>first-name-first</b>	First name is entered first in the Cisco IP phone directory name field.
<b>last-name-first</b>	Last name is entered first in the Cisco IP phone directory name field.

## Command Default

Default is **first-name-first**.

## Command Modes

Telephony-service configuration (config-telephony)

## Command History

Cisco IOS Release	Cisco Product	Modification
12.2(2)XT	Cisco ITS 2.0	This command was introduced.
12.2(8)T	Cisco ITS 2.0	This command was integrated into Cisco IOS Release 12.2(8)T.

## Usage Guidelines

This command defines name order in the local directory. The directory itself is generated from entries made using the **name** command and the **number** command in ephone-dn configuration mode.



### Note

The name information must be entered in the correct order in the **name** command.

The location for the file that is accessed when the Directories button is pressed is specified in the **url** (telephony-service) command.

## Examples

The following example shows how to configure the local directory with the last name first:

```
Router(config)# telephony-service
Router(config-telephony)# directory last-name-first
```

## Related Commands

Command	Description
<b>name</b>	Specifies a name to be associated with an extension (ephone-dn).
<b>number</b>	Specifies a telephone number to be associated with an extension (ephone-dn).
<b>url</b>	Provisions URLs for the displays associated with buttons on Cisco IP phones.

# directory entry

To add a system-wide phone directory and speed-dial definition, use the **directory entry** command in telephony-service configuration mode. To remove a definition, use the **no** form of this command.

**directory entry** {*directory-tag* *number* **name** *name* | **clear**}

**no directory entry** {*directory-tag* | **clear**}

## Syntax Description

<i>directory-tag</i>	Digit string that provides a unique identifier for this entry. Range: 1 to 250.
<i>number</i>	String of up to 32 digits that provides the full telephone number for this entry.
<b>name</b> <i>name</i>	String of up to 24 alphanumeric characters, including spaces. Cannot include opening or closing quotation marks (‘ ’, “ ”, or ”).
<b>clear</b>	Removes all directory entries that were made with this command.

## Command Default

Entries do not exist.

## Command Modes

Telephony-service configuration (config-telephony)

## Command History

Cisco IOS Release	Cisco Product	Modification
12.2(15)ZJ	Cisco CME 3.0	This command was introduced.
12.3(4)T	Cisco CME 3.0	This command was integrated into Cisco IOS Release 12.3(4)T.
12.3(11)XL	Cisco CME 3.2.1	This feature was modified to enable systemwide speed-dialing of entries from 34 to 99.
12.3(14)T	Cisco CME 3.3	This command was integrated into Cisco IOS Release 12.3(14)T.
12.4(15)XZ	Cisco Unified CME 4.3	The maximum number of directory entries was increased from 100 to 250.
12.4(20)T	Cisco Unified CME 7.0	This command was integrated into Cisco IOS Release 12.4(20)T.

## Usage Guidelines

Cisco Unified CME automatically creates a local phone directory consisting of the telephone numbers and names that are entered during ephone-dn configuration. Additional directory entries can be made by administrators using the **directory entry** command. Phone number directory listings are displayed in the order in which they are entered.

A single entry can be removed using the **no directory entry directory-tag** command.

Directory entries that have directory-tag numbers from 34 to 99 also can be used as system-wide speed-dial numbers. That is, if you have the following definition for the headquarters office, any phone user can speed-dial the number:

```
Router(config)# telephony-service
```

```
Router(config-telephony)# directory entry 51 4085550123 name Headquarters
```

Analog phone users press the asterisk (\*) key and the speed-dial identifier (tag number) to dial a speed-dial number.

IP phone users follow this procedure to dial a speed-dial number:

1. With the phone on-hook, an IP phone user presses a two-digit speed-dial code (that is, 05 for the entry with tag 5). A new soft key, Abbr, appears in the phone display.
2. The phone user picks up the phone handset and presses the Abbr soft key. The full telephone number associated with the speed-dial tag is dialed.

## Examples

The following example adds six telephone listings to the local directory. The last two entries, with the identifiers 50 and 51, can be speed-dialed by anyone on the system because their identifiers (directory-tags) are between 34 and 99.

```
Router(config)# telephony-service  
Router(config-telephony)# directory entry 1 4045550110 name Atlanta  
Router(config-telephony)# directory entry 2 3125550120 name Chicago  
Router(config-telephony)# directory entry 4 2125550140 name New York City  
Router(config-telephony)# directory entry 5 2065550150 name Seattle  
Router(config-telephony)# directory entry 50 4085550123 name Corp Headquarters  
Router(config-telephony)# directory entry 51 4085550145 name Division Headquarters
```

## Related Commands

Command	Description
<b>show telephony-service directory-entry</b>	Displays the configured directory entries.
<b>url directories</b>	Provisions the directory URL to select an external directory resource and disables the Cisco Unified CME local directory service.

# display-logout

To specify a message to display on phones in an ephone hunt group when all phones in the hunt group are logged out, use the **display-logout** command in ephone-hunt configuration mode. To return this value to the default, use the **no** form of this command.

**display-logout** *string*  
**no display-logout**

## Syntax Description

<i>string</i>	Character string to be displayed on hunt group member IP phones when all members are logged out.
---------------	--

## Command Default

No logout message exists.

## Command Modes

Ephone-hunt configuration (config-ephone-hunt)

## Command History

Cisco IOS Release	Cisco Product	Modification
12.4(4)XC	Cisco Unified CME 4.0	This command was introduced.
12.4(9)T	Cisco Unified CME 4.0	This command was integrated into Cisco IOS 12.4(9)T.

## Usage Guidelines

This command defines a plain-text message that displays on phones with ephone-dns that are members of a hunt group when all the members of the group are logged out. The message can be used to notify agents that no agents are available to take hunt group calls. It can also be used to tell agents about the disposition of any incoming calls to the hunt group when no agents are available to answer calls. For example, you could set the display to read “All Agents Unavailable,” or “Hunt Group Voice Mail” or “Hunt Group Night Service.”

## Examples

The following example specifies a message to display when all agents are logged out of hunt group 3.

```
ephone-hunt 3 peer
pilot 4200
list 1001, 1002, 1003
display-logout All Agents Logged Out
hops 3
timeout 7, 10, 15
max-timeout 25
final 4500
```

## dnd (voice register pool)

To enable the Do-Not-Disturb (DND) feature, use the **dnd-control** command in voice register pool configuration mode. To disable the DND, use the **no** form of this command.

**dnd**  
**no dnd**

<b>Syntax Description</b>	This command has no arguments or keywords.
---------------------------	--

<b>Command Default</b>	DND is disabled
------------------------	-----------------

<b>Command Modes</b>	Voice register pool configuration (config-register-pool)
----------------------	--

Command History	Cisco IOS Release	Cisco Product	Modification
	12.4(4)T	Cisco CME 3.4	This command was introduced.

### Examples

The following example shows how to enable DND:

```
Router(config)# voice register pool 1
Router(config-register-pool)# dnd
```

Related Commands	Command	Description
	<b>dnd-control (voice register template)</b>	Enables DND soft key in template to be assigned to SIP phones in Cisco Unified CME.

## dnd feature-ring

To disable ringing on phone buttons configured for feature ring when the phone is in do-not-disturb (DND) mode, use the **dnd feature-ring** command in ephone configuration mode. To allow lines configured for feature ring to ring when the phone is in DND mode, use the **no** form of this command.

**dnd feature-ring**

**no dnd feature-ring**

### Syntax Description

This command has no arguments or keywords.

### Command Default

Incoming calls to buttons configured for feature ring do not ring in DND mode.

### Command Modes

Ephone configuration (config-ephone)

### Command History

Cisco IOS Release	Cisco Product	Modification
12.3(11)XL	Cisco CME 3.2.1	This command was introduced.
12.3(14)T	Cisco CME 3.3	This command was integrated into Cisco IOS Release 12.3(14)T.

### Usage Guidelines

This command applies only to phone lines that are configured for the feature-ring option with the **button f** command.

Note that the affirmative form of the command is enabled by default and feature-ring lines will not ring when the phone is in DND mode. To enable feature-ring lines to ring when the phone is in DND mode, use the **no dnd feature-ring** command.

### Examples

For the following example, when DND is active on ephone 1 and ephone 2, button 1 will ring, but button 2 will not.

```
Router(config)# ephone-dn 1
Router(config-ephone-dn)# number 1001

Router(config)# ephone-dn 2
Router(config-ephone-dn)# number 1002

Router(config)# ephone-dn 10

Router(config-ephone)# number 1110
Router(config-ephone)# preference 0
Router(config-ephone)# no huntstop

Router(config)# ephone-dn 11
Router(config-ephone)# number 1111
Router(config-ephone)# preference 1
Router(config-ephone)# no huntstop

Router(config)# ephone 1
Router(config-ephone)# button 1f1
Router(config-ephone)# button 2o10,11
Router(config-ephone)# no dnd feature-ring
```



```
Router(config-ephone-dn)# ephone 2
Router(config-ephone)# button 1f2
Router(config-ephone)# button 2o10,11
Router(config-ephone)# no dnd feature-ring
```

**Related Commands**

Command	Description
<b>button</b>	Associates ephone-dns with individual buttons on a Cisco IP phone and specifies ring behavior.

## dnd-control (voice register template)

To enable the Do-Not-Disturb (DND) soft key on SIP phones, use the **dnd-control** command in voice register template configuration mode. To disable the DND soft key on a SIP phone, use the **no** form of this command.

**dnd-control**  
**no dnd-control**

### Syntax Description

This command has no arguments or keywords.

### Command Default

DND soft key is enabled on SIP phones in Cisco Unified CME.

### Command Modes

Voice register template configuration (config-register-temp)

### Command History

Cisco IOS Release	Cisco Product	Modification
12.4(4)T	Cisco CME 3.4	This command was introduced.

### Usage Guidelines

This command enables a soft key for Do-Not-Disturb (DND) in the specified template which can then be applied to SIP phones. The DND soft key is enabled by default. To disable the DND soft key, use the **dnd** command. To apply a template to a SIP phone, use the template command in voice register pool configuration mode.

### Examples

The following example shows how to disable the DND soft key:

```
Router(config)# voice register template 1
Router(config-register-template)# dnd-control
```

### Related Commands

Command	Description
<b>dnd (voice register pool)</b>	Enables DND feature.

# dn-webedit

To enable the adding of extensions (ephone-dns) through the Cisco Unified CME graphical user interface (GUI), use the **dn-webedit** command in telephony-service configuration mode. To disable this feature, use the **no** form of this command.

**dn-webedit**  
**no dn-webedit**

## Syntax Description

This command has no arguments or keywords.

## Command Default

Extensions cannot be added through the Cisco Unified CME GUI.

## Command Modes

Telephony-service configuration (config-telephony)

## Command History

Cisco IOS Release	Cisco Product	Modification
12.2(2)XT	Cisco ITS 2.0	This command was introduced.
12.2(8)T	Cisco ITS 2.0	This command was integrated into Cisco IOS Release 12.2(8)T.
Cisco IOS XE Gibraltar 16.11.1a Release	Unified CME 12.6	The command is deprecated. It is not supported on Unified CME 12.6 and later releases.

## Usage Guidelines

The **dn-webedit** command enables the adding of extensions through the web-based GUI. If the **dn-webedit** command is enabled, a customer administrator or a system administrator can modify and assign extensions associated with the Cisco Unified CME router. If this ability is disabled, extensions must be added using Cisco IOS commands.

If the set of extension numbers used by the router is part of a larger telephone network, limitations on modification might be needed to ensure network integrity. Disabling the **dn-webedit** command prevents an administrator from allocating phone numbers and prevents assignment of numbers that may already be used elsewhere in the network.

## Examples

The following example enables editing of directory numbers through the web-based GUI interface:

```
Router(config)# telephony-service
Router(config-telephony)# dn-webedit
```

## Related Commands

Command	Description
<b>time-webedit</b>	Enables time setting through the web interface.

## dst (voice register global)

To set the time period for daylight saving time on SIP phones, use the **dst** command in voice register global configuration mode. To disable daylight saving time, use the **no** form of this command.

**dst auto-adjust**  
**no dst {start | stop}**

### Syntax Description

<b>start</b>	Sets beginning time for daylight saving time.
<b>stop</b>	Sets ending time for daylight saving time.
<i>month</i>	Abbreviated month. The following abbreviations are valid: <b>jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov, dec</b> .
<b>day</b> <i>day-of-month</i>	Date of the month. Range is 1 to 31.
<b>week</b> <i>week-number</i>	Number identifying the week of the month. Range is 1 to 4, or 8, where 8 represents the last week of the month.
<b>day</b> <i>day-of-week</i>	Abbreviated day of the week. The following abbreviations are valid: <b>sun, mon, tue, wed, thu, fri, sat</b> .
<b>time</b> <i>hour:minutes</i>	Beginning and ending time for daylight saving time, in HH:MM format using a 24-hour clock. The stop time must be greater than the start time. The value 24:00 is not valid. If you enter 00:00 for both start time and stop time, daylight saving time is enabled for the entire 24-hour period on the specified date.

### Command Default

Default start time is first week of April, Sunday, 2:00 a.m and default stop time is last week of October, Sunday 2:00 a.m.

### Command Modes

Voice register global configuration (config-register-global)

### Command History

Cisco IOS Release	Cisco Product	Modification
12.4(4)T	Cisco CME 3.4	This command was introduced.

### Usage Guidelines

This command sets the stop and start times for daylight saving time if the **dst auto-adjust** command is configured.

### Examples

The following example shows how to set automatic adjustment of daylight saving time:

```
Router(config)# voice register global
Router(config-register-global)# dst start Jan day 1 time 00:00
Router(config-register-global)# dst stop Mar day 31 time 23:99
```

**Related Commands**

Command	Description
<b>date-format (voice register global)</b>	Sets the date display format on SIP phones in a Cisco CME system.
<b>dst auto-adjust (voice register global)</b>	Enables automatic adjustment of daylight saving time on SIP phones.
<b>time-format (voice register global)</b>	Selects a 12-hour clock or a 24-hour clock for the time display format on Cisco IP phones in a SIP CME system.
<b>timezone (voice register global)</b>	Sets the time zone used for SIP phones in a Cisco CME system.

## dst auto-adjust (voice register global)

To enable automatic adjustment of daylight saving time on SIP phones, use the **dst auto-adjust** command in voice register global configuration mode. To disable daylight saving time auto adjustment, use the **no** form of this command.

**dst auto-adjust**  
**no dst auto-adjust**

<b>Syntax Description</b>	This command has no arguments or keywords.
<b>Command Default</b>	Automatic adjustment of daylight saving time on SIP phones is enabled.
<b>Command Modes</b>	Voice register global configuration (config-register-global)

<b>Command History</b>	Cisco IOS Release	Cisco Product	Modification
	12.4(4)T	Cisco CME 3.4	This command was introduced.

**Usage Guidelines** Automatic adjustment for daylight saving time is enabled by default. To disable auto adjusting for DST, use the **no dst auto-adjust** command. To set the start and stop times for DST, use the **dst** command.

**Examples** The following example shows how to disable the automatic adjustment for daylight saving time:

```
Router(config)# voice register global
Router(config-register-global)# no dst auto-adjust
```

<b>Related Commands</b>	Command	Description
	<b>date-format (voice register global)</b>	Sets the date display format on SIP phones in a Cisco CME system.
	<b>dst (voice register global)</b>	Sets the start and stop time if using daylight saving time on SIP phones.
	<b>time-format (voice register global)</b>	Selects a 12-hour clock or a 24-hour clock for the time display format on Cisco IP phones in a SIP CME system.
	<b>timezone (voice register global)</b>	Sets the time zone used for SIP phones in a Cisco CME system.

## dtmf-relay (voice register pool)

To specify the list of DTMF relay methods that can be used to relay dual-tone multifrequency (DTMF) audio tones between Session Initiation Protocol (SIP) endpoints, use the **dtmf-relay** command in voice register pool configuration mode. To send the DTMF audio tones as part of an audio stream, use the **no** form of this command.

**dtmf-relay** [**cisco-rtp**] [**rtp-nte**] [**sip-notify**] [**sip-kpml**]  
**no dtmf-relay**

### Syntax Description

<b>cisco-rtp</b>	Forwards DTMF audio tones by using Real-Time Transport Protocol (RTP) with a Cisco proprietary payload type. This keyword is supported only for dial peers that are created by incoming REGISTERS from a SIP gateway. It is not supported for dial peers that are created by a SIP Cisco IP phone.
<b>rtp-nte</b>	Forwards DTMF audio tones by using Real-Time Transport Protocol (RTP) with a Named Telephone Event (NTE) payload.
<b>sip-notify</b>	Forwards DTMF audio tones by using SIP-NOTIFY messages. This keyword is supported only for dial peers that are created by incoming REGISTERS from a SIP gateway. It is not supported for dial peers that are created by a SIP Cisco IP phone.
<b>sip-kpml</b>	Forwards DTMF audio tones through Keypad Markup Language (KPML) messages.

### Command Default

DTMF tones are disabled and sent in-band. That is, they remain in the audio stream.

### Command Modes

Voice register pool configuration (config-register-pool)

### Command History

Cisco IOS Release	Cisco Product	Modification
12.3(4)T	Cisco SIP SRST 3.0	This command was introduced.
12.4(4)T	Cisco CME 3.4 Cisco SIP SRST 3.4	This command was added to Cisco Unified CME.
15.1(1)T1	Cisco Unified CME 8.1 Cisco SIP SRST 8.1	The sip-kpml keyword was added to this command.

### Usage Guidelines

During Cisco Unified Session Initiation Protocol (SIP) Survivable Remote Site Telephony (SRST) or Cisco Unified CME registration, a dial peer is created and that dial peer has a default DTMF relay of in-band.

This command allows you to change the default to a desired value. You must use one or more keywords when configuring this command.

DTMF audio tones are generated when you press a button on a Touch-Tone phone. The tones are compressed at one end of the call and when the digits are decompressed at the other end, there is a risk that they can become distorted. DTMF relay reliably transports the DTMF audio tones generated after call establishment out-of-band.

The SIP Notify method sends Notify messages bidirectionally between the originating and terminating gateways for a DTMF event during a call. If multiple DTMF relay mechanisms are enabled on a SIP dial peer and are negotiated successfully, the SIP Notify method takes precedence.

SIP Notify messages are advertised in an Invite message to the remote end only if the **dtmf-relay** command is set.

For SIP calls, the most appropriate methods to transport DTMF tones are RTP-NTE or SIP-NOTIFY.

**Note**

The **cisco-rtp** keyword is a proprietary Cisco implementation. If the proprietary Cisco implementation is not supported, the DTMF relay feature does not function, and the gateway sends DTMF tones in-band.

- The **sip-notify** keyword is available only if the VoIP dial peer is configured for SIP.

**Examples****Cisco Unified CME**

The following example shows how to enable the RTP-NTE and SIP-NOTIFY mechanisms for DTMF relay for SIP phone 4:

```
Router(config)# voice register pool 4
Router(config-register-pool)# dtmf-relay rtp-nte sip-notify
```

The following example shows sip-kpml option configured for dtmf-relay in voice register pool 5:

```
Router#show running config
voice register global
  mode cme
  source-address 10.32.153.49 port 5060
  max-dn 200
  max-pool 100
!
voice register pool 5
  id mac 0023.3319.8B7B
  type 7945
  number 1 dn 5
  dtmf-relay sip-kpml
  username betaone password cisco
  codec g711ulaw
  no vad
```

**Cisco Unified SIP SRST**

The following is sample output from the **show running-config** command that shows that voice register pool 1 has been set up to send DTMF tones:

```
voice register pool 1
  application SIP.app
  incoming called-number 308
  voice-class codec 1
  dtmf-relay rtp-nte
```



**Related Commands**

Command	Description
<b>dtmf-relay (voice over IP)</b>	Specifies how an H.323 or SIP gateway relays DTMF tones between telephony interfaces and an IP network.

dtmf-relay (voice register pool)