



# Cisco Management Information Base

This chapter describes the Management Information Base (MIB) text files that are supported by Cisco Unified Communications Manager (Cisco Unified CM) and are used with Simple Network Management Protocol (SNMP).

- [CISCO-CCM-MIB](#), on page 1
- [CISCO-CCM-CAPABILITY](#), on page 131
- [CISCO-CDP-MIB](#), on page 136
- [CISCO-SYSLOG-MIB](#), on page 154
- [CISCO-SYSLOG-EXT-MIB](#), on page 163

## CISCO-CCM-MIB



**Note** This is a reformatted version of CISCO-CCM-MIB. Download and compile all of the MIBs in this section from <http://tools.cisco.com/Support/SNMP/do/BrowseMIB.do?local=en&step=2>.

This MIB manages the Cisco Unified Communications Manager (Cisco Unified CM) application running with a Cisco Communication Network (CCN) system. Cisco Unified CM is an IP-PBX that controls the call processing of a VoIP network.

A CCN system comprises multiple regions, with each region consisting of several Cisco Unified CM groups with multiple Cisco Unified CM servers. The MIB can be used by the Cisco Unified CM application, Cisco Unified CM Administration, to present provision and statistics information.

The following terminology applies to this MIB:

- SCCP—Skinny Client Control Protocol
- SIP—Session Initiation Protocol
- TLS—Transport Layer Security
- MGCP—Media Gateway Control Protocol

Before you can compile CISCO-CCM-MIB, you need to download and compile the MIBs listed below in the order listed.

1. SNMPv2-SMI

2. SNMPv2-TC
3. SNMPv2-CONF
4. CISCO-SMI
5. INET-ADDRESS-MIB
6. SNMP-FRAMEWORK-MIB
7. RFC1155-SMI
8. RFC1212
9. SNMPv2-TC-v1
10. CISCO-CCM-MIB

Additional downloads are:

- OID File: CISCO-CCM-MIB.OID
- Capability File: CISCO-CCM-CAPABILITY

## CISCO-CCM-MIB Revisions

The following table lists the revisions to this MIB beginning with the latest revision first.

*Table 1: History of CISCO-CCM-MIB Revisions*

Date	Action	Description
July 2010	Updated the TEXTUAL-CONVENTIONS	CcmDevUnregCauseCode, CcmDevRegFailCauseCode

Date	Action	Description
Dec 2009	Deprecated	CcmDevFailCauseCode; Added CcmDevRegFailCauseCode and CcmDevUnregCauseCode
	Deprecated	ccmPhoneStatusReason; Added ccmPhoneUnregReason and ccmPhoneRegFailReason in ccmPhoneTable
	Deprecated	ccmPhoneFailCauseCode; Added ccmPhoneFailedRegFailReason in ccmPhoneFailedTable
	Deprecated	ccmPhoneStatusUpdateReason; Added ccmPhoneStatusUnregReason and ccmPhoneStatusRegFailReason in ccmPhoneStatusUpdateTable
	Deprecated	ccmGatewayStatusReason; Added ccmGatewayUnregReason and ccmGatewayRegFailReason in ccmGatewayTable.
	Deprecated	ccmMediaDeviceStatusReason; Added ccmMediaDeviceUnregReason and ccmMediaDeviceRegFailReason in ccmMediaDeviceTable.
	Deprecated	ccmCTIDeviceStatusReason; Added ccmCTIDeviceUnregReason and ccmCTIDeviceRegFailReason in ccmCTIDeviceTable
	Deprecated	ccmH323DevStatusReason; Added ccmH323DevUnregReason and ccmH323DevRegFailReason in ccmH323Device Table.
	Deprecated	ccmVMailDevStatusReason; Added ccmVMailDevUnregReason and ccmVMailDevRegFailReason in ccmVoiceMailDeviceTable.
	Deprecated	

Date	Action	Description
		ccmGatewayFailCauseCode; Added ccmGatewayRegFailCauseCode in ccmNotificationsInfo.
	Deprecated the following Notification Type	ccmGatewayFailed and added ccmGatewayFailedReason.
	Deprecated following OBJECT_GROUPS	ccmPhoneInfoGroupRev5, ccmNotificationsInfoGroupRev4, ccmGatewayInfoGroupRev3, ccmMediaDeviceInfoGroupRev3, ccmCTIDeviceInfoGroupRev3, ccmH323DeviceInfoGroupRev2, ccmVoiceMailDeviceInfoGroupRev1 and ccmNotificationsGroupRev2; Added following OBJECT_GROUPS: ccmPhoneInfoGroupRev6, ccmNotificationsInfoGroupRev5, ccmGatewayInfoGroupRev4, ccmMediaDeviceInfoGroupRev4, ccmCTIDeviceInfoGroupRev4, ccmH323DeviceInfoGroupRev3, ccmVoiceMailDeviceInfoGroupRev2, ccmNotificationsGroupRev3.
	Deprecated following MODULE-COMPLIANCE	ciscoCcmMIBComplianceRev6; Added ciscoCcmMIBComplianceRev7.
	Obsoleted following OBJECT_GROUPS	ccmInfoGroupRev3, ccmH323DeviceInfoGroupRev1

Date	Action	Description
08-21-2008	Added following objects in ccmCTIDeviceTable	ccmCTIDeviceInetAddressIPv4 ccmCTIDeviceInetAddressIPv6  These objects replaced the ccmCTIDeviceInetAddressType and ccmCTIDeviceInetAddress.
	Deprecated following objects in ccmCTIDeviceTable	ccmCTIDeviceInetAddressType ccmCTIDeviceInetAddress
	Added following OBJECT-GROUP	ccmCTIDeviceInfoGroupRev3.  This group replaced the ccmCTIDeviceInfoGroupRev2
	Deprecated following OBJECT-GROUP	ccmCTIDeviceInfoGroupRev2
	Added following MODULE-COMPLIANCE	ciscoCcmMIBComplianceRev6  This compliance replaced the ciscoCcmMIBComplianceRev5.
	Deprecated	ciscoCcmMIBComplianceRev5 MODULE-COMPLIANCE

Date	Action	Description
02-12-2008	Added following objects in ccmTable	ccmInetAddress2 ccmInetAddress2Type
	Added following objects in ccmPhoneTable	ccmPhoneInetAddressIPv4 ccmPhoneInetAddressIPv6 ccmPhoneIPv4Attribute ccmPhoneIPv6Attribute ccmPhoneActiveLoadID
	Added following objects in ccmPhoneFailedTable	ccmPhoneFailedInetAddressIPv4 ccmPhoneFailedInetAddressIPv6 ccmPhoneFailedIPv4Attribute ccmPhoneFailedIPv6Attribute
	Added following objects in ccmSIPDeviceTable	ccmSIPDevInetAddressIPv4 ccmSIPDevInetAddressIPv6
	Added following objects in ccmMediaDeviceTable	ccmMediaDeviceInetAddressIPv4 ccmMediaDeviceInetAddressIPv6
	Deprecated following objects in ccmPhoneTable	ccmPhoneInetAddressType ccmPhoneInetAddress
	Deprecated following objects in ccmPhoneFailedTable	ccmPhoneFailedInetAddressType ccmPhoneFailedInetAddress
	Deprecated following objects in ccmSIPDeviceTable	ccmSIPDevInetAddressType ccmSIPDevInetAddress
	Deprecated following objects in ccmMediaDeviceTable	ccmMediaDeviceInetAddressType ccmMediaDeviceInetAddress
	Added following scalar objects	ccmH323TableEntries ccmSIPTableEntries
	Obsoleted	ciscoCcmMIBComplianceRev3 MODULE-COMPLIANCE
	Deprecated	ciscoCcmMIBComplianceRev4 MODULE-COMPLIANCE
	Added	ciscoCcmMIBComplianceRev5 MODULE-COMPLIANCE

Date	Action	Description
	Obsoleted following NOTIFICATION-GROUPS	ccmNotificationsGroup ccmNotificationsGroupRev1
	Obsoleted following OBJECT-GROUPS	ccmInfoGroupRev2 ccmPhoneInfoGroupRev3 ccmSIPDeviceInfoGroup ccmNotificationsInfoGroupRev1 ccmNotificationsInfoGroupRev2
	Deprecated following OBJECT-GROUPS	ccmInfoGroupRev3 ccmPhoneInfoGroupRev4 ccmSIPDeviceInfoGroupRev1 ccmMediaDeviceInfoGroupRev2 ccmH323DeviceInfoGroupRev1 ccmNotificationsInfoGroupRev3
	Added following OBJECT-GROUPS	ccmInfoGroupRev4 ccmPhoneInfoGroupRev5 ccmMediaDeviceInfoGroupRev3 ccmNotificationsInfoGroupRev4 ccmH323DeviceInfoGroupRev2 ccmSIPDeviceInfoGroupRev2

Date	Action	Description
09-14-2005	Updated CcmDevFailCauseCode definition to include more cause codes.	authenticationError invalidX509NameInCertificate invalidTLSCipher, directoryNumberMismatch malformedRegisterMsg
	Updated the description of these objects.	ccmPhoneFailedInetAddress ccmGatewayInetAddress ccmMediaDeviceInetAddress ccmGatekeeperInetAddress ccmCTIDeviceInetAddress ccmH323DevInetAddress ccmH323DevCnfgGKInetAddress ccmH323DevAltGK2InetAddress ccmH323DevAltGK3InetAddress ccmH323DevAltGK4InetAddress ccmH323DevAltGK5InetAddress ccmH323DevActGKInetAddress ccmH323DevRmtCM1InetAddress ccmH323DevRmtCM2InetAddress ccmH323DevRmtCM3InetAddress ccmVMailDevInetAddress

Date	Action	Description
09-05-2005	Added partiallyregistered to CcmDeviceStatus TC	—
	Added phonePartiallyregistered to ccmPhoneStatusUpdateType TC	—
	Added these TCs	CcmPhoneProtocolType CcmDeviceLineStatus CcmSIPTransportProtocolType
	Added these objects to ccmPhoneTable	ccmPhoneProtocol ccmPhoneName
	Added ccmPhoneExtnStatus to ccmPhoneExtnTable	—
	Added following objects to ccmSIPDeviceTable:	ccmSIPInTransportProtocolType ccmSIPOutTransportProtocolType ccmSIPInPortNumber, ccmSIPOutPortNumber
	Added ccmTLSConnectionFailure notification	—
	Updated the description of following objects under ccmSIPDeviceTable	ccmTLSConnectionFailReasonCode ccmSIPDevName ccmSIPDevDescription ccmSIPDevInetAddress
	Updated the description of ccmCallManagerAlarmEnable	—
	Added the following object groups	ccmPhoneInfoGroupRev4 ccmNotificationsInfoGroupRev3 ccmSIPDeviceInfoGroupRev1
	Added the following notification groups: ccmNotificationsGroupRev2	—
	Added MIB compliance ciscoCcmMIBComplianceRev4	—

Date	Action	Description
08-02-2004	Obsoleted	ccmDeviceProductId ccmTimeZoneOffset ccmPhoneType ccmPhoneLastError ccmPhoneTimeLastError ccmPhoneExtensionTable ccmPhoneExtensionTable ccmPhoneExtensionEntry ccmPhoneExtensionEntry ccmPhoneExtensionIndex ccmPhoneExtensionIndex ccmPhoneExtension ccmPhoneExtensionMultiLines ccmPhoneExtensionInetAddressType ccmPhoneExtensionInetAddress ccmPhoneFailedName ccmGatewayType ccmGatewayProductId ccmActivePhones ccmInActivePhones ccmActiveGateways ccmInActiveGateways ccmMediaDeviceType ccmCTIDeviceType ccmCTIDeviceAppInfo

Date	Action	Description
		ccmH323DevProductId, ccmVMailDevProductId ciscoCcmMIBComplianceRev2 ccmInfoGroupRev1 ccmPhoneInfoGroupRev1 ccmGatewayInfoGroupRev1 ccmCTIDeviceInfoGroup ccmNotificationsInfoGroup ccmPhoneInfoGroupRev2 ccmGatewayInfoGroupRev2 ccmMediaDeviceInfoGroupRev1 ccmCTIDeviceInfoGroupRev1 ccmH323DeviceInfoGroup ccmVoiceMailDeviceInfoGroup

Date	Action	Description
08-25-2003	Added	The definition of ccmMaliciousCall and ccmQualityReport notifications and its objects
	Added	H323 trunk types and SIP trunk type in ccmDeviceProductId
	Added	More media device types in ccmMediaDevice table
	Added	The definition of ccmSystemVersion and ccmInstallationId objects to ccmGlobalInfo group
	Added	ccmSIPDeviceInfo definition
	Added	More phone types
	Added	The definition of ccmProductTypeTable to list the product types supported at run time
	Added	ccmPhoneProductTypeIndex ccmGatewayProductTypeIndex ccmMediaDeviceProductTypeIndex ccmCTIDeviceProductTypeIndex ccmH323DevProductTypeIndex ccmVMailDevProductTypeIndex objects
05-08-2003	Deprecated	ccmPhoneType ccmGatewayType ccmGatewayProductId ccmMediaDeviceType ccmCTIDeviceTYpe ccmH323DevProductId ccmVMailDevProductId and objects CcmDeviceProductId
	Added	More phone types in the ccmPhoneType definition
	Added	More gateway types in the ccmGatewayType and CcmDeviceProductId definition

Date	Action	Description
01-11-2002	Updated	CcmDevFailCauseCode definition to include more cause codes deviceInitiatedReset, callManagerReset and noError
	Added	ccmH323DeviceInfo and ccmVoiceMailDeviceInfo objects
	Updated	ccmRegionAvailableBandwidth definition to include two more bandwidth types: bwGSM and bwWideband
	Deprecated	ccmTimeZoneOffset object
	Added	ccmTimeZoneOffsetHours and ccmTimeZoneOffsetMinutes to ccmTimeZoneTable
	Added	ccmCTIDeviceStatusReason ccmCTIDeviceStatusReason ccmCTIDeviceTimeLastStatusUpdt ccmCTIDeviceTimeLastRegistered to ccmCTIDevice Table
	Added	Rejected status to ccmCTIDeviceStatus
	Added	More objects to the ccmGlobalInfo
	Added	ccmPhoneStatusUpdate ccmPhoneStatusUpdateReason ccmPhoneStatusUpdate ccmPhoneStatusUpdateReason object to ccmPhoneStatusUpdate ccmPhoneStatusUpdate table
	Added	ccmGatewayProductId ccmGatewayStatusReason ccmGatewayStatusReason ccmGatewayTimeLastStatusUpdt ccmGatewayTimeLastRegistered ccmGatewayDChannelStatus ccmGatewayDChannelNumber objects to ccmGatewayTable
	Added	New types to ccmGatewayType

Date	Action	Description
	Added	Rejected status to ccmGatewayStatus
	Obsoleted	The ccmGatewayTrunkInfo (this was never supported)
	Added	ccmMediaDeviceStatusReason ccmMediaDeviceStatusReason, ccmMediaDeviceTimeLastStatusUpdt ccmMediaDeviceTimeLastRegistered to ccmMediaDeviceTable
	Added	More types to ccmMediaDeviceType
	Added	Rejected status to ccmMediaDeviceStatus
	Deprecated	The ccmGatekeeperTable definition
	Added	Rejected status to ccmGatekeeperstatus
	Updated	ccmMIBCompliance statements
	Added	ccmPhoneStatusReason ccmPhoneStatusReason ccmPhoneTimeLastStatusUpdt to ccmPhoneTable
	Added	Rejected status to ccmPhoneStatus
	Deprecated	ccmPhoneFailedName and added ccmPhoneMacAddress to ccmPhoneFailedTable
	Deprecated	ccmPhoneLastError and ccmPhoneTimeLastError in ccmPhoneTable
	Deprecated	ccmCTIDeviceAppInfo in ccmCTIDeviceTable
	Defined	CcmDeviceProductId and CcmDeviceStatus textual conventions
	Added	ccmPhoneExtnTable ccmPhStatUpdtTblLastAddedIndex ccmPhFailedTblLastAddedIndex
	Deprecated	ccmPhoneExtensionTable
	Changed the default values	

Date	Action	Description
		ccmCallManagerAlarmEnable ccmGatewayAlarmEnable ccmPhoneFailedStorePeriod ccmPhoneStatusUpdate ccmPhoneStatusUpdateStorePeriod objects ccmPhoneFailedStorePeriod ccmPhoneStatusUpdate ccmPhoneStatusUpdateStorePeriod objects
12-01-2000	Added	ccmMediaDeviceInfo ccmGatekeeperInfo ccmCTIDeviceInfo ccmAlarmConfigInfo ccmNotificationsInfo objects
	Added	ccmClusterId to the ccmEntry
	Deprecated	ccmGatewayTrunkInfo (this was never implemented and it should have been in the gateway MIB)
	Added	ccmPhoneFailedTable and ccmPhoneStatusUpdateTable
	Added	ccmMIBNotifications
	Added	New ccmGatewayType and ccmPhoneType
	Added	This revision clause.
03-10-2000	The initial version of this MIB module	::= { ciscoMgmt 156 }

## CISCO-CCM-MIB Definitions

The following definitions are imported for CISCO-CCM-MIB:

- MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE, IpAddress, Counter32, Integer32, Unsigned32
- From SNMPv2-SMI—DateAndTime, TruthValue, MacAddress, TEXTUAL-CONVENTION
- From SNMPv2-TC—SnmAdminString
- From SNMP-FRAMEWORK-MIB—MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP

- From SNMPv2-CONF—ciscoMgmt
- From CISCO-SMI—InetAddressType, InetAddress, InetPortNumber
- From INET-ADDRESS-MIB

## CISCO-CCM-MIB Textual Conventions

### **CcmIndex ::= TEXTUAL-CONVENTION**

DISPLAY-HINT d

STATUS current

DESCRIPTION

This syntax is used as the Index into a table. A positive value is used to identify a unique entry in the table.

SYNTAX Unsigned32(1..4294967295)

### **CcmIndexOrZero ::= TEXTUAL-CONVENTION**

DISPLAY-HINT d

STATUS current

DESCRIPTION

This textual convention is an extension of the CcmIndex convention. The latter defines a greater than zero to identify an entry of the CCM MIB table in the managed system. This extension permits the additional value of zero. The value zero is object-specific and must be defined as part of the description of any object that uses this syntax.

SYNTAX Unsigned32 (0..4294967295)

### **CcmDevRegFailCauseCode ::= TEXTUAL-CONVENTION**

STATUS current

DESCRIPTION

This syntax is used as means of identifying the reasons for a device registration failure. The scope of this enumeration can expand to comply with RFC 2578.

noError: No Error

unknown: Unknown error cause

noEntryInDatabase: Device not configured properly in the Cisco Unified CM database

databaseConfigurationError: Device configuration error in the Cisco Unified CM database

deviceNameUnresolveable: The Cisco Unified CM is unable to resolve the device name to an IP Address internally

maxDevRegExceeded: Maximum number of device registrations have been reached

connectivityError: Cisco Unified CM is unable to establish communication with the device during registration

initializationError: Indicates an error occurred when the Cisco Unified CM tries to initialize the device

`deviceInitiatedReset`: Indicates that the error was due to device initiated reset

`callManagerReset`: Indicates that the error was due to Cisco Unified CM reset

`authenticationError`: Indicates mismatch between configured authentication mode and the authentication mode that the device is using to connect to the Cisco Unified CM

`invalidX509NameInCertificate`: Indicates mismatch between the peer X.509 certificate subject name and what is configured for the device

`invalidTLSCipher`: Indicates Cipher mismatch during TLS handshake process

`directoryNumberMismatch`: Indicates mismatch between the directory number that the SIP device is trying to register with and the directory number configured in the Cisco Unified CM for the SIP device

`malformedRegisterMsg`: Indicates that SIP device attempted to register with Cisco Unified CM, but the REGISTER message contained formatting errors

`protocolMismatch`: The protocol of the device (SIP or SCCP) does not match the configured protocol in Cisco Unified CM

`deviceNotActive`: The device has not been activated

`authenticatedDeviceAlreadyExists`: A device with the same name is already registered with Cisco Unified CM

`obsoleteProtocolVersion`: The SCCP device registered with an obsolete protocol version

`databaseTimeout`: Cisco Unified CM requested device configuration data from the database but did not receive a response within 10 minutes

`registrationSequenceError`: (SCCP only) A device requested configuration information from the Cisco Unified CM at an unexpected time. The Cisco Unified CM had not yet obtained the requested information. The device will automatically attempt to register again. If this alarm occurs again, manually reset the device. If this alarm continues to occur after the manual reset, there may be an internal firmware error

`invalidCapabilities`: (SCCP only) The Cisco Unified CM detected an error in the media capabilities reported in the `StationCapabilitiesRes` message by the device during registration. The device will automatically attempt to register again. If this alarm occurs again, manually reset the device. If this alarm continues to occur after the manual reset, there may be a protocol error

`capabilityResponseTimeout`: (SCCP only) The Cisco Unified CM timed out while waiting for the device to respond to a request to report its media capabilities. Possible causes include device power outage, network power outage, network configuration error, network delay, packet drops, and packet corruption. It is also possible to get this error if the Cisco Unified CM node is experiencing high CPU usage. Verify that the device is powered up and operating. Verify that network connectivity exists between the device and Cisco Unified CM, and verify that the CPU utilization is in the safe range

`securityMismatch`: The Cisco Unified CM detected a mismatch in the security settings of the device and/or the Cisco Unified CM. The mismatches that can be detected are:

- The device established a secure connection, yet reported that it does not have the ability to do authenticated signaling.
- The device did not establish a secure connection, but the security mode configured for the device indicates that it should have done so.
- The device established a secure connection, but the security mode configured for the device indicates that it should not have done so.

autoRegisterDBError—Auto-registration of a device failed for one of the following reasons:

- Auto-registration is not allowed for the device type.
- An error occurred while adding the auto-registering device to the database (stored procedure).

dbAccessError: Device registration failed because of an error that occurred while building the station registration profile. This usually indicates a synchronization problem with the database

autoRegisterDBConfigTimeout: (SCCP only) The Cisco Unified CM timed out during auto-registration of a device. The registration profile of the device did not get inserted into the database in time. The device will automatically attempt to register again

deviceTypeMismatch: The device type reported by the device does not match the device type configured on the Cisco Unified CM

addressingModeMismatch: (SCCP only) The Cisco Unified CM detected an error related to the addressing mode configured for the device. One of the following errors were detected:

- The device is configured to use only IPv4 addressing, but did not specify an IPv4 address.
- The device is configured to use only IPv6 addressing, but did not specify an IPv6 address.

```
SYNTAX INTEGER {
noError(0),
unknown(1),
noEntryInDatabase(2),
databaseConfigurationError(3),
deviceNameUnresolveable(4),
maxDevRegExceeded(5),
connectivityError(6),
initializationError(7),
deviceInitiatedReset(8),
callManagerReset(9),
authenticationError(10),
invalidX509NameInCertificate(11),
invalidTLSCipher(12),
directoryNumberMismatch(13),
malformedRegisterMsg(14),
protocolMismatch(15),
deviceNotActive(16),
authenticatedDeviceAlreadyExists(17),
obsoleteProtocolVersion(18),
databaseTimeout(23),
registrationSequenceError(25),
```

```

invalidCapabilities(26),
capabilityResponseTimeout(27),
securityMismatch(28),
autoRegisterDBError(29),
dbAccessError(30),
autoRegisterDBConfigTimeout(31),
deviceTypeMismatch(32),
addressingModeMismatch(33)
}

```

### **CcmDevUnregCauseCode ::= TEXTUAL-CONVENTION**

STATUS current

DESCRIPTION

This syntax is used as means of identifying the reasons for a device getting unregistered. The scope of this enumeration can expand to comply with RFC 2578.

noError: No Error

unknown: Unknown error cause

noEntryInDatabase: Device not configured properly in the Cisco Unified CM database

databaseConfigurationError: Device configuration error in the Cisco Unified CM database

deviceNameUnresolveable: The Cisco Unified CM is unable to resolve the device name to an IP Address internally

maxDevRegExceeded: Maximum number of device registrations have been reached

connectivityError: Cisco Unified CM is unable to establish communication with the device during registration

initializationError: Indicates that an error occurred when the Cisco Unified CM tries to initialize the device

deviceInitiatedReset: Indicates that the error was due to device initiated reset

callManagerReset: Indicates that the error was due to Cisco Unified CM reset.

deviceUnregistered: DeviceUnregistered.

malformedRegisterMsg: Indicates that SIP device attempted to register with Cisco Unified CM, but the REGISTER message contained formatting errors.

sccpDeviceThrottling: The indicated SCCP device exceeded the maximum number of events allowed per-SCCP device.

keepAliveTimeout: A KeepAlive message was not received. Possible causes include device power outage, network power outage, network configuration error, network delay, packet drops, packet corruption and Cisco Unified CM node experiencing high CPU usage.

configurationMismatch: The configuration on the SIP device does not match the configuration in Cisco Unified CM.

**callManagerRestart:** A device restart was initiated from Cisco Unified CM Administration, either due to an explicit command from an administrator or due to a configuration change such as adding, deleting or changing a directory number associated with the device.

**duplicateRegistration:** Cisco Unified CM detected that the device attempted to register to two nodes at the same time. Cisco Unified CM initiated a restart to the phone to force it to re-home to a single node.

**callManagerApplyConfig:** Cisco Unified CM configuration is changed.

**deviceNoResponse:** Device is not responding Service Control Notify from Cisco Unified CM.

**emLoginLogout:** The device has been unregistered due to an Extension Mobility login or logout.

**emccLoginLogout:** The device has been unregistered due to an Extension Mobility Cross Cluster login or logout.

**powerSavePlus:** The device powered off as a result of the Power Save Plus feature that is enabled for this device. When the device powers off, it remains unregistered from Cisco Unified CM until the Phone On Time defined in the Product Specific Configuration for this device.

**callManagerForcedRestart:** (SIP Only) The device did not respond to an Apply Config request and as a result, Cisco Unified CM had sent a restart request to the device. The device may be offline due to a power outage or network problem. Confirm that the device is powered-up and that network connectivity exists between the device and Cisco Unified CM.

**sourceIPAddrChanged:** (SIP Only) The device has been unregistered because the IP address in the Contact header of the REGISTER message has changed. The device will be automatically reregistered. No action is necessary.

**sourcePortChanged:** (SIP Only) The device has been unregistered because the port number in the Contact header of the REGISTER message has changed. The device will be automatically reregistered. No action is necessary.

**registrationSequenceError:** (SCCP only) A device requested configuration information from the Cisco Unified CM at an unexpected time. The Cisco Unified CM no longer had the requested information in memory.

**invalidCapabilities:** (SCCP only) The Cisco Unified CM detected an error in the updated media capabilities reported by the device. The device reported the capabilities in one of the StationUpdateCapabilities message variants.

**fallbackInitiated:** The device has initiated a fallback and will automatically reregister to a higher-priority Cisco Unified CM. No action is necessary.

**deviceSwitch:** A second instance of an endpoint with the same device name has registered and assumed control. No action is necessary.

SYNTAX INTEGER {

noError(0),

unknown(1),

noEntryInDatabase(2),

databaseConfigurationError(3),

deviceNameUnresolveable(4),

maxDevRegExceeded(5),

connectivityError(6),

```

initializationError(7),
deviceInitiatedReset(8),
callManagerReset(9),
deviceUnregistered(10),
malformedRegisterMsg(11),
sccpDeviceThrottling(12),
keepAliveTimeout(13),
configurationMismatch(14),
callManagerRestart(15),
duplicateRegistration(16),
callManagerApplyConfig(17),
deviceNoResponse(18),
emLoginLogout(19),
emccLoginLogout(20),
energywisePowerSavePlus(21),
callManagerForcedRestart(22),
sourceIPAddrChanged(23),
sourcePortChanged(24),
registrationSequenceError(25),
invalidCapabilities(26),
fallbackInitiated(28),
deviceSwitch(29)
}

```

### **CcmDeviceStatus ::= TEXTUAL-CONVENTION**

STATUS current

DESCRIPTION

This syntax is used to identify the registration status of a device with the local Cisco Unified CM. The status is as follows:

- unknown—The registration status of the device is unknown
- registered—The device has successfully registered with the local Cisco Unified CM.
- unregistered—The device is no longer registered with the local Cisco Unified CM.
- rejected—Registration request from the device was rejected by the local Cisco Unified CM.
- partiallyregistered—At least one but not all of the lines are successfully registered to the Cisco Unified CM. Applicable only to SIP Phones.

SYNTAX INTEGER { unknown (1), registered (2), unregistered (3), rejected (4), partiallyregistered (5)}

**CcmPhoneProtocolType ::= TEXTUAL-CONVENTION**

STATUS current

DESCRIPTION

This syntax is used to identify the protocol between phone and Cisco Unified CM. The protocols are as follows:

- unknown—The phone protocol is unknown
- sccp—The phone protocol is SCCP
- sip—The phone protocol is SIP

SYNTAX INTEGER { unknown(1), sccp (2), sip(3) }

**CcmDeviceLineStatus ::= TEXTUAL-CONVENTION**

STATUS current

DESCRIPTION

This syntax is used to identify the registration status of a line of the device with the local Cisco Unified CM. The status is as follows:

- unknown—The registration status of the device line is unknown
- registered—The device line has successfully registered with the local Cisco Unified CM.
- unregistered—The device line is no longer registered with the local Cisco Unified CM.
- rejected—Registration request from the device line was rejected by the local Cisco Unified CM.

SYNTAX INTEGER { unknown (1), registered(2), unregistered (3), rejected (4)}

**CcmSIPTransportProtocolType ::= TEXTUAL-CONVENTION**

STATUS current

DESCRIPTION

This textual convention defines the possible transport protocol types that are used for setting up SIP calls unknown. The possible transport types are:

- unknown—The SIP Trunk transport type is unknown
- tcp—The SIP Trunk transport type is tcp
- udp—The SIP Trunk transport type is udp
- tcpAndUdp—The SIP Trunk transport type is tcp and udp
- tls—Applicable only for InTransportProtocolType is tls. The SIP Trunk transport type is tls.

SYNTAX INTEGER { unknown(1), tcp(2), udp(3), tcpAndUdp (4), tls(5) }

## CISCO-CCM-MIB Objects

ciscoCcmMIBObjects OBJECT IDENTIFIER ::= { ciscoCcmMIB 1 }  
 ccmGeneralInfo OBJECT IDENTIFIER ::= { ciscoCcmMIBObjects 1 }  
 ccmPhoneInfo OBJECT IDENTIFIER ::= { ciscoCcmMIBObjects 2 }  
 ccmGatewayInfo OBJECT IDENTIFIER ::= { ciscoCcmMIBObjects 3 }  
 ccmGatewayTrunkInfo OBJECT IDENTIFIER ::= { ciscoCcmMIBObjects 4 }  
 ccmGlobalInfo OBJECT IDENTIFIER ::= { ciscoCcmMIBObjects 5 }  
 ccmMediaDeviceInfo OBJECT IDENTIFIER ::= { ciscoCcmMIBObjects 6 }  
 ccmGatekeeperInfo OBJECT IDENTIFIER ::= { ciscoCcmMIBObjects 7 }  
 ccmCTIDeviceInfo OBJECT IDENTIFIER ::= { ciscoCcmMIBObjects 8 }  
 ccmAlarmConfigInfo OBJECT IDENTIFIER ::= { ciscoCcmMIBObjects 9 }  
 ccmNotificationsInfo OBJECT IDENTIFIER ::= { ciscoCcmMIBObjects 10 }  
 ccmH323DeviceInfo OBJECT IDENTIFIER ::= { ciscoCcmMIBObjects 11 }  
 ccmVoiceMailDeviceInfo OBJECT IDENTIFIER ::= { ciscoCcmMIBObjects 12 }  
 ccmQualityReportAlarmConfigInfo OBJECT IDENTIFIER ::= { ciscoCcmMIBObjects 13 }  
 ccmSIPDeviceInfo OBJECT IDENTIFIER ::= { ciscoCcmMIBObjects 14 }

## CISCO-CCM-MIB Tables

### Cisco Unified CM Group Table

#### **ccmGroupTable OBJECT-TYPE**

SYNTAX SEQUENCE OF CcmGroupEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

The table containing the CallManager groups in a Cisco Unified CM cluster.

::= { ccmGeneralInfo 1 }

#### **ccmGroupEntry OBJECT-TYPE**

SYNTAX CcmGroupEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

An entry (conceptual row) in the CallManager Group table, containing the information about a CallManager group in a Cisco Unified CM cluster. An entry is created to represent a CallManager Group. New entries to the CallManager Group table in the database are created when the User inserts a new

CallManager Group via the CallManager Web Admin pages. This entry is subsequently picked up by the Cisco Unified CM SNMP Agent.

```
INDEX { ccmGroupIndex }
 ::= { ccmGroupTable 1 }
```

### **CcmGroupEntry**

```
::= SEQUENCE
 {
  ccmGroupIndex CcmIndex,
  ccmGroupName SnmpAdminString,
  ccmGroupTftpDefault TruthValue
 }
```

### **ccmGroupIndex OBJECT-TYPE**

```
SYNTAX CcmIndex
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
  An arbitrary integer, selected by the local Cisco Unified CM that uniquely identifies a Cisco Unified CM Group.
 ::= { ccmGroupEntry 1 }
```

### **ccmGroupName OBJECT-TYPE**

```
SYNTAX SnmpAdminString (SIZE(0..128))
MAX-ACCESS read-only
STATUS current
DESCRIPTION
  The name of the Cisco Unified CM Group.
 ::= { ccmGroupEntry 2 }
```

### **ccmGroupTftpDefault OBJECT-TYPE**

```
SYNTAX TruthValue
MAX-ACCESS read-only
STATUS current
DESCRIPTION
  Whether this is the default TFTP server group or not.
 ::= { ccmGroupEntry 3 }
```

## Cisco Unified CM Table

### ccmTable OBJECT-TYPE

SYNTAX SEQUENCE of CcmEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

The table containing information of all the Cisco Unified CMs in a Cisco Unified CM cluster that the local Cisco Unified CM knows about. When the local Cisco Unified CM is restarted, this table will be refreshed.

::= { ccmGeneralInfo 2 }

### ccmEntry OBJECT-TYPE

SYNTAX CcmEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

An entry (conceptual row) in the CallManager table, containing the information about a CallManager.

INDEX { ccmIndex }

::= { ccmTable 1 }

CcmEntry ::= SEQUENCE

{

ccmIndex CcmIndex,

ccmName SnmpAdminString,

ccmDescription SnmpAdminString,

ccmVersion SnmpAdminString,

ccmStatus Integer,

ccmInetAddressType InetAddressType,

ccmInetAddress InetAddress,

ccmClusterId SnmpAdminString,

ccmInetAddress2Type InetAddressType,

ccmInetAddress2 InetAddress

}

### ccmIndex OBJECT-TYPE

SYNTAX CcmIndex

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

An arbitrary integer, selected by the local Cisco Unified CM, that uniquely identifies a CallManager in a Cisco Unified CM cluster.

::= { ccmEntry 1 }

**ccmName OBJECT-TYPE**

SYNTAX SnmpAdminString (SIZE(0..128))

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

The host name of the CallManager.

::= { ccmEntry 2 }

**ccmDescription OBJECT-TYPE**

SYNTAX SnmpAdminString (SIZE(0..255))

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

The description for the CallManager.

::= { ccmEntry 3 }

**ccmVersion OBJECT-TYPE**

SYNTAX SnmpAdminString (SIZE(0..24))

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

The version number of the CallManager software.

::= { ccmEntry 4 }

**ccmStatus OBJECT-TYPE**

SYNTAX INTEGER

```
{
unknown(1),
up(2),
down(3)
}
```

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

The current status of the CallManager. A CallManager is up if the SNMP Agent received a system up event from the local Cisco Unified CM:

unknown: Current status of the CallManager is Unknown

up: CallManager is running and is able to communicate with other CallManagers

down: CallManager is down or the Agent is unable to communicate with the local CallManager.

::= { ccmEntry 5 }

#### **ccmInetAddressType OBJECT-TYPE**

SYNTAX InetAddressType

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object identifies the IP address type of the Cisco Unified CM defined in ccmInetAddress.

::= { ccmEntry 6 }

#### **ccmInetAddress OBJECT-TYPE**

SYNTAX InetAddress

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object identifies IP address of the Cisco Unified CM. The type of address for this is identified by ccmInetAddressType.

::= { ccmEntry 7 }

#### **ccmClusterId OBJECT-TYPE**

SYNTAX SnmpAdminString (SIZE(0..128))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The unique ID of the Cluster to which this Cisco Unified CM belongs. At any point in time, the Cluster ID helps in associating a Cisco Unified CM to any given Cluster.

::= { ccmEntry 8 }

#### **ccmInetAddress2Type OBJECT-TYPE**

SYNTAX InetAddressType

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object identifies IP address type of the Cisco Unified Communications Manager defined in ccmInetAddress2.

```
::= { ccmEntry 9 }
```

#### **ccmInetAddress2 OBJECT-TYPE**

SYNTAX InetAddress

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object identifies the additional IP(v4/v6) address details of Cisco Unified Communications Manager. The type of address for this object is identified by ccmInetAddress2Type.

```
::= { ccmEntry 10 }
```

## **Cisco Unified CM Group Mapping Table**

#### **ccmGroupMappingTable OBJECT-TYPE**

SYNTAX SEQUENCE OF CcmGroupMappingEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

The table containing the list of all CallManager to group mappings in a Cisco Unified CM cluster. When the local Cisco Unified CM is down, this table will be empty.

```
::= { ccmGeneralInfo 3 }
```

#### **ccmGroupMappingEntry OBJECT-TYPE**

SYNTAX CcmGroupMappingEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

An entry (conceptual row) in the CallManager group Mapping table, containing the information about a mapping between a CallManager and a CallManager group.

INDEX { ccmGroupIndex, ccmIndex }

```
::= { ccmGroupMappingTable 1 }
```

```
CcmGroupMappingEntry ::= SEQUENCE {
    ccmCMGroupMappingCMPriority Unsigned32
}
```

#### **ccmCMGroupMappingCMPriority OBJECT-TYPE**

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The priority of the CallManager in the group. Sets the order of the CallManager in the list.

::= { ccmGroupMappingEntry 1 }

## Cisco Unified CM Region Table

### **ccmRegionTable OBJECT-TYPE**

SYNTAX SEQUENCE OF CcmRegionEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

The table containing the list of all geographically separated regions in a CCN system.

::= { ccmGeneralInfo 4 }

### **ccmRegionEntry OBJECT-TYPE**

SYNTAX CcmRegionEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

An entry (conceptual row) in the Region Table, containing the information about a region.

INDEX { ccmRegionIndex }

::= { ccmRegionTable 1 }

CcmRegionEntry ::= SEQUENCE {

ccmRegionIndex CcmIndex,

ccmRegionName SnmpAdminString

}

### **ccmRegionIndex OBJECT-TYPE**

SYNTAX CcmIndex

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

An arbitrary integer, selected by the local Cisco Unified CM, that uniquely identifies a Region Name in the table.

::= { ccmRegionEntry 1 }

### **ccmRegionName OBJECT-TYPE**

SYNTAX SnmpAdminString (SIZE(0..128))

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

The name of the CallManager region.

```
::= { ccmRegionEntry 2 }
```

## Cisco Unified CM Region Pair Table

### ccmRegionPairTable OBJECT-TYPE

SYNTAX SEQUENCE OF CcmRegionPairEntry

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

The table containing the list of all geographical region pairs defined for a Cisco Unified CM cluster. The pair consists of the Source region and Destination region.

```
::= { ccmGeneralInfo 5 }
```

### ccmRegionPairEntry OBJECT-TYPE

SYNTAX CcmRegionPairEntry

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

An entry (conceptual row) in the Region Pair Table, containing the information about bandwidth restrictions when communicating between the two specified regions.

INDEX { ccmRegionSrcIndex, ccmRegionDestIndex }

```
::= { ccmRegionPairTable 1 }
```

CcmRegionPairEntry ::= SEQUENCE {

ccmRegionSrcIndex CcmIndex,

ccmRegionDestIndex CcmIndex,

ccmRegionAvailableBandWidth INTEGER

}

### ccmRegionSrcIndex OBJECT-TYPE

SYNTAX CcmIndex

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

The index of the Source Region in the Region table.

```
::= { ccmRegionPairEntry 1 }
```

**ccmRegionDestIndex OBJECT-TYPE**

SYNTAX CcmIndex

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

The index of the Destination Region in the Region table.

::= { ccmRegionPairEntry 2 }

**ccmRegionAvailableBandWidth OBJECT-TYPE**

SYNTAX INTEGER {

unknown(1),

other(2),

bwG723(3),

bwG729(4),

bwG711(5),

bwGSM(6),

bwWideband(7)

}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The maximum available bandwidth between the two given regions.

unknown: Unknown Bandwidth

other: Unidentified Bandwidth

bwG723: For low bandwidth using G.723 codec

bwG729: For low bandwidth using G.729 codec

bwG711: For high bandwidth using G.711 codec

bwGSM: For GSM bandwidth 13K

bwWideband: For Wideband 256K.

::= { ccmRegionPairEntry 3 }

**Cisco Unified CM Time Zone Table****ccmTimeZoneTable OBJECT-TYPE**

SYNTAX SEQUENCE OF CcmTimeZoneEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

The table containing the list of all time zone groups in a call manager cluster.

::= { ccmGeneralInfo 6 }

**ccmTimeZoneEntry OBJECT-TYPE**

SYNTAX CcmTimeZoneEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

An entry (conceptual row) in the time zone Table, containing the information about a particular time zone group.

INDEX { ccmTimeZoneIndex }

::= { ccmTimeZoneTable 1 }

```
CcmTimeZoneEntry ::= SEQUENCE {
    ccmTimeZoneIndex CcmIndex,
    ccmTimeZoneName SnmpAdminString,
    ccmTimeZoneOffset Integer32,
    ccmTimeZoneOffsetHours Integer32,
    ccmTimeZoneOffsetMinutes Integer32
}
```

**ccmTimeZoneIndex OBJECT-TYPE**

SYNTAX CcmIndex

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

An arbitrary integer, selected by the local Cisco Unified CM, that uniquely identifies a Time Zone group entry in the table.

::= { ccmTimeZoneEntry 1 }

**ccmTimeZoneName OBJECT-TYPE**

SYNTAX SnmpAdminString (SIZE(0..128))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The name of the time zone group.

::= { ccmTimeZoneEntry 2 }

**ccmTimeZoneOffsetHours OBJECT-TYPE**

SYNTAX Integer32 (-12..12)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The offset hours of the time zone group's time zone from GMT.

::= { ccmTimeZoneEntry 4 }

#### **ccmTimeZoneOffsetMinutes OBJECT-TYPE**

SYNTAX Integer32 (-59..59)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The offset minutes of the time zone group's time zone from GMT.

::= { ccmTimeZoneEntry 5 }

## **Device Pool Table**

#### **ccmDevicePoolTable OBJECT-TYPE**

SYNTAX SEQUENCE OF CcmDevicePoolEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

The table containing the list of all device pools in a call manager cluster. A Device Pool contains Region, Date/Time Group and CallManager Group criteria that will be common among many devices.

::= { ccmGeneralInfo 7 }

#### **ccmDevicePoolEntry OBJECT-TYPE**

SYNTAX CcmDevicePoolEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

An entry (conceptual row) in the device pool Table, containing the information about a particular device pool.

INDEX { ccmDevicePoolIndex }

::= { ccmDevicePoolTable 1 }

CcmDevicePoolEntry

::= SEQUENCE {

ccmDevicePoolIndex CcmIndex, ccmDevicePoolName SnmpAdminString, ccmDevicePoolRegionIndex CcmIndexOrZero, ccmDevicePoolTimeZoneIndex CcmIndexOrZero, ccmDevicePoolGroupIndex CcmIndexOrZero

}

#### **ccmDevicePoolIndex OBJECT-TYPE**

SYNTAX CcmIndex

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

An arbitrary integer, selected by the local Cisco Unified CM, that uniquely identifies a Device Pool entry in the table. Each entry contains Region, Date/Time Group and CallManager Group criteria that will be common among many devices, for that entry.

::= { ccmDevicePoolEntry 1 }

#### **ccmDevicePoolName OBJECT-TYPE**

SYNTAX SnmpAdminString (SIZE(0..128))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The name of the device pool.

::= { ccmDevicePoolEntry 2 }

#### **ccmDevicePoolRegionIndex OBJECT-TYPE**

SYNTAX CcmIndexOrZero

MAX-ACCESS read-only

STATUS current

DESCRIPTION

A positive value of this index is used to identify the Region to which this Device Pool entry belongs. A value of zero indicates that the index to the Region table is Unknown.

::= { ccmDevicePoolEntry 3 }

#### **ccmDevicePoolTimeZoneIndex OBJECT-TYPE**

SYNTAX CcmIndexOrZero

MAX-ACCESS read-only

STATUS current

DESCRIPTION

A positive value of this index is used to identify the TimeZone to which this Device Pool entry belongs. A value of zero indicates that the index to the TimeZone table is Unknown.

::= { ccmDevicePoolEntry 4 }

#### **ccmDevicePoolGroupIndex OBJECT-TYPE**

SYNTAX CcmIndexOrZero

MAX-ACCESS read-only

STATUS current

DESCRIPTION

A positive value of this index is used to identify the CallManager Group to which this Device Pool entry belongs. A value of zero indicates that the index to the CallManager Group table is Unknown.

::= { ccmDevicePoolEntry 5 }

## Cisco Unified CM Product Type Table

### ccmProductTypeTable OBJECT-TYPE

SYNTAX CcmProductTypeEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

The table containing the list of product types supported in a call manager cluster. The product types will include the list of phone types, gateway types, media device types, H323 device types, CTI device types, Voice Messaging device types and SIP device types.

::= { ccmGeneralInfo 8 }

### ccmProductTypeEntry OBJECT-TYPE

SYNTAX CcmProductTypeEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

An entry (conceptual row) in the ccmProductTypeTable, containing the information about a product type supported in a call manager cluster. An entry is created to represent a product type.

INDEX { ccmProductTypeIndex }

::= { ccmProductTypeTable 1 }

CcmProductTypeEntry ::= SEQUENCE {

ccmProductTypeIndex CcmIndex,

ccmProductType Unsigned32,

ccmProductName SnmpAdminString,

ccmProductCategory INTEGER

}

### ccmProductTypeIndex OBJECT-TYPE

SYNTAX CcmIndex

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

An arbitrary integer, selected by the local Cisco Unified CM, that uniquely identifies an entry in the ccmProductTypeTable.

```
::= { ccmProductTypeEntry 1 }
```

**ccmProductType OBJECT-TYPE**

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The type of the product as defined in the Cisco Unified CM database.

```
::= { ccmProductTypeEntry 2 }
```

**ccmProductName OBJECT-TYPE**

SYNTAX SnmpAdminString (SIZE(0..100))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The name of the product as defined in the Cisco Unified CM database.

```
::= { ccmProductTypeEntry 3 }
```

**ccmProductCategory OBJECT-TYPE**

SYNTAX INTEGER {

```
    unknown(-1),
    notApplicable(0),
    phone(1),
    gateway(2),
    h323Device(3),
    ctiDevice(4),
    voiceMailDevice(5),
    mediaResourceDevice(6),
    huntListDevice(7),
    sipDevice(8)
}
```

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The category of the product.

unknown: Unknown product category

notApplicable: Not Applicable

phone: Phone

gateway: Gateway

h323Device: H323 Device

ctiDevice: CTI Device

voiceMailDevice: Voice Messaging Device  
 mediaResourceDevice: Media Resource Device  
 huntListDevice: Hunt List Device  
 sipDevice: SIP Device.

::= { ccmProductTypeEntry 4 }

## Phone Table

### ccmPhoneTable OBJECT-TYPE

SYNTAX SEQUENCE OF CcmPhoneEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

The table containing the list of all IP Phone devices that have tried to register with the local Cisco Unified CM at least once. When the local Cisco Unified CM is restarted, this table will be refreshed.

::= { ccmPhoneInfo 1 }

### ccmPhoneEntry OBJECT-TYPE

SYNTAX CcmPhoneEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

An entry (conceptual row) in the phone Table, containing information about a particular phone device.

INDEX { ccmPhoneIndex }

::= { ccmPhoneTable 1 }

CcmPhoneEntry ::= SEQUENCE {

ccmPhoneIndex CcmIndex,

ccmPhonePhysicalAddress MacAddress,

ccmPhoneType INTEGER,

ccmPhoneDescription SnmpAdminString,

ccmPhoneUserName SnmpAdminString,

ccmPhoneIpAddress IpAddress,

ccmPhoneStatus CcmDeviceStatus,

ccmPhoneTimeLastRegistered DateAndTime,

ccmPhoneE911Location SnmpAdminString,

ccmPhoneLoadID SnmpAdminString,

ccmPhoneLastError Integer32,

ccmPhoneTimeLastError DateAndTime,

```

ccmPhoneDevicePoolIndex CcmIndexOrZero,
ccmPhoneInetAddressType InetAddressType,
ccmPhoneInetAddress InetAddress,
ccmPhoneStatusReason CcmDevFailCauseCode,
ccmPhoneTimeLastStatusUpdt DateAndTime,
ccmPhoneProductTypeIndexCcmIndexOrZero,
ccmPhoneProtocolCcmPhoneProtocolType,
ccmPhoneName SnmpAdminString
ccmPhoneInetAddressIPv4 InetAddressIPv4,
ccmPhoneInetAddressIPv6 InetAddressIPv6,
ccmPhoneIPv4Attribute INTEGER,
ccmPhoneIPv6Attribute INTEGER,
ccmPhoneActiveLoadID SnmpAdminString,
ccmPhoneUnregReason CcmDevUnregCauseCode,
ccmPhoneRegFailReason CcmDevRegFailCauseCode
}

```

**ccmPhoneIndex OBJECT-TYPE**

SYNTAX CcmIndex

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

An arbitrary integer, selected by the local Cisco Unified CM, that uniquely identifies a Phone within the Cisco Unified CM.

::= { ccmPhoneEntry 1 }

ccmPhonePhysicalAddress OBJECT-TYPE

SYNTAX MacAddress

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The physical address(MAC address) of the IP phone.

::= { ccmPhoneEntry 2 }

**ccmPhoneDescription OBJECT-TYPE**

SYNTAX SnmpAdminString (SIZE(0..255))

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

The description of the phone.

::= { ccmPhoneEntry 4 }

**ccmPhoneUserName OBJECT-TYPE**

SYNTAX SnmpAdminString (SIZE(0..255))

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

The name of the user of the phone. When the phone is not in use, the name would refer to the last known user of the phone.

::= { ccmPhoneEntry 5 }

**ccmPhoneStatus OBJECT-TYPE**

SYNTAX CcmDeviceStatus

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

The status of the phone. The status of the Phone changes from Unknown to registered when it registers itself with the local Cisco Unified CM.

::= { ccmPhoneEntry 7 }

**ccmPhoneTimeLastRegistered OBJECT-TYPE**

SYNTAX DateAndTime

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

The time when the phone last registered with the Cisco Unified CM.

::= { ccmPhoneEntry 8 }

**ccmPhoneE911Location OBJECT-TYPE**

SYNTAX SnmpAdminString (SIZE(0..255))

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

The E911 location of the phone.

::= { ccmPhoneEntry 9 }

**ccmPhoneLoadID OBJECT-TYPE**

SYNTAX SnmpAdminString (SIZE(0..128))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object identifies the configured load ID for the phone device.

::= { ccmPhoneEntry 10 }

#### **ccmPhoneDevicePoolIndex OBJECT-TYPE**

SYNTAX CcmIndexOrZero

MAX-ACCESS read-only

STATUS current

DESCRIPTION

A positive value of this index is used to identify the Device Pool to which this Phone entry belongs. A value of 0 indicates that the index to the Device Pool table is Unknown.

::= { ccmPhoneEntry 13 }

#### **ccmPhoneInetAddressType OBJECT-TYPE**

SYNTAX InetAddressType

MAX-ACCESS read-only

STATUS deprecated

DESCRIPTION

This object identifies the IP address type of the phone.

::= { ccmPhoneEntry 14 }

#### **ccmPhoneInetAddress OBJECT-TYPE**

SYNTAX InetAddress

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object identifies the last known IP address of the phone. The type of address for this is identified by ccmPhoneInetAddressType.

::= { ccmPhoneEntry 15 }

#### **ccmPhoneTimeLastStatusUpdt OBJECT-TYPE**

SYNTAX DateAndTime

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The time the status of the phone changed.

::= { ccmPhoneEntry 17 }

**ccmPhoneProductTypeIndex OBJECT-TYPE**

SYNTAX CcmIndexOrZero

MAX-ACCESS read-only

STATUS current

DESCRIPTION

A positive value of this index is used to identify the related product type entry in the ccmProductTypeTable. A value of 0 indicates that the index to the ccmProductTypeTable is Unknown.

::= { ccmPhoneEntry 18 }

**ccmPhoneProtocol OBJECT-TYPE**

SYNTAX CcmPhoneProtocolType

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The protocol used between the phone and Cisco Unified CM.

::= { ccmPhoneEntry 19 }

**ccmPhoneName OBJECT-TYPE**

SYNTAX SnmpAdminString (SIZE(0..128))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The name of the phone. The name of the phone can be <prefix> + MAC Address, where <prefix> is SEP for Cisco SCCP and SIP Phones. In the case of other phones such as communicator (soft phone) it can be free-form name, a string that uniquely identifies the phone.

::= { ccmPhoneEntry 20 }

**ccmPhoneInetAddressIPv4 OBJECT-TYPE**

SYNTAX InetAddressIPv4

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object identifies the last known primary IPv4 address of the Phone Device. This object contains value zero if IPV4 address is not available.

::= { ccmPhoneEntry 21 }

**ccmPhoneInetAddressIPv6 OBJECT-TYPE**

SYNTAX InetAddressIPv6

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

This object identifies the last known primary IPv6 address of the Phone device. This object contains value zero if IPV6 address is not available.

::= { ccmPhoneEntry 22 }

**ccmPhoneIPv4Attribute OBJECT-TYPE**

SYNTAX INTEGER {

unknown(0),

adminOnly(1),

controlOnly(2),

adminAndControl(3)

}

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

This object identifies the scope of ccmPhoneInetAddressIPv4.

unknown(0): It is not known if ccmPhoneInetAddressIPv4 is used for Administration purpose or Controlling purpose.

adminOnly(1): ccmPhoneInetAddressIPv4 is used for the serviceability or administrative purpose.

controlOnly(2): ccmPhoneInetAddressIPv4 is used for signaling or registration purpose.

adminAndControl(3): ccmPhoneInetAddressIPv4 is used for controlling as well as administrative purpose.

::= { ccmPhoneEntry 23 }

**ccmPhoneIPv6Attribute OBJECT-TYPE**

SYNTAX INTEGER {

unknown(0),

adminOnly(1),

controlOnly(2),

adminAndControl(3)

}

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

This object identifies the scope of ccmPhoneInetAddressIPv6.

unknown(0): It is not known if ccmPhoneInetAddressIPv6 is used for Administration purpose or Controlling purpose.

adminOnly(1): ccmPhoneInetAddressIPv6 is used for the serviceability or administrative purpose.

controlOnly(2): ccmPhoneInetAddressIPv6 is used for signaling or registration purpose.

adminAndControl(3): ccmPhoneInetAddressIPv6 is used for controlling as well as administrative purpose.

::= { ccmPhoneEntry 24 }

#### **ccmPhoneActiveLoadID OBJECT-TYPE**

SYNTAX SnmpAdminString

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object identifies the ID of actual load that is successfully loaded and running on the phone device. If the phone is successfully upgraded to the new load then ccmPhoneLoadID and ccmPhoneActiveLoadID will have same value. If the upgrade fails then the ccmPhoneLoadID has the configured load ID and ccmPhoneActiveLoadID has the actual load ID that is running on the phone.

::= { ccmPhoneEntry 25 }

#### **ccmPhoneUnregReason OBJECT-TYPE**

SYNTA CcmDevUnregCauseCode

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The reason code associated with unregistered phone.

::= { ccmPhoneEntry 26 }

#### **ccmPhoneRegFailReason OBJECT-TYPE**

SYNTAX CcmDevRegFailCauseCode

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The reason code associated with registration failed phone.

::= { ccmPhoneEntry 27 }

## **Phone Failed Table**

#### **ccmPhoneFailedTable OBJECT-TYPE**

SYNTAX SEQUENCE OF CcmPhoneFailedEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

The table containing the list of all phones that attempted to register with the local call manager and failed. The entries that have not been updated and kept at least for the duration specified in the

ccmPhoneFailedStorePeriod will be deleted. Reasons for these failures could be due to configuration error, maximum number of phones has been reached, lost contact, etc.

::= { ccmPhoneInfo 3 }

#### **ccmPhoneFailedEntry OBJECT-TYPE**

SYNTAX CcmPhoneFailedEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

An entry (conceptual row) in the PhoneFailed Table, one for each phone failure in the Cisco Unified CM.

INDEX { ccmPhoneFailedIndex }

::= { ccmPhoneFailedTable 1 }

CcmPhoneFailedEntry ::= SEQUENCE {

ccmPhoneFailedIndex CcmIndex,

ccmPhoneFailedTime DateAndTime,

ccmPhoneFailedName SnmpAdminString,

ccmPhoneFailedInetAddressType InetAddressType,

ccmPhoneFailedInetAddress InetAddress,

ccmPhoneFailCauseCode CcmDevFailCauseCode,

ccmPhoneFailedMacAddress MacAddress

ccmPhoneFailedInetAddressIPv4 InetAddressIPv4,

ccmPhoneFailedInetAddressIPv6 InetAddressIPv6,

ccmPhoneFailedIPv4Attribute INTEGER,

ccmPhoneFailedIPv6Attribute INTEGER,

ccmPhoneFailedRegFailReason CcmDevRegFailCauseCode

}

#### **ccmPhoneFailedIndex OBJECT-TYPE**

SYNTAX CcmIndex

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

An arbitrary integer, selected by the local Cisco Unified CM, that is incremented with each new entry in the ccmPhoneFailedTable. This integer value will wrap if needed.

::= { ccmPhoneFailedEntry 1 }

**ccmPhoneFailedTime OBJECT-TYPE**

SYNTAX DateAndTime

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The time when the phone failed to register with the Cisco Unified CM.

::= { ccmPhoneFailedEntry 2 }

**ccmPhoneFailedMacAddress OBJECT-TYPE**

SYNTAX MacAddress

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The MAC address of the failed phone.

::= { ccmPhoneFailedEntry 7 }

**ccmPhoneFailedInetAddressIPv4 OBJECT-TYPE**

SYNTAX InetAddressIPv4

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object identifies the last known primary IPv4 address of the phone experiencing a communication failure. This object contains value zero if IPV4 address is not available.

::= { ccmPhoneFailedEntry 8 }

**ccmPhoneFailedInetAddressIPv6 OBJECT-TYPE**

SYNTAX InetAddressIPv6

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object identifies the last known primary IPv6 address of the phone experiencing a communication failure. This object contains value zero if IPV6 address is not available.

::= { ccmPhoneFailedEntry 9 }

**ccmPhoneFailedIPv4Attribute OBJECT-TYPE**

SYNTAX INTEGER

{

unknown(0),

adminOnly(1),

```
controlOnly(2),
adminAndControl(3)
}
```

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object identifies the scope of `ccmPhoneFailedInetAddressIPv4`.

`unknown(0)`: It is not known if `ccmPhoneFailedInetAddressIPv4` is used for Administration purpose or Controlling purpose.

`adminOnly(1)`: `ccmPhoneFailedInetAddressIPv4` is used for the serviceability or administrative purpose.

`controlOnly(2)`: `ccmPhoneFailedInetAddressIPv4` is used for signaling or registration purpose.

`adminAndControl(3)`: `ccmPhoneFailedInetAddressIPv4` is used for controlling as well as administrative purpose.

```
::= { ccmPhoneFailedEntry 10 }
```

#### **ccmPhoneFailedIPv6Attribute OBJECT-TYPE**

SYNTAX INTEGER

```
{
unknown(0),
adminOnly(1),
controlOnly(2),
adminAndControl(3)
}
```

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object identifies the scope of `ccmPhoneFailedInetAddressIPv6`.

`unknown(0)`: It is not known if `ccmPhoneFailedInetAddressIPv6` is used for Administration purpose or Controlling purpose.

`adminOnly(1)`: `ccmPhoneFailedInetAddressIPv6` is used for the serviceability or administrative purpose.

`controlOnly(2)`: `ccmPhoneFailedInetAddressIPv6` is used for signaling or registration purpose.

`adminAndControl(3)`: `ccmPhoneFailedInetAddressIPv6` is used for controlling as well as administrative purpose.

```
::= { ccmPhoneFailedEntry 11 }
```

#### **ccmPhoneFailedRegFailReason OBJECT-TYPE**

SYNTAX CcmDevRegFailCauseCode

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The reason code associated with registration failed phone.

::= { ccmPhoneFailedEntry 12 }

## Phone Status Update Table

### **ccmPhoneStatusUpdateTable OBJECT-TYPE**

SYNTAX SEQUENCE OF CcmPhoneStatusUpdateEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

The table containing the list of all phone status updates with respect to the local call manager. This table will only have registered, unregistered, and partially-registered status updates. The rejected phones are stored in the ccmPhoneFailedTable. Each entry of this table is stored at least for the duration specified in the ccmPhoneStatusUpdateStorePeriod object, after that it will be deleted.

::= { ccmPhoneInfo 4 }

### **ccmPhoneStatusUpdateEntry OBJECT-TYPE**

SYNTAX CcmPhoneStatusUpdateEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

An entry (conceptual row) in the PhoneStatusUpdate Table, one for each phone status update in the Cisco Unified CM.

INDEX { ccmPhoneStatusUpdateIndex }

::= { ccmPhoneStatusUpdateTable 1 }

CcmPhoneStatusUpdateEntry ::= SEQUENCE {

ccmPhoneStatusUpdateIndex CcmIndex,

ccmPhoneStatusPhoneIndex CcmIndexOrZero,

ccmPhoneStatusUpdateTime DateAndTime,

ccmPhoneStatusUpdateType INTEGER,

ccmPhoneStatusUpdateReason CcmDevFailCauseCode

ccmPhoneStatusUnregReason CcmDevUnregCauseCode,

ccmPhoneStatusRegFailReason CcmDevRegFailCauseCode

}

### **ccmPhoneStatusUpdateIndex OBJECT-TYPE**

SYNTAX CcmIndex

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

An arbitrary integer, selected by the local Cisco Unified CM, that is incremented with each new entry in the ccmPhoneStatusUpdateTable. This integer value will wrap if needed.

::= { ccmPhoneStatusUpdateEntry 1 }

#### **ccmPhoneStatusPhoneIndex OBJECT-TYPE**

SYNTAX CcmIndexOrZero

MAX-ACCESS read-only

STATUS current

DESCRIPTION

A positive value of this index is used to identify an entry in the ccmPhoneTable. A value of zero indicates that the index to the ccmPhoneTable is Unknown.

::= { ccmPhoneStatusUpdateEntry 2 }

#### **ccmPhoneStatusUpdateTime OBJECT-TYPE**

SYNTAX DateAndTime

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The time of the phone's registration status change.

::= { ccmPhoneStatusUpdateEntry 3 }

#### **ccmPhoneStatusUpdateType OBJECT-TYPE**

SYNTAX INTEGER {

unknown(1),

phoneRegistered(2),

phoneUnregistered(3),

phonePartiallyregistered(4)

}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

States the type of phone status change.

unknown: Unknown status

phoneRegistered: Phone has registered with the Cisco Unified CM

phoneUnregistered: Phone is no longer registered with the Cisco Unified CM

phonePartiallyregistered: Phone is partially registered with the Cisco Unified CM

::= { ccmPhoneStatusUpdateEntry 4 }

#### **ccmPhoneStatusUnregReason OBJECT-TYPE**

SYNTAX CcmDevUnregCauseCode

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The reason code associated with unregistered phone.

::= { ccmPhoneStatusUpdateEntry 6 }

#### **ccmPhoneStatusRegFailReason OBJECT-TYPE**

SYNTAX CcmDevRegFailCauseCode

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The reason code associated with registration failed phone.

::= { ccmPhoneStatusUpdateEntry 7 }

## **Enhanced Phone Extension Table with Combination Index**

#### **ccmPhoneExtnTable OBJECT-TYPE**

SYNTAX SEQUENCE OF CcmPhoneExtnEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

The table containing the list of all phone extensions associated with the registered and unregistered phones in the ccmPhoneTable. This table has combination index ccmPhoneIndex, ccmPhoneExtnIndex so the ccmPhoneTable and the ccmPhoneExtnTable entries can be related.

::= { ccmPhoneInfo 5 }

#### **ccmPhoneExtnEntry OBJECT-TYPE**

SYNTAX CcmPhoneExtnEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

An entry (conceptual row) in the phone extension Table, containing the information about a particular phone extension.

INDEX { ccmPhoneIndex, ccmPhoneExtnIndex }

::= { ccmPhoneExtnTable 1 }

```

CcmPhoneExtnEntry ::= SEQUENCE {
    ccmPhoneExtnIndex CcmIndex,
    ccmPhoneExtn SnmpAdminString,
    ccmPhoneExtnMultiLines Unsigned32,
    ccmPhoneExtnInetAddressType InetAddressType,
    ccmPhoneExtnInetAddress InetAddress,
    ccmPhoneExtnStatus CcmDeviceLineStatus
}

```

**ccmPhoneExtnIndex OBJECT-TYPE**

SYNTAX CcmIndex

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

An arbitrary integer, selected by the local Cisco Unified CM, that uniquely identifies a Phone Extension within the Cisco Unified CM.

::= { ccmPhoneExtnEntry 1 }

**ccmPhoneExtn OBJECT-TYPE**

SYNTAX SnmpAdminString (SIZE(0..24))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The extension number of the extension.

::= { ccmPhoneExtnEntry 2 }

**ccmPhoneExtnMultiLines OBJECT-TYPE**

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The number of multiline appearances for each phone extension.

::= { ccmPhoneExtnEntry 3 }

**ccmPhoneExtnInetAddressType OBJECT-TYPE**

SYNTAX InetAddressType

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object identifies the IP address type of the phone extension.

::= { ccmPhoneExtnEntry 4 }

#### **ccmPhoneExtnInetAddress OBJECT-TYPE**

SYNTAX InetAddress

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object identifies the IP address of the phone extension. The type of address for this is identified by ccmPhoneExtnInetAddressType.

::= { ccmPhoneExtnEntry 5 }

#### **ccmPhoneExtnStatus OBJECT-TYPE**

SYNTAX CcmDeviceLineStatus

MAX-ACCESS read-only

STATUS current

DESCRIPTION

Represents the status of this phone line.

::= { ccmPhoneExtnEntry 6 }

## Gateway Table

#### **ccmGatewayTable OBJECT-TYPE**

SYNTAX SEQUENCE OF CcmGatewayEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

The table containing the list of all gateway devices that have tried to register with the local Cisco Unified CM at least once. When the local Cisco Unified CM is restarted, this table will be refreshed.

::= { ccmGatewayInfo 1 }

#### **ccmGatewayEntry OBJECT-TYPE**

SYNTAX CcmGatewayEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

An entry (conceptual row) in the gateway Table, one for each gateway device in the Cisco Unified CM.

INDEX { ccmGatewayIndex }

::= { ccmGatewayTable 1 }

```

CcmGatewayEntry ::= SEQUENCE {
    ccmGatewayIndex CcmIndex,
    ccmGatewayName SnmpAdminString,
    ccmGatewayType Integer,
    ccmGatewayDescription SnmpAdminString,
    ccmGatewayStatus CcmDeviceStatus,
    ccmGatewayDevicePoolIndex CcmIndexOrZero,
    ccmGatewayInetAddressType InetAddressType,
    ccmGatewayInetAddress InetAddress,
    ccmGatewayProductId CcmDeviceProductId,
    ccmGatewayStatusReason CcmDevFailCauseCode,
    ccmGatewayTimeLastStatusUpdt DateAndTime,
    ccmGatewayTimeLastRegistered DateAndTime,
    ccmGatewayDChannelStatus INTEGER,
    ccmGatewayDChannelNumber Integer32,
    ccmGatewayProductTypeIndex CcmIndexOrZero
    ccmGatewayUnregReason CcmDevUnregCauseCode,
    ccmGatewayRegFailReason CcmDevRegFailCauseCode
}

```

**ccmGatewayIndex OBJECT-TYPE**

SYNTAX CcmIndex

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

An arbitrary integer, selected by the local Cisco Unified CM, that uniquely identifies a Gateway within the scope of the local call manager.

::= { ccmGatewayEntry 1 }

**ccmGatewayName OBJECT-TYPE**

SYNTAX SnmpAdminString (SIZE(0..128))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This is the Gateway name assigned to the Gateway in the Cisco Unified CM. This name is assigned when a new device of type Gateway is added to the Cisco Unified CM.

::= { ccmGatewayEntry 2 }

**ccmGatewayType OBJECT-TYPE**

```
SYNTAX INTEGER {  
    unknown(1),  
    other(2),  
    ciscoAnalogAccess(3),  
    ciscoDigitalAccessPRI(4),  
    ciscoDigitalAccessT1(5),  
    ciscoDigitalAccessPRIPlus(6),  
    ciscoDigitalAccessWSX6608E1(7),  
    ciscoDigitalAccessWSX6608T1(8),  
    ciscoAnalogAccessWSX6624(9),  
    ciscoMGCPStation(10),  
    ciscoDigitalAccessE1Plus(11),  
    ciscoDigitalAccessT1Plus(12),  
    ciscoDigitalAccessWSX6608PRI(13),  
    ciscoAnalogAccessWSX6612(14),  
    ciscoMGCPTrunk(15),  
    ciscoVG200(16),  
    cisco26XX(17),  
    cisco362X(18),  
    cisco364X(19),  
    cisco366X(20),  
    ciscoCat4224VoiceGatewaySwitch(21),  
    ciscoCat4000AccessGatewayModule(22),  
    ciscoIAD2400(23),  
    ciscoVGCEndPoint(24),  
    ciscoVG224VG248Gateway(25),  
    ciscoVGCBBox(26),  
    ciscoATA186(27),  
    ciscoICS77XXMRP2XX(28),  
    ciscoICS77XXASI81(29),  
    ciscoICS77XXASI160(30),  
    ciscoSlotVGCPort(31),  
    ciscoCat6000AVVIDServModule(32),
```

ciscoWSX6600(33),  
 ciscoWSSVCCMMMS(34),  
 cisco3745(35),  
 cisco3725(36),  
 ciscoICS77XXMRP3XX(37),  
 ciscoICS77XXMRP38FXS(38),  
 ciscoICS77XXMRP316FXS(39),  
 ciscoICS77XXMRP38FXOM1(40),  
 cisco269X(41),  
 cisco1760(42),  
 cisco1751(43),

#### **ccmGatewayDescription OBJECT-TYPE**

SYNTAX SnmpAdminString (SIZE(0..255))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The description attached to the gateway device.

::= { ccmGatewayEntry 4 }

#### **ccmGatewayStatus OBJECT-TYPE**

SYNTAX CcmDeviceStatus

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The status of the gateway. The Gateway status changes from Unknown to Registered when the Gateway registers itself with the local Cisco Unified CM.

::= { ccmGatewayEntry 5 }

#### **ccmGatewayDevicePoolIndex OBJECT-TYPE**

SYNTAX CcmIndexOrZero

MAX-ACCESS read-only

STATUS current

DESCRIPTION

A positive value of this index is used to identify the Device Pool to which this Gateway entry belongs. A value of zero indicates that the index to the Device Pool table is Unknown.

::= { ccmGatewayEntry 6 }

**ccmGatewayInetAddressType OBJECT-TYPE**

SYNTAX InetAddressType

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object identifies the IP address type of the Gateway device. The value of this object is 'unknown(0)' if the IP address of a Gateway device is not available.

::= { ccmGatewayEntry 7 }

**ccmGatewayInetAddress OBJECT-TYPE**

SYNTAX InetAddress

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object identifies last known IP Address of the gateway. If the IP address is not available then this object contains an empty string. The type of address for this is identified by ccmGatewayInetAddressType.

::= { ccmGatewayEntry 8 }

**ccmGatewayTimeLastStatusUpdt OBJECT-TYPE**

SYNTAX DateAndTime

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The time the status of the gateway changed.

::= { ccmGatewayEntry 11 }

**ccmGatewayTimeLastRegistered OBJECT-TYPE**

SYNTAX DateAndTime

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The time the gateway last registered with the call manager.

::= { ccmGatewayEntry 12 }

**ccmGatewayDChannelStatus OBJECT-TYPE**

SYNTAX INTEGER {

active(1),

inActive(2),

unknown(3),

notApplicable(4)

}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The D-Channel status of the gateway.

active(1): The D-Channel is up

inActive(1): The D-Channel is down

unknown(3):The D-Channel status is unknown

notApplicable(4): The D-channel status is not applicable for this gateway.

::= { ccmGatewayEntry 13 }

#### **ccmGatewayDChannelNumber OBJECT-TYPE**

SYNTAX Integer32 (-1..24)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The D-Channel number of the gateway. A value of -1 in this field indicates that the DChannel number is not applicable for this gateway.

::= { ccmGatewayEntry 14 }

#### **ccmGatewayProductTypeIndex OBJECT-TYPE**

SYNTAX CcmIndexOrZero

MAX-ACCESS read-only

STATUS current

DESCRIPTION

A positive value of this index is used to identify the related product type entry in the ccmProductTypeTable. A value of 0 indicates that the index to the ccmProductTypeTable is Unknown.

::= { ccmGatewayEntry 15 }

#### **ccmGatewayUnregReason OBJECT-TYPE**

SYNTAX CcmDevUnregCauseCode

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The reason code associated with unregistered gateway.

::= { ccmGatewayEntry 16 }

#### **ccmGatewayRegFailReason OBJECT-TYPE**

SYNTAX CcmDevRegFailCauseCode

MAX-ACCESS read-only  
 STATUS current  
 DESCRIPTION  
 The reason code associated with registration failed gateway.  
 ::= { ccmGatewayEntry 17 }

## Gateway Trunk Table

### **CcmGatewayTrunkEntry**

```
 ::= SEQUENCE {
   ccmGatewayTrunkIndex CcmIndex,
   ccmGatewayTrunkType INTEGER,
   ccmGatewayTrunkName SnmpAdminString,
   ccmTrunkGatewayIndex CcmIndexOrZero,
   ccmGatewayTrunkStatus INTEGER
 }
```

## All Scalar Objects

### **ccmRegisteredPhones OBJECT-TYPE**

SYNTAX Counter32  
 MAX-ACCESS read-only  
 STATUS current  
 DESCRIPTION  
 The number of phones that are registered and actively in communication with the local call manager.  
 ::= { ccmGlobalInfo 5 }

### **ccmUnregisteredPhones OBJECT-TYPE**

SYNTAX Counter32  
 MAX-ACCESS read-only  
 STATUS current  
 DESCRIPTION  
 The number of phone that are unregistered or have lost contact with the local call manager.  
 ::= { ccmGlobalInfo 6 }

### **ccmRejectedPhones OBJECT-TYPE**

SYNTAX Counter32  
 MAX-ACCESS read-only  
 STATUS current

## DESCRIPTION

The number of phones whose registration requests were rejected by the local call manager.

::= { ccmGlobalInfo 7 }

**ccmRegisteredGateways OBJECT-TYPE**

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

The number of gateways that are registered and actively in communication with the local call manager.

::= { ccmGlobalInfo 8 }

**ccmUnregisteredGateways OBJECT-TYPE**

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

The number of gateways that are unregistered or have lost contact with the local call manager.

::= { ccmGlobalInfo 9 }

**ccmRejectedGateways OBJECT-TYPE**

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

The number of gateways whose registration requests were rejected by the local call manager.

::= { ccmGlobalInfo 10 }

**ccmRegisteredMediaDevices OBJECT-TYPE**

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

The number of media devices that are registered and actively in communication with the local call manager.

::= { ccmGlobalInfo 11 }

**ccmUnregisteredMediaDevices OBJECT-TYPE**

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The number of media devices that are unregistered or have lost contact with the local call manager.

::= { ccmGlobalInfo 12 }

**ccmRejectedMediaDevices OBJECT-TYPE**

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The number of media devices whose registration requests were rejected by the local call manager.

::= { ccmGlobalInfo 13 }

**ccmRegisteredCTIDevices OBJECT-TYPE**

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The number of CTI devices that are registered and actively in communication with the local call manager.

::= { ccmGlobalInfo 14 }

**ccmUnregisteredCTIDevices OBJECT-TYPE**

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The number of CTI devices that are unregistered or have lost contact with the local call manager.

::= { ccmGlobalInfo 15 }

**ccmRejectedCTIDevices OBJECT-TYPE**

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The number of CTI devices whose registration requests were rejected by the local call manager.

::= { ccmGlobalInfo 16 }

**ccmRegisteredVoiceMailDevices OBJECT-TYPE**

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The number of voice messaging devices that are registered and actively in communication with the local call manager.

::= { ccmGlobalInfo 17 }

#### **ccmUnregisteredVoiceMailDevices OBJECT-TYPE**

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The number of voice messaging devices that are unregistered or have lost contact with the local call manager.

::= { ccmGlobalInfo 18 }

#### **ccmRejectedVoiceMailDevices OBJECT-TYPE**

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The number of voice messaging devices whose registration requests were rejected by the local call manager.

::= { ccmGlobalInfo 19 }

#### **ccmCallManagerStartTime OBJECT-TYPE**

SYNTAX DateAndTime

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The last time the local call manager service started. This is available only when the local call manager is up and running.

::= { ccmGlobalInfo 20 }

#### **ccmPhoneTableStateId OBJECT-TYPE**

SYNTAX Integer32 (0..2147483647)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The current state of ccmPhoneTable. The initial value of this object is 0 and it will be incremented every time when there is a change (addition/deletion/modification) to the ccmPhoneTable. This value and ccmCallManagerStartTime should be used together to find if the table has changed or not. When the call manager is restarted, this will be reset to 0.

::= { ccmGlobalInfo 21 }

#### **ccmPhoneExtensionTableStateId OBJECT-TYPE**

SYNTAX Integer32 (0..2147483647)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The current state of ccmPhoneExtensionTable. The initial value of this object is 0 and it will be incremented every time when there is a change (addition/deletion/modification) to the ccmPhoneExtensionTable. This value and ccmCallManagerStartTime should be used together to find if the table has changed or not. When the call manager is restarted, this will be reset to 0.

::= { ccmGlobalInfo 22 }

#### **ccmPhoneStatusUpdateTableStateId OBJECT-TYPE**

SYNTAX Integer32 (0..2147483647)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The current state of ccmPhoneStatusUpdateTable. The initial value of this object is 0 and it will be incremented every time when there is a change (addition/deletion/modification) to the ccmPhoneStatusUpdateTable. This value and sysUpTime should be used together to find if the table has changed or not. When the SNMP service is restarted this value will be reset to 0.

::= { ccmGlobalInfo 23 }

#### **ccmGatewayTableStateId OBJECT-TYPE**

SYNTAX Integer32 (0..2147483647)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The current state of ccmGatewayTable. The initial value of this object is 0 and it will be incremented every time when there is a change (addition/deletion/modification) to the ccmGatewayTable. This value and ccmCallManagerStartTime should be used together to find if the table has changed or not. When the call manager is restarted, this will be reset to 0.

::= { ccmGlobalInfo 24 }

#### **ccmCTIDeviceTableStateId OBJECT-TYPE**

SYNTAX Integer32 (0..2147483647)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The current state of ccmCTIDeviceTable. The initial value of this object is 0 and it will be incremented every time when there is a change (addition/deletion/modification) to the ccmCTIDeviceTable. This value and ccmCallManagerStartTime should be used together to find if the table has changed or not. When the call manager is restarted, this will be reset to 0.

::= { ccmGlobalInfo 25 }

#### **ccmCTIDeviceDirNumTableStateId OBJECT-TYPE**

SYNTAX Integer32 (0..2147483647)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The current state of ccmCTIDeviceDirNumTable. The initial value of this object is 0 and it will be incremented every time when there is a change (addition/deletion/modification) to the ccmCTIDeviceDirNumTable. This value and ccmCallManagerStartTime should be used together to find if the table has changed or not. When the call manager is restarted, this will be reset to 0.

::= { ccmGlobalInfo 26 }

#### **ccmPhStatUpdtTblLastAddedIndex OBJECT-TYPE**

SYNTAX CcmIndexOrZero

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The ccmPhoneStatusUpdateIndex value of the last entry that was added to the ccmPhoneStatusUpdateTable. This value together with sysUpTime can be used by the manager applications to identify the new entries in the ccmPhoneStatusUpdateTable since their last poll. This value need not be the same as the highest index in the ccmPhoneStatusUpdateTable as the index could have wrapped around. The initial value of this object is 0, which indicates that no entries have been added to this table. When the SNMP service is restarted this value will be reset to 0.

::= { ccmGlobalInfo 27 }

#### **ccmPhFailedTblLastAddedIndex OBJECT-TYPE**

SYNTAX CcmIndexOrZero

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The ccmPhoneFailedIndex value of the last entry that was added to the ccmPhoneFailedTable. This value together with sysUpTime can be used by the manager applications to identify the new entries in the ccmPhoneFailedTable since their last poll. This value need not be the same as the highest index in the ccmPhoneFailedTable as the index could have wrapped around. The initial value of this object is 0, which indicates that no entries have been added to this table. When the SNMP service is restarted this value will be reset to 0.

::= { ccmGlobalInfo 28 }

**ccmSystemVersion OBJECT-TYPE**

SYNTAX SnmpAdminString (SIZE(0..128))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The installed version of the local Cisco Unified CM system.

::= { ccmGlobalInfo 29 }

**ccmInstallationId OBJECT-TYPE**

SYNTAX SnmpAdminString (SIZE(0..128))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The installation component identifier of the local Cisco Unified CM component(ccm.exe).

::= { ccmGlobalInfo 30 }

**ccmPartiallyRegisteredPhones OBJECT-TYPE**

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The number of phones that are partially registered with the local Cisco Unified CM.

::= { ccmGlobalInfo 31 }

**ccmH323TableEntries OBJECT-TYPE**

SYNTAX Integer32 (0..2147483647)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The current number of entries in ccmH323DeviceTable. The initial value of this object is 0 and it will be incremented every time when there is an addition to the ccmH323DeviceTable. When the Cisco Unified CM is restarted, this will be reset to 0.

::= { ccmGlobalInfo 32 }

**ccmSIPTableEntries OBJECT-TYPE**

SYNTAX Integer32 (0..2147483647)

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

The current number of entries in ccmSIPDeviceTable. The initial value of this object is 0 and it will be incremented every time when there is an addition to the ccmSIPDeviceTable. When the Cisco Unified CM is restarted, this will be reset to zero.

::= { ccmGlobalInfo 33 }

## Media Device Table

### ccmMediaDeviceTable OBJECT-TYPE

SYNTAX SEQUENCE OF CcmMediaDeviceEntry

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

The table containing a list of all Media Devices that have tried to register with the local Cisco Unified CM at least once. When the local Cisco Unified CM is restarted, this table will be refreshed.

::= { ccmMediaDeviceInfo 1 }

### ccmMediaDeviceEntry OBJECT-TYPE

SYNTAX CcmMediaDeviceEntry

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

An entry (conceptual row) in the MediaDevice Table, containing the information about a particular Media Resource device.

INDEX { ccmMediaDeviceIndex }

::= { ccmMediaDeviceTable 1 }

CcmMediaDeviceEntry ::= SEQUENCE {

ccmMediaDeviceIndex CcmIndex,

ccmMediaDeviceName SnmpAdminString,

ccmMediaDeviceType INTEGER,

ccmMediaDeviceDescription

SnmpAdminString,

ccmMediaDeviceStatus CcmDeviceStatus,

ccmMediaDeviceDevicePoolIndex CcmIndexOrZero,

ccmMediaDeviceInetAddressType InetAddressType,

ccmMediaDeviceInetAddress InetAddress,

ccmMediaDeviceStatusReason CcmDevFailCauseCode,

ccmMediaDeviceTimeLastStatusUpdt DateAndTime,

```

ccmMediaDeviceTimeLastRegistered DateAndTime,
ccmMediaDeviceProductTypeIndex CcmIndexOrZero
ccmMediaDeviceInetAddressIPv4 InetAddressIPv4,
ccmMediaDeviceInetAddressIPv6 InetAddressIPv6,
ccmMediaDeviceUnregReason CcmDevUnregCauseCode,
ccmMediaDeviceRegFailReason CcmDevRegFailCauseCode
}

```

**ccmMediaDeviceIndex OBJECT-TYPE**

SYNTAX CcmIndex

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

An arbitrary integer, selected by the local Cisco Unified CM, that identifies a Media Device entry in the table.

::= { ccmMediaDeviceEntry 1 }

**ccmMediaDeviceName OBJECT-TYPE**

SYNTAX SnmpAdminString (SIZE(0..128))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This is the device name assigned to the Media Device. This name is assigned when a new device of this type is added to the Cisco Unified CM.

::= { ccmMediaDeviceEntry 2 }

**ccmMediaDeviceDescription OBJECT-TYPE**

SYNTAX SnmpAdminString (SIZE(0..128))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This description is given when the device is configured in the Cisco Unified CM.

::= { ccmMediaDeviceEntry 4 }

**ccmMediaDeviceStatus OBJECT-TYPE**

SYNTAX CcmDeviceStatus

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The status of the Media Device. The status changes from unknown to registered when it registers itself with the local Cisco Unified CM.

::= { ccmMediaDeviceEntry 5 }

#### **ccmMediaDeviceDevicePoolIndex OBJECT-TYPE**

SYNTAX CcmIndexOrZero

MAX-ACCESS read-only

STATUS current

DESCRIPTION

A positive value of this index is used to identify the Device Pool to which this MediaDevice entry belongs. A value of zero indicates that the index to the Device Pool table is Unknown.

::= { ccmMediaDeviceEntry 6 }

#### **ccmMediaDeviceTimeLastStatusUpdt OBJECT-TYPE**

SYNTAX DateAndTime

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The time the status of the media device changed.

::= { ccmMediaDeviceEntry 10 }

#### **ccmMediaDeviceTimeLastRegistered OBJECT-TYPE**

SYNTAX DateAndTime

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The time the media device last registered with the call manager.

::= { ccmMediaDeviceEntry 11 }

#### **ccmMediaDeviceProductTypeIndex OBJECT-TYPE**

SYNTAX CcmIndexOrZero

MAX-ACCESS read-only

STATUS current

DESCRIPTION

A positive value of this index is used to identify the related product type entry in the ccmProductTypeTable. A value of zero indicates that the index to the ccmProductTypeTable is Unknown.

::= { ccmMediaDeviceEntry 12 }

#### **ccmMediaDeviceInetAddressIPv4 OBJECT-TYPE**

SYNTAX InetAddressIPv4

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object identifies the last known primary IPv4 address of the Media Device. This object contains value zero if IPV4 address is not available.

::= { ccmMediaDeviceEntry 13 }

#### **ccmMediaDeviceInetAddressIPv6 OBJECT-TYPE**

SYNTAX InetAddressIPv6

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object identifies the last known primary IPv6 address of the Media Device. This object contains value zero if IPV6 address is not available.

::= { ccmMediaDeviceEntry 14 }

#### **ccmMediaDeviceUnregReason OBJECT-TYPE**

SYNTAX CcmDevUnregCauseCode

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The reason code associated with unregistered Media Device.

::= { ccmMediaDeviceEntry 15 }

#### **ccmMediaDeviceRegFailReason OBJECT-TYPE**

SYNTAX CcmDevRegFailCauseCode

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The reason code associated with registration failed Media Device.

::= { ccmMediaDeviceEntry 16 }

## CTI Device Table

#### **ccmCTIDeviceTable OBJECT-TYPE**

SYNTAX SEQUENCE OF CcmCTIDeviceEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION—The table containing a list of all CTI (Computer Telephony Integration) Devices that have tried to register with the local Cisco Unified CM at least once. When the local Cisco Unified CM is restarted, this table will be refreshed.

::= { ccmCTIDeviceInfo 1 }

#### **ccmCTIDeviceEntry OBJECT-TYPE**

SYNTAX CcmCTIDeviceEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION—An entry (conceptual row) in the CTIDevice Table, containing the information about a particular CTI Device.

INDEX { ccmCTIDeviceIndex }

::= { ccmCTIDeviceTable 1 }

CcmCTIDeviceEntry ::= SEQUENCE {  
 ccmCTIDeviceIndexCcmIndex,  
 ccmCTIDeviceNameSnmppAdminString,  
 ccmCTIDeviceTypeINTEGER,  
 ccmCTIDeviceDescriptionSnmppAdminString,  
 ccmCTIDeviceStatusCcmDeviceStatus,  
 ccmCTIDevicePoolIndexCcmIndexOrZero,  
 ccmCTIDeviceInetAddressType [DEPRECATED]InetAddressType,  
 ccmCTIDeviceInetAddress [DEPRECATED]InetAddress,  
 ccmCTIDeviceAppInfoSnmppAdminString,  
 ccmCTIDeviceStatusReasonCcmDevFailCauseCode,  
 ccmCTIDeviceTimeLastStatusUpdtDateAndTime,  
 ccmCTIDeviceTimeLastRegisteredDateAndTime,  
 ccmCTIDeviceProductTypeIndexCcmIndexOrZero  
 ccmCTIDeviceInetAddressIPv4InetAddressIPv4  
 ccmCTIDeviceInetAddressIPv6InetAddressIPv6  
 ccmCTIDeviceUnregReason CcmDevUnregCauseCode,  
 ccmCTIDeviceRegFailReasonCcmDevRegFailCauseCode  
 }

#### **ccmCTIDeviceIndex OBJECT-TYPE**

SYNTAX CcmIndex

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

An arbitrary integer, selected by the local Cisco Unified CM, that identifies a CTI Device entry in the table.

::= { ccmCTIDeviceEntry 1 }

**ccmCTIDeviceName OBJECT-TYPE**

SYNTAX SnmpAdminString (SIZE(0..64))

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

The name of the CTI Device. This name is assigned to the CTI Device when it is added to the Cisco Unified CM.

::= { ccmCTIDeviceEntry 2 }

**ccmCTIDeviceDescription OBJECT-TYPE**

SYNTAX SnmpAdminString (SIZE(0..128))

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

A description of the CTI Device. This description is given when the CTI Device is configured in the Cisco Unified CM.

::= { ccmCTIDeviceEntry 4 }

**ccmCTIDeviceStatus OBJECT-TYPE**

SYNTAX CcmDeviceStatus

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

The status of the CTI Device. The CTI Device status changes from unknown to registered when it registers itself with the local Cisco Unified CM.

::= { ccmCTIDeviceEntry 5 }

**ccmCTIDevicePoolIndex OBJECT-TYPE**

SYNTAX CcmIndexOrZero

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

A positive value of this index is used to identify the Device Pool to which this CTI Device entry belongs. A value of zero indicates that the index to the Device Pool table is Unknown.

::= { ccmCTIDeviceEntry 6 }

**ccmCTIDeviceTimeLastStatusUpdt OBJECT-TYPE**

SYNTAX DateAndTime

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The time the status of the CTI device changed.

::= { ccmCTIDeviceEntry 11 }

**ccmCTIDeviceTimeLastRegistered OBJECT-TYPE**

SYNTAX DateAndTime

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The time the CTI Device last registered with the call manager.

::= { ccmCTIDeviceEntry 12 }

**ccmCTIDeviceProductTypeIndex OBJECT-TYPE**

SYNTAX CcmIndexOrZero

MAX-ACCESS read-only

STATUS current

DESCRIPTION

A positive value of this index is used to identify the related product type entry in the ccmProductTypeTable. A value of 0 indicates that the index to the ccmProductTypeTable is Unknown.

::= { ccmCTIDeviceEntry 13 }

**ccmCTIDeviceInetAddressIPv4 OBJECT-TYPE**

SYNTAX InetAddressIPv4

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object identifies IPv4 Address of the host where this CTI Device is running. If the IPv4 address is not available then this object contains an empty string.

::= { ccmCTIDeviceEntry 14 }

**ccmCTIDeviceInetAddressIPv6 OBJECT-TYPE**

SYNTAX InetAddressIPv6

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object identifies IPv6 Address of the host where this CTI Device is running. If the IPv6 address is not available then this object contains an empty string.

::= { ccmCTIDeviceEntry 15 }

#### **ccmCTIDeviceUnregReason OBJECT-TYPE**

SYNTAX CcmDevUnregCauseCode

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The reason code associated with unregistered CTI Device.

::= { ccmCTIDeviceEntry 16 }

#### **ccmCTIDeviceRegFailReason OBJECT-TYPE**

SYNTAX CcmDevRegFailCauseCode

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The reason code associated with registration failed CTI Device.

::= { ccmCTIDeviceEntry 17 }

## **CTI Device Directory Number Table**

#### **ccmCTIDeviceDirNumTable OBJECT-TYPE**

SYNTAX SEQUENCE OF CcmCTIDeviceDirNumEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

The table containing a list of directory numbers that are assigned to all of the registered and unregistered CTI Devices in the ccmCTIDeviceTable.

::= { ccmCTIDeviceInfo 2 }

#### **ccmCTIDeviceDirNumEntry OBJECT-TYPE**

SYNTAX CcmCTIDeviceDirNumEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

An entry (conceptual row) in the CTIDeviceDirNum Table, containing the information about a particular CTI Device extension.

INDEX { ccmCTIDeviceIndex, ccmCTIDeviceDirNumIndex }

::= { ccmCTIDeviceDirNumTable 1 }

```
CcmCTIDeviceDirNumEntry ::= SEQUENCE {
    ccmCTIDeviceDirNumIndex CcmIndex,
    ccmCTIDeviceDirNum SnmpAdminString
}
```

**ccmCTIDeviceDirNumIndex OBJECT-TYPE**

```
SYNTAX CcmIndex
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
    An arbitrary integer, selected by the local system, that identifies a Directory Number of a CTI Device.
 ::= { ccmCTIDeviceDirNumEntry 1 }
```

**ccmCTIDeviceDirNum OBJECT-TYPE**

```
SYNTAX SnmpAdminString (SIZE(0..24))
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    A Directory Number of the CTI Device.
 ::= { ccmCTIDeviceDirNumEntry 2 }
--
```

# Alarms

## Cisco Unified CM Alarm Enable

**ccmCallManagerAlarmEnable OBJECT-TYPE**

```
SYNTAX TruthValue
MAX-ACCESS read-write
STATUS current
DESCRIPTION
    Allows the generation of alarms in response to Cisco Unified CM general failures.
    true(1): Enabling this object will allow the Cisco Unified CM agent to generate the following alarms:
        ccmCallManagerFailure,
        ccmMediaResourceListExhausted,
        ccmRouteListExhausted and
        ccmTLSConnectionFailure. This is the default value.
    false(2): Disabling this object will stop the generation of the following alarms by the Cisco Unified CM agent:
```

ccmCallManagerFailure  
 ccmMediaResourceListExhausted,  
 ccmRouteListExhausted and  
 ccmTLSConnectionFailure.

DEFVAL { true }

::= { ccmAlarmConfigInfo 1 }

## Phone Failed Config Objects

### ccmPhoneFailedAlarmInterval OBJECT-TYPE

SYNTAX Integer32 (0 | 30..3600)

UNITS seconds

MAX-ACCESS read-write

STATUS current

DESCRIPTION

The minimum interval between sending of the ccmPhoneFailed notification in seconds. The ccmPhoneFailed notification is only sent when there is at least one entry in the ccmPhoneFailedTable and the notification has not been sent for the last ccmPhoneFailedAlarmInterval defined in this object. A value of zero indicates that the alarm notification is disabled.

DEFVAL { 0 }

::= { ccmAlarmConfigInfo 2 }

### ccmPhoneFailedStorePeriod OBJECT-TYPE

SYNTAX Integer32 (1800..3600)

UNITS seconds

MAX-ACCESS read-write

STATUS current

DESCRIPTION

The time duration for storing each entry in the ccmPhoneFailedTable. The entries that have not been updated and kept at least this period will be deleted. This value should ideally be set to a higher value than the ccmPhoneFailedAlarmInterval object.

DEFVAL { 1800 }

::= { ccmAlarmConfigInfo 3 }

## Phone Status Update Config Objects

### ccmPhoneStatusUpdateAlarmInterval OBJECT-TYPE

SYNTAX Integer32 (0 | 30..3600)

UNITS seconds

MAX-ACCESS read-write

STATUS current

## DESCRIPTION

The minimum interval between sending of the `ccmPhoneStatusUpdate` notification in seconds. The `ccmPhoneStatusUpdate` notification is only sent when there is at least one entry in the `ccmPhoneStatusUpdateTable` and the notification has not been sent for the last `ccmPhoneStatusUpdateAlarmInterv` defined in this object. A value of zero indicates that the alarm notification is disabled.

DEFVAL { 0 }

::= { `ccmAlarmConfigInfo` 4 }

**ccmPhoneStatusUpdateStorePeriod OBJECT-TYPE**

SYNTAX Integer32 (1800..3600)

UNITS seconds

MAX-ACCESS read-write

STATUS current

## DESCRIPTION

The time duration for storing each entry in the `ccmPhoneStatusUpdateTable`. The entries that have been kept at least this period will be deleted. This value should ideally be set to a higher value than the `ccmPhoneStatusUpdateAlarmInterv` object.

DEFVAL { 1800 }

::= { `ccmAlarmConfigInfo` 5 }

**Gateway Alarm Enable****ccmGatewayAlarmEnable OBJECT-TYPE**

SYNTAX TruthValue

MAX-ACCESS read-write

STATUS current

## DESCRIPTION

Allows the generation of alarms in response to Gateway general failures that the Cisco Unified CM is aware of.

true(1): Enabling this object will allow the Cisco Unified CM agent to generate the following alarms:

- `ccmGatewayFailedReason`
- `ccmGatewayLayer2Change` (This is the default value.)

false(2): Disabling this object will stop the generation of the following alarms by the Cisco Unified agent:

- `ccmGatewayFailed`
- `ccmGatewayLayer2Change`.

DEFVAL { true }

::= { `ccmAlarmConfigInfo` 6 }

## Malicious Call Alarm Enable

### ccmMaliciousCallAlarmEnable OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-write

STATUS current

DESCRIPTION

Allows the generation of alarms for malicious calls that the local call manager is aware of.

true(1): Enabling this object will allow the Cisco Unified CM agent to generate the ccmMaliciousCall alarm. This is the default value.

false(2): Disabling this object will stop the generation of the ccmMaliciousCall alarm.

DEFVAL { true }

::= { ccmAlarmConfigInfo 7 }

## Notification and Alarms

### ccmAlarmSeverity OBJECT-TYPE

SYNTAX INTEGER {

emergency(1),

alert(2),

critical(3),

error(4),

warning(5),

notice(6),

informational(7)

}

MAX-ACCESS accessible-for-notify

STATUS current

DESCRIPTION

The Alarm Severity code.

emergency: System unusable

alert: Immediate response needed

critical: Critical condition

error: Error condition

warning: Warning condition

notice: Normal but significant condition

informational: Informational situation.

::= { ccmNotificationsInfo 1 }

**ccmFailCauseCode OBJECT-TYPE**

```

SYNTAX INTEGER {
  unknown(1),
  heartBeatStopped(2),
  routerThreadDied(3),
  timerThreadDied(4),
  criticalThreadDied(5),
  deviceMgrInitFailed(6),
  digitAnalysisInitFailed(7),
  callControlInitFailed(8),
  linkMgrInitFailed(9),
  dbMgrInitFailed(10),
  msgTranslatorInitFailed(11),
  suppServicesInitFailed(12)
}

```

MAX-ACCESS accessible-for-notify

STATUS current

DESCRIPTION

The Cause code of the failure. This cause is derived from a monitoring thread in the Cisco Unified CM or from a heartbeat monitoring process.

unknown: Unknown

heartBeatStopped: The Cisco Unified CM stops generating a heartbeat

routerThreadDied: The Cisco Unified CM detects the death of the router thread

timerThreadDied: The Cisco Unified CM detects the death of the timer thread

criticalThreadDied: The Cisco Unified CM detects the death of one of its critical threads

deviceMgrInitFailed: The Cisco Unified CM fails to start its device manager subsystem

digitAnalysisInitFailed: The Cisco Unified CM fails to start its digit analysis subsystem

callControlInitFailed: The Cisco Unified CM fails to start its call control subsystem

linkMgrInitFailed: The Cisco Unified CM fails to start its link manager subsystem

dbMgrInitFailed: The Cisco Unified CM fails to start its database manager subsystem

msgTranslatorInitFailed: The Cisco Unified CM fails to start its message translation manager subsystem

suppServicesInitFailed: The Cisco Unified CM fails to start its supplementary services subsystem.

::= { ccmNotificationsInfo 2 }

**ccmPhoneFailures OBJECT-TYPE**

```

SYNTAX Unsigned32

```

MAX-ACCESS accessible-for-notify

STATUS current

## DESCRIPTION

The count of the phone initialization or communication failures that are stored in the ccmPhoneFailedTable object.

::= { ccmNotificationsInfo 3 }

**ccmPhoneUpdates OBJECT-TYPE**

SYNTAX Unsigned32

MAX-ACCESS accessible-for-notify

STATUS current

## DESCRIPTION

The count of the phone status changes that are stored in the ccmPhoneStatusUpdateTable object.

::= { ccmNotificationsInfo 4 }

**ccmMediaResourceType OBJECT-TYPE**

SYNTAX INTEGER {

unknown(1),

mediaTerminationPoint(2),

transcoder(3),

conferenceBridge(4),

musicOnHold(5)

}

MAX-ACCESS accessible-for-notify

STATUS current

## DESCRIPTION

The type of media resource.

unknown: Unknown resource type

mediaTerminationPoint: Media Termination Point

transcoder: Transcoder

conferenceBridge: Conference Bridge

musicOnHold: Music On Hold.

::= { ccmNotificationsInfo 6 }

**ccmMediaResourceListName OBJECT-TYPE**

SYNTAX SnmpAdminString (SIZE(0..128))

MAX-ACCESS accessible-for-notify

STATUS current

## DESCRIPTION

The name of a Media Resource List. This name is assigned when a new Media Resource List is added to the Cisco Unified CM.

::= { ccmNotificationsInfo 7 }

**ccmRouteListName OBJECT-TYPE**

SYNTAX SnmpAdminString (SIZE(0..128))

MAX-ACCESS accessible-for-notify

STATUS current

DESCRIPTION

The name of a Route List. This name is assigned when a new Route List is added to the Cisco Unified CM.

::= { ccmNotificationsInfo 8 }

**ccmGatewayPhysIfIndex OBJECT-TYPE**

SYNTAX Integer32 (1..2147483647)

MAX-ACCESS accessible-for-notify

STATUS current

DESCRIPTION

This object is the identifier of an interface in a gateway that has registered with the local Cisco Unified CM. On a DS1/E1 interface, this should be the same as the ifIndex value in the gateway.

::= { ccmNotificationsInfo 9 }

**ccmGatewayPhysIfL2Status OBJECT-TYPE**

SYNTAX INTEGER {

unknown(1),

up(2),

down(3)

}

MAX-ACCESS accessible-for-notify

STATUS current

DESCRIPTION

The layer 2 status of a physical interface in a gateway that has registered with the local Cisco Unified CM.

unknown: Unknown status

up: Interface is up

down: Interface is down.

::= { ccmNotificationsInfo 10 }

**ccmMaliCallCalledPartyName OBJECT-TYPE**

SYNTAX SnmpAdminString (SIZE(0..128))

MAX-ACCESS accessible-for-notify

STATUS current

## DESCRIPTION

The display name of the called party who received the malicious call.

::= { ccmNotificationsInfo 11 }

**ccmMaliCallCalledPartyNumber OBJECT-TYPE**

SYNTAX SnmpAdminString (SIZE(0..128))

MAX-ACCESS accessible-for-notify

STATUS current

## DESCRIPTION

The phone number of the device where the malicious call is received.

::= { ccmNotificationsInfo 12 }

**ccmMaliCallCalledDeviceName OBJECT-TYPE**

SYNTAX SnmpAdminString (SIZE(0..128))

MAX-ACCESS accessible-for-notify

STATUS current

## DESCRIPTION

The name of the device where the malicious call is received.

::= { ccmNotificationsInfo 13 }

**ccmMaliCallCallingPartyName OBJECT-TYPE**

SYNTAX SnmpAdminString (SIZE(0..128))

MAX-ACCESS accessible-for-notify

STATUS current

## DESCRIPTION

The display name of the caller whose call is registered as malicious with the local call manager.

::= { ccmNotificationsInfo 14 }

**ccmMaliCallCallingPartyNumber OBJECT-TYPE**

SYNTAX SnmpAdminString (SIZE(0..128))

MAX-ACCESS accessible-for-notify

STATUS current

## DESCRIPTION

The phone number of the caller whose call is registered as malicious with the local call manager.

::= { ccmNotificationsInfo 15 }

**ccmMaliCallCallingDeviceName OBJECT-TYPE**

SYNTAX SnmpAdminString (SIZE(0..128))

MAX-ACCESS accessible-for-notify

STATUS current

DESCRIPTION

The edge device name through which the malicious call originated or passed through.

::= { ccmNotificationsInfo 16 }

**ccmMaliCallTime OBJECT-TYPE**

SYNTAX DateAndTime

MAX-ACCESS accessible-for-notify

STATUS current

DESCRIPTION

The time when the malicious call is detected by the local call manager.

::= { ccmNotificationsInfo 17 }

**ccmQualityRprtSourceDevName OBJECT-TYPE**

SYNTAX SnmpAdminString (SIZE(0..128))

MAX-ACCESS accessible-for-notify

STATUS current

DESCRIPTION

The name of the source device from where the problem was reported.

::= { ccmNotificationsInfo 18 }

**ccmQualityRprtClusterId OBJECT-TYPE**

SYNTAX SnmpAdminString (SIZE(0..128))

MAX-ACCESS accessible-for-notify

STATUS current

DESCRIPTION

The cluster identifier of the source device.

::= { ccmNotificationsInfo 19 }

**ccmQualityRprtCategory OBJECT-TYPE**

SYNTAX SnmpAdminString (SIZE(0..128))

MAX-ACCESS accessible-for-notify

STATUS current

DESCRIPTION

The category of the problem reported.

::= { ccmNotificationsInfo 20 }

**ccmQualityRprtReasonCode OBJECT-TYPE**

SYNTAX SnmpAdminString (SIZE(0..128))

MAX-ACCESS accessible-for-notify

STATUS current

DESCRIPTION

The description of the problem reported.

::= { ccmNotificationsInfo 21 }

#### **ccmQualityRprtTime OBJECT-TYPE**

SYNTAX DateAndTime

MAX-ACCESS accessible-for-notify

STATUS current

DESCRIPTION

The time when the problem was reported.

::= { ccmNotificationsInfo 22 }

#### **ccmTLSDevName OBJECT-TYPE**

SYNTAX SnmpAdminString (SIZE(0..128))

MAX-ACCESS accessible-for-notify

STATUS current

DESCRIPTION

The device for which TLS connection failure was reported.

::= { ccmNotificationsInfo 23 }

#### **ccmTLSDevInetAddressType OBJECT-TYPE**

SYNTAX InetAddressType

MAX-ACCESS accessible-for-notify

STATUS current

DESCRIPTION

This object identifies the type of address for the device for which TLS connection failure was reported.

::= { ccmNotificationsInfo 24 }

#### **ccmTLSDevInetAddress OBJECT-TYPE**

SYNTAX InetAddress

MAX-ACCESS accessible-for-notify

STATUS current

DESCRIPTION

This object identifies IP Address of the device, for which TLS connection failure was reported. The type of address for this is identified by ccmTLSDevInetAddressType.

::= { ccmNotificationsInfo 25 }

**ccmTLSConnFailTime OBJECT-TYPE**

SYNTAX DateAndTime

MAX-ACCESS accessible-for-notify

STATUS current

DESCRIPTION

The time when TLS connection failure was detected by the local Cisco Unified CM.

::= { ccmNotificationsInfo 26 }

**ccmTLSConnectionFailReasonCode OBJECT-TYPE**

SYNTAX INTEGER {

unknown (1),

authenticationerror(2),

invalidx509nameincertificate(3),

invalidtlscipher(4)

}

MAX-ACCESS accessible-for-notify

STATUS current

DESCRIPTION

The reason for connection failure.

::= { ccmNotificationsInfo 27 }

**ccmGatewayRegFailCauseCode OBJECT-TYPE**

SYNTAX CcmDevRegFailCauseCode

MAX-ACCESS accessible-for-notify

STATUS current

DESCRIPTION

States the reason for a gateway device registration failure.

::= { ccmNotificationsInfo 28 }

**H323 Device Table****ccmH323DeviceTable OBJECT-TYPE**

SYNTAX SEQUENCE OF CcmH323DeviceEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

The table containing a list of all H323 devices in the Cisco Unified CM cluster that the local Cisco Unified CM is aware of. When the local Cisco Unified CM is restarted, this table will be refreshed.

```
::= { ccmH323DeviceInfo 1 }
```

### **ccmH323DeviceEntry OBJECT-TYPE**

SYNTAX CcmH323DeviceEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

An entry (conceptual row) in the H323Device Table, containing the information about a particular H323 Device.

INDEX { ccmH323DevIndex }

```
::= { ccmH323DeviceTable 1 }
```

CcmH323DeviceEntry ::= SEQUENCE {

ccmH323DevIndex CcmIndex,

ccmH323DevName SnmpAdminString,

ccmH323DevProductId CcmDeviceProductId,

ccmH323DevDESCRIPTION SnmpAdminString,

ccmH323DevInetAddressType InetAddressType,

ccmH323DevInetAddress InetAddress,

ccmH323DevCnfgGKInetAddressType InetAddressType,

ccmH323DevCnfgGKInetAddress InetAddress,

ccmH323DevAltGK1InetAddressType InetAddressType,

ccmH323DevAltGK1InetAddress InetAddress,

ccmH323DevAltGK2InetAddressType InetAddressType,

ccmH323DevAltGK2InetAddress InetAddress,

ccmH323DevAltGK3InetAddressType InetAddressType,

ccmH323DevAltGK3InetAddress InetAddress,

ccmH323DevAltGK4InetAddressType InetAddressType,

ccmH323DevAltGK4InetAddress InetAddress,

ccmH323DevAltGK5InetAddressType InetAddressType,

ccmH323DevAltGK5InetAddress InetAddress,

ccmH323DevActGKInetAddressType InetAddressType,

ccmH323DevActGKInetAddress InetAddress,

ccmH323DevStatus INTEGER,

ccmH323DevStatusReason CcmDevFailCauseCode,

ccmH323DevTimeLastStatusUpdt DateAndTime,

ccmH323DevTimeLastRegistered DateAndTime,

```

ccmH323DevRmtCM1InetAddressType InetAddressType,
ccmH323DevRmtCM1InetAddress InetAddress,
ccmH323DevRmtCM2InetAddressType InetAddressType,
ccmH323DevRmtCM2InetAddress InetAddress,
ccmH323DevRmtCM3InetAddressType InetAddressType,
ccmH323DevRmtCM3InetAddress InetAddress,
ccmH323DevProductTypeIndex CcmIndexOrZero
ccmH323DevUnregReason CcmDevUnregCauseCode,
ccmH323DevRegFailReason CcmDevRegFailCauseCode
}

```

**ccmH323DevIndex OBJECT-TYPE**

SYNTAX CcmIndex

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

An arbitrary integer, selected by the local Cisco Unified CM, that identifies a H323 Device entry in the table.

::= { ccmH323DeviceEntry 1 }

**ccmH323DevName OBJECT-TYPE**

SYNTAX SnmpAdminString (SIZE(0..128))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The device name assigned to the H323 Device. This name is assigned when a new H323 device is added to the Cisco Unified CM.

::= { ccmH323DeviceEntry 2 }

**ccmH323DevDESCRIPTION OBJECT-TYPE**

SYNTAX SnmpAdminString (SIZE(0..255))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

A DESCRIPTION

A description of the H323 device. This description is given when the H323 device is configured in the Cisco Unified CM.

::= { ccmH323DeviceEntry 4 }

**ccmH323DevInetAddressType OBJECT-TYPE**

SYNTAX InetAddressType

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object identifies the IP address type of the H323 device. The value of this object is 'unknown(0)' if the IP address of a H323 device is not available.

::= { ccmH323DeviceEntry 5 }

**ccmH323DevInetAddress OBJECT-TYPE**

SYNTAX InetAddress

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object identifies last known IP Address of the H323 device. If the IP address is not available then this object contains an empty string. The type of address for this is identified by ccmH323DevInetAddressType.

::= { ccmH323DeviceEntry 6 }

**ccmH323DevCnfgGKInetAddressType OBJECT-TYPE**

SYNTAX InetAddressType

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object identifies the IP address type of the gatekeeper device. The value of this object is 'unknown(0)' if the IP address of a H323 gatekeeper is not available.

::= { ccmH323DeviceEntry 7 }

**ccmH323DevCnfgGKInetAddress OBJECT-TYPE**

SYNTAX InetAddress

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object represents configured gatekeeper DNS name or IP address for this H323 device. This is applicable only for H323 devices with gatekeepers configured. When there is no H323 gatekeeper configured, this object contains an empty string. The type of address for this is identified by ccmH323DevCnfgGKInetAddressType.

::= { ccmH323DeviceEntry 8 }

**ccmH323DevAltGK1InetAddressType OBJECT-TYPE**

SYNTAX InetAddressType

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object identifies the IP address type of the first alternate gatekeeper. The value of this object is 'unknown(0)' if the IP address of a H323 gatekeeper is not available.

::= { ccmH323DeviceEntry 9 }

#### **ccmH323DevAltGK1InetAddress OBJECT-TYPE**

SYNTAX InetAddress

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object identifies the first alternate gatekeeper DNS name or IP address for this H323 device. This is applicable only for H323 devices with gatekeepers configured. When there is no first alternate H323 gatekeeper, this object contains an empty string. The type of address for this is identified by ccmH323DevAltGK1InetAddressType.

::= { ccmH323DeviceEntry 10 }

#### **ccmH323DevAltGK2InetAddressType OBJECT-TYPE**

SYNTAX InetAddressType

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object identifies the IP address type of the second alternate gatekeeper. The value of this object is 'unknown(0)' if the IP address of a H323 gatekeeper is not available.

::= { ccmH323DeviceEntry 11 }

#### **ccmH323DevAltGK2InetAddress OBJECT-TYPE**

SYNTAX InetAddress

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object identifies the second alternate gatekeeper DNS name or IP address for this H323 device. This is applicable only for H323 devices with gatekeepers configured. When there is no second alternate H323 gatekeeper, this object contains an empty string. The type of address for this is identified by ccmH323DevAltGK2InetAddressType.

::= { ccmH323DeviceEntry 12 }

#### **ccmH323DevAltGK3InetAddressType OBJECT-TYPE**

SYNTAX InetAddressType

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object identifies the IP address type of the third alternate gatekeeper. The value of this object is 'unknown(0)' if the IP address of a H323 gatekeeper is not available.

::= { ccmH323DeviceEntry 13 }

#### **ccmH323DevAltGK3InetAddress OBJECT-TYPE**

SYNTAX InetAddress

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object identifies the third alternate gatekeeper DNS name or IP address for this H323 device. This is applicable only for H323 devices with gatekeepers configured. When there is no third alternate H323 gatekeeper, this object contains an empty string. The type of address for this is identified by ccmH323DevAltGK3InetAddressType.

::= { ccmH323DeviceEntry 14 }

#### **ccmH323DevAltGK4InetAddressType OBJECT-TYPE**

SYNTAX InetAddressType

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object identifies the IP address type of the fourth alternate gatekeeper. The value of this object is 'unknown(0)' if the IP address of a H323 gatekeeper is not available.

::= { ccmH323DeviceEntry 15 }

#### **ccmH323DevAltGK4InetAddress OBJECT-TYPE**

SYNTAX InetAddress

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object identifies the fourth alternate gatekeeper DNS name or IP address for this H323 device. This is applicable only for H323 devices with gatekeepers configured. When there is no fourth H323 alternate gatekeeper, this object contains an empty string. The type of address for this is identified by ccmH323DevAltGK4InetAddressType.

::= { ccmH323DeviceEntry 16 }

#### **ccmH323DevAltGK5InetAddressType OBJECT-TYPE**

SYNTAX InetAddressType

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

This object identifies the IP address type of the fifth alternate gatekeeper. The value of this object is 'unknown(0)' if the IP address of a H323 gatekeeper is not available.

::= { ccmH323DeviceEntry 17 }

**ccmH323DevAltGK5InetAddress OBJECT-TYPE**

SYNTAX InetAddress

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

This object identifies the fifth alternate gatekeeper DNS name or IP address for this H323 device. This is applicable only for H323 devices with gatekeepers configured. When there is no fifth H323 alternate gatekeeper, this object contains an empty string. The type of address for this is identified by ccmH323DevAltGK5InetAddressType.

::= { ccmH323DeviceEntry 18 }

**ccmH323DevActGKInetAddressType OBJECT-TYPE**

SYNTAX InetAddressType

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

This object identifies the IP address type of the active gatekeeper. The value of this object is 'unknown(0)' if the IP address of a gatekeeper is not available.

::= { ccmH323DeviceEntry 19 }

**ccmH323DevActGKInetAddress OBJECT-TYPE**

SYNTAX InetAddress

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

This object identifies the active alternate gatekeeper DNS name or IP address for this H323 device. This is applicable only for H323 devices with gatekeepers configured. When there is no active alternate H323 gatekeeper, this object contains an empty string. The type of address for this is identified by ccmH323DevActGKInetAddressType.

::= { ccmH323DeviceEntry 20 }

**ccmH323DevStatus OBJECT-TYPE**

SYNTAX INTEGER {

notApplicable(0),

unknown(1),

registered(2),

unregistered(3),

rejected(4)

}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The H323 device registration status with the gatekeeper. The status changes from unknown to registered when the H323 device successfully registers itself with the gatekeeper.

notApplicable: The registration status is not applicable for this H323 device

unknown: The registration status of the H323 device with the gatekeeper is unknown

registered: The H323 device has registered with the gatekeeper successfully

unregistered: The H323 device is no longer registered with the gatekeeper

rejected: Registration request from the H323 device was rejected by the gatekeeper.

::= { ccmH323DeviceEntry 21 }

#### **ccmH323DevTimeLastStatusUpdt OBJECT-TYPE**

SYNTAX DateAndTime

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The time the registration status with the gatekeeper changed. This is applicable only for H323 devices with gatekeepers configured.

::= { ccmH323DeviceEntry 23 }

#### **ccmH323DevTimeLastRegistered OBJECT-TYPE**

SYNTAX DateAndTime

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The time when the H323 device last registered with the gatekeeper. This is applicable only for H323 devices with gatekeepers configured.

::= { ccmH323DeviceEntry 24 }

#### **ccmH323DevRmtCM1InetAddressType OBJECT-TYPE**

SYNTAX InetAddressType

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object identifies the IP address type of the first remote call manager. The value of this object is

'unknown(0)' if the first remote call manager is not configured.

::= { ccmH323DeviceEntry 25 }

#### **ccmH323DevRmtCM1InetAddress OBJECT-TYPE**

SYNTAX InetAddress

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object identifies the first remote call manager DNS name or IP address configured for this H323 device. When there is no first remote call manager configured, this object contains an empty string. The type of address for this is identified by ccmH323DevRmtCM1InetAddressType.

::= { ccmH323DeviceEntry 26 }

#### **ccmH323DevRmtCM2InetAddressType OBJECT-TYPE**

SYNTAX InetAddressType

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object identifies the IP address type of the second remote call manager. The value of this object is 'unknown(0)' if the second remote call manager is not configured.

::= { ccmH323DeviceEntry 27 }

#### **ccmH323DevRmtCM2InetAddress OBJECT-TYPE**

SYNTAX InetAddress

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object identifies the second remote call manager DNS name or IP address configured for this H323 device. When there is no second remote call manager configured, this object contains an empty string. The type of address for this is identified by ccmH323DevRmtCM2InetAddressType.

::= { ccmH323DeviceEntry 28 }

#### **ccmH323DevRmtCM3InetAddressType OBJECT-TYPE**

SYNTAX InetAddressType

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object identifies the IP address type of the third remote call manager. The value of this object is 'unknown(0)' if the third remote call manager is not configured.

::= { ccmH323DeviceEntry 29 }

**ccmH323DevRmtCM3InetAddress OBJECT-TYPE**

SYNTAX InetAddress

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object identifies the third remote call manager DNS name or IP address configured for this H323 device. When there is no third remote call manager configured, this object contains an empty string. The type of address for this is identified by ccmH323DevRmtCM3InetAddressType.

::= { ccmH323DeviceEntry 30 }

**ccmH323DevProductTypeIndex OBJECT-TYPE**

SYNTAX CcmIndexOrZero

MAX-ACCESS read-only

STATUS current

DESCRIPTION

A positive value of this index is used to identify the related product type entry in the ccmProductTypeTable. A value of zero indicates that the index to the ccmProductTypeTable is Unknown.

::= { ccmH323DeviceEntry 31 }

**ccmH323DevUnregReason OBJECT-TYPE**

SYNTAX CcmDevUnregCauseCode

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The reason code associated with unregistered H323 Device. This is applicable only for H323 devices with gatekeepers configured.

::= { ccmH323DeviceEntry 32 }

**ccmH323DevRegFailReason OBJECT-TYPE**

SYNTAX CcmDevRegFailCauseCode

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The reason code associated with registration failed H323 Device. This is applicable only for H323 devices with gatekeepers configured.

::= { ccmH323DeviceEntry 33 }

## Voice Mail Device Table

### **ccmVoiceMailDeviceTable OBJECT-TYPE**

SYNTAX SEQUENCE OF CcmVoiceMailDeviceEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

The table containing a list of all voice messaging devices that have tried to register with the local Cisco Unified CM at least once. When the local Cisco Unified CM is restarted, this table will be refreshed.

::= { ccmVoiceMailDeviceInfo 1 }

### **ccmVoiceMailDeviceEntry OBJECT-TYPE**

SYNTAX CcmVoiceMailDeviceEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

An entry (conceptual row) in the VoiceMailDevice Table, containing the information about a particular Voice Messaging Device.

INDEX { ccmVMailDevIndex }

::= { ccmVoicMailDeviceTable 1 }

CcmVoiceMailDeviceEntry ::= SEQUENCE {

ccmVMailDevIndex CcmIndex,

ccmVMailDevName SnmpAdminString,

ccmVMailDevProductId CcmDeviceProductId,

ccmVMailDevDescription, SnmpAdminString,

ccmVMailDevStatus CcmDeviceStatus,

ccmVMailDevInetAddressType InetAddressType,

ccmVMailDevInetAddress InetAddress,

ccmVMailDevStatusReason CcmDevFailCauseCode,

ccmVMailDevTimeLastStatusUpdt DateAndTime,

ccmVMailDevTimeLastRegistered DateAndTime,

ccmVMailDevProductTypeIndex CcmIndexOrZero

ccmVMailDevUnregReason CcmDevUnregCauseCode,

ccmVMailDevRegFailReason CcmDevRegFailCauseCode

}

### **ccmVMailDevIndex OBJECT-TYPE**

SYNTAX CcmIndex

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

An arbitrary integer, selected by the local Cisco Unified CM, that identifies a voice messaging device entry in the table.

::= { ccmVoiceMailDeviceEntry 1 }

**ccmVMailDevName OBJECT-TYPE**

SYNTAX SnmpAdminString (SIZE(0..128))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The name of the Voice Messaging Device. This name is assigned to the Voice Messaging Device when it is added to the Cisco Unified CM.

::= { ccmVoiceMailDeviceEntry 2 }

**ccmVMailDevDESCRIPTION OBJECT-TYPE**

SYNTAX SnmpAdminString (SIZE(0..255))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The description of the Voice Messaging Device. This description is given when the Voice Messaging Device is configured in the Cisco Unified CM.

::= { ccmVoiceMailDeviceEntry 4 }

**ccmVMailDevStatus OBJECT-TYPE**

SYNTAX CcmDeviceStatus

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The status of the Voice Messaging Device. The Voice Messaging Device status changes from unknown to registered when it registers itself with the local Cisco Unified CM.

::= { ccmVoiceMailDeviceEntry 5 }

**ccmVMailDevInetAddressType OBJECT-TYPE**

SYNTAX InetAddressType

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object identifies the IP address type of the Voice Messaging device. The value of this object is 'unknown(0)' if the IP address of the Voice Messaging device is not available.

::= { ccmVoiceMailDeviceEntry 6 }

#### **ccmVMailDevInetAddress OBJECT-TYPE**

SYNTAX InetAddress

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object identifies the IP Address of the Voice Messaging Device. If the IP Address is not available then this object contains an empty string. The type of address for this is identified by ccmVMailDevInetAddressType.

::= { ccmVoiceMailDeviceEntry 7 }

#### **ccmVMailDevTimeLastStatusUpdt OBJECT-TYPE**

SYNTAX DateAndTime

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The time the status of the voice messaging device changed.

::= { ccmVoiceMailDeviceEntry 9 }

#### **ccmVMailDevTimeLastRegistered OBJECT-TYPE**

SYNTAX DateAndTime

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The time the Voice Messaging Device has last registered with the call manager.

::= { ccmVoiceMailDeviceEntry 10 }

#### **ccmVMailDevProductTypeIndex OBJECT-TYPE**

SYNTAX CcmIndexOrZero

MAX-ACCESS read-only

STATUS current

DESCRIPTION

A positive value of this index is used to identify the related product type entry in the ccmProductTypeTable. A value of 0 indicates that the index to the ccmProductTypeTable is Unknown.

::= { ccmVoiceMailDeviceEntry 11 }

#### **ccmVMailDevUnregReason OBJECT-TYPE**

SYNTAX CcmDevUnregCauseCode

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The reason code associated with unregistered Voice Messaging Device.

::= { ccmVoiceMailDeviceEntry 12 }

#### **ccmVMailDevRegFailReason OBJECT-TYPE**

SYNTAX CcmDevRegFailCauseCode

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The reason code associated with registration failed Voice Messaging Device.

::= { ccmVoiceMailDeviceEntry 13 }

## **Voice Mail Directory Number Table**

#### **ccmVoiceMailDeviceDirNumTable OBJECT-TYPE**

SYNTAX SEQUENCE OF CcmVoiceMailDeviceDirNumEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

The table containing a list of directory numbers that are assigned to all of the registered and unregistered Voice Messaging Devices in the ccmVoiceMailDeviceTable.

::= { ccmVoiceMailDeviceInfo 2 }

#### **ccmVoiceMailDeviceDirNumEntry OBJECT-TYPE**

SYNTAX CcmVoiceMailDeviceDirNumEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

An entry (conceptual row) in the VoiceMailDirNum Table, has the associated directory number for a Voice Messaging Device.

INDEX { ccmVMailDevIndex, ccmVMailDevDirNumIndex }

::= { ccmVoiceMailDeviceDirNumTable 1 }

CcmVoiceMailDeviceDirNumEntry ::= SEQUENCE {

ccmVMailDevDirNumIndexCcmIndex,

ccmVMailDevDirNum SnmpAdminString

}

**ccmVMailDevDirNumIndex OBJECT-TYPE**

SYNTAX CcmIndex

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

An arbitrary integer, selected by the local system, that identifies a Directory Number of a Voice Messaging Device.

::= { ccmVoiceMailDeviceDirNumEntry 1 }

**ccmVMailDevDirNum OBJECT-TYPE**

SYNTAX SnmpAdminString (SIZE(0..24))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The Directory Number of the Voice Messaging Device.

::= { ccmVoiceMailDeviceDirNumEntry 2 }

**Quality Report Alarm Configuration Information****ccmQualityReportAlarmEnable OBJECT-TYPE**

SYNTAX TruthValue

MAX-ACCESS read-write

STATUS current

DESCRIPTION

Allows the generation of the quality report alarm.

true(1): Enabling this object will allow the Cisco Unified CM agent to generate the ccmQualityReport alarm. This is the default value.

false(2): Disabling this object will stop the generation of the ccmQualityReport alarm by the Cisco Unified CM agent.

DEFVAL { true }

::= { ccmQualityReportAlarmConfigInfo 1 }

**Sip Device Table****ccmSIPDeviceTable OBJECT-TYPE**

SYNTAX SEQUENCE OF CcmSIPDeviceEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

The table containing a list of all SIP trunks in the Cisco Unified CM cluster that the local Cisco Unified CM is aware of. When the local Cisco Unified CM is restarted, this table will be refreshed. If the local Cisco Unified CM is down, then this table will be empty.

```
::= { ccmSIPDeviceInfo 1 }
```

#### **ccmSIPDeviceEntry OBJECT-TYPE**

SYNTAX CcmSIPDeviceEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

An entry (conceptual row) in the SIP Device Table, containing the information about a particular SIP Trunk Device.

INDEX { ccmSIPDevIndex }

```
::= { ccmSIPDeviceTable 1 }
```

CcmSIPDeviceEntry ::= SEQUENCE {

ccmSIPDevIndex CcmIndex,

ccmSIPDevName SnmpAdminString,

ccmSIPDevProductTypeIndex CcmIndexOrZero,

ccmSIPDevDescription SnmpAdminString,

ccmSIPDevInetAddressType InetAddressType,

ccmSIPDevInetAddress InetAddress,

ccmSIPInTransportProtocolType CcmSIPTransportProtocolType,

ccmSIPInPortNumber InetPortNumber,

ccmSIPOutTransportProtocolType CcmSIPTransportProtocolType,

ccmSIPOutPortNumber InetPortNumber

ccmSIPDevInetAddressIPv4 InetAddressIPv4,

ccmSIPDevInetAddressIPv6 InetAddressIPv6

}

#### **ccmSIPDevIndex OBJECT-TYPE**

SYNTAX CcmIndex

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

An arbitrary integer, selected by the local Cisco Unified CM, that identifies a SIP Trunk Device entry in the table.

```
::= { ccmSIPDeviceEntry 1 }
```

**ccmSIPDevName OBJECT-TYPE**

SYNTAX SnmpAdminString (SIZE(0..128))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The device name assigned to the SIP Trunk Device. This name is assigned when a new SIP Trunk device is added to the Cisco Unified CM.

::= { ccmSIPDeviceEntry 2 }

**ccmSIPDevProductTypeIndex OBJECT-TYPE**

SYNTAX CcmIndexOrZero

MAX-ACCESS read-only

STATUS current

DESCRIPTION

A positive value of this index is used to identify the related product type entry in the ccmProductTypeTable. A value of zero indicates that the index to the ccmProductTypeTable is Unknown.

::= { ccmSIPDeviceEntry 3 }

**ccmSIPDevDescription OBJECT-TYPE**

SYNTAX SnmpAdminString (SIZE(0..255))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

A description of the SIP Trunk device. This Description is given when the SIP Trunk device is configured in the Cisco Unified CM.

::= { ccmSIPDeviceEntry 4 }

**ccmSIPInTransportProtocolType OBJECT-TYPE**

SYNTAX CcmSIPTransportProtocolType

MAX-ACCESS read-only

STATUS current

DESCRIPTION

Specifies the transport protocol type used by Cisco Unified CM for setting up incoming SIP call.

::= { ccmSIPDeviceEntry 7 }

**ccmSIPInPortNumber OBJECT-TYPE**

SYNTAX InetPortNumber

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

Specifies the port number used by Cisco Unified CM for setting up incoming SIP call.

::= { ccmSIPDeviceEntry 8 }

**ccmSIPOutTransportProtocolType OBJECT-TYPE**

SYNTAX CcmSIPTransportProtocolType

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

Specifies the transport protocol type used by Cisco Unified CM for setting up outgoing SIP call.

::= { ccmSIPDeviceEntry 9 }

**ccmSIPOutPortNumber OBJECT-TYPE**

SYNTAX InetPortNumber

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

Specifies the port number used by Cisco Unified CM for setting up outgoing SIP call.

::= { ccmSIPDeviceEntry 10 }

**ccmSIPDevInetAddressIPv4 OBJECT-TYPE**

SYNTAX InetAddressIPv4

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

This object identifies the last known primary IPv4 address of the SIP Trunk Device. This object contains value zero if IPV4 address is not available.

::= { ccmSIPDeviceEntry 11 }

**ccmSIPDevInetAddressIPv6 OBJECT-TYPE**

SYNTAX InetAddressIPv6

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

This object identifies the last known primary IPv6 address of the SIP Trunk Device. This object contains value zero if IPV6 address is not available.

::= { ccmSIPDeviceEntry 12 }

## Notifications Types

### **ccmMIBNotificationPrefix OBJECT IDENTIFIER**

::= { ciscoCcmMIB 2 }

### **ccmMIBNotifications OBJECT IDENTIFIER**

::= { ccmMIBNotificationPrefix 0 }

### **ccmCallManagerFailed NOTIFICATION-TYPE**

OBJECTS {

ccmAlarmSeverity,

ccmFailCauseCode

}

STATUS current

DESCRIPTION

This Notification signifies that the Cisco Unified CM process detects a failure in one of its critical subsystems. It can also be detected from a heartbeat/event monitoring process.

::= { ccmMIBNotifications 1 }

### **ccmPhoneFailed NOTIFICATION-TYPE**

OBJECTS {

ccmAlarmSeverity,

ccmPhoneFailures

}

STATUS current

DESCRIPTION

This Notification will be generated in the intervals specified in `ccmPhoneFailedAlarmInterval` if there is at least one entry in the `ccmPhoneFailedTable`.

::= { ccmMIBNotifications 2 }

### **ccmPhoneStatusUpdate NOTIFICATION-TYPE**

OBJECTS {

ccmAlarmSeverity,

ccmPhoneUpdates

}

STATUS current

DESCRIPTION

This Notification will be generated in the intervals specified in `ccmPhoneStatusUpdateInterv` if there is at least one entry in the `ccmPhoneStatusUpdateTable`.

::= { ccmMIBNotifications 3 }

**ccmMediaResourceListExhausted NOTIFICATION-TYPE**

```
OBJECTS {  
  ccmAlarmSeverity,  
  ccmMediaResourceType,  
  ccmMediaResourceListName  
}
```

STATUS current

**DESCRIPTION**

This Notification indicates that the Cisco Unified CM has run out a certain specified type of resource.

```
::= { ccmMIBNotifications 5 }
```

**ccmRouteListExhausted NOTIFICATION-TYPE**

```
OBJECTS {  
  ccmAlarmSeverity,  
  ccmRouteListName  
}
```

STATUS current

**DESCRIPTION**

This Notification indicates that the Cisco Unified CM could not find an available route in the indicated route list.

```
::= { ccmMIBNotifications 6 }
```

**ccmGatewayLayer2Change NOTIFICATION-TYPE**

```
OBJECTS {  
  ccmAlarmSeverity,  
  ccmGatewayName,  
  ccmGatewayInetAddressType,  
  ccmGatewayInetAddress,  
  ccmGatewayPhysIfIndex,  
  ccmGatewayPhysIfL2Status  
}
```

STATUS current

**DESCRIPTION**

This Notification is sent when the D-Channel/Layer 2 of an interface in a skinny gateway that has registered with the Cisco Unified CM changes state.

```
::= { ccmMIBNotifications 7 }
```

**ccmMaliciousCall NOTIFICATION-TYPE**

OBJECTS {

ccmAlarmSeverity,  
 ccmMaliCallCalledPartyName,  
 ccmMaliCallCalledPartyNumber,  
 ccmMaliCallCalledDeviceName,  
 ccmMaliCallCallingPartyName,  
 ccmMaliCallCallingPartyNumber,  
 ccmMaliCallCallingDeviceName,  
 ccmMaliCallTime

}

STATUS current

DESCRIPTION

This Notification is sent when a user registers a call as malicious with the local call manager.

::= { ccmMIBNotifications 8 }

**ccmQualityReport NOTIFICATION-TYPE**

OBJECTS {

ccmAlarmSeverity,  
 ccmQualityRprtSourceDevName,  
 ccmQualityRprtClusterId,  
 ccmQualityRprtCategory,  
 ccmQualityRprtReasonCode,  
 ccmQualityRprtTime

}

STATUS current

DESCRIPTION

This Notification is sent when a user reports a quality problem using the Quality Report Tool.

::= { ccmMIBNotifications 9 }

**ccmTLSConnectionFailure NOTIFICATION-TYPE**

OBJECTS {

ccmAlarmSeverity,  
 ccmTLSDevName,  
 ccmTLSDevInetAddressType,  
 ccmTLSDevInetAddress,

```

ccmTLSConnectionFailReasonCode,
ccmTLSConnFailTime
}

```

STATUS current

DESCRIPTION

This Notification is sent when Cisco Unified CM fails to open TLS connection for the indicated device.

```
 ::= { ccmMIBNotifications 10 }
```

#### **ccmGatewayFailedReason NOTIFICATION-TYPE**

```
OBJECTS {
```

```

ccmAlarmSeverity,
ccmGatewayName,
ccmGatewayInetAddressType,
ccmGatewayInetAddress,
ccmGatewayRegFailCauseCode
}

```

STATUS current

DESCRIPTION

This Notification indicates that at least one gateway has attempted to register or communicate with the Cisco Unified CM and failed.

```
 ::= { ccmMIBNotifications 11 }
```

## **MIB Conformance Statements**

#### **ciscoCcmMIBConformance OBJECT IDENTIFIER**

```
 ::= { ciscoCcmMIB 3 }
```

#### **ciscoCcmMIBCompliances OBJECT IDENTIFIER**

```
 ::= { ciscoCcmMIBConformance 1 }
```

#### **ciscoCcmMIBGroups OBJECT IDENTIFIER**

```
 ::= { ciscoCcmMIBConformance 2 }
```

## **Compliance Statements**

#### **ciscoCcmMIBComplianceRev7 MODULE-COMPLIANCE**

STATUS current

DESCRIPTION

The compliance statement for entities that implement the Cisco Unified CM MIB.

```
MANDATORY-GROUPS {
```

```

ccmInfoGroupRev4,

```

```

ccmPhoneInfoGroupRev6,
ccmGatewayInfoGroupRev4,
ccmMediaDeviceInfoGroupRev4,
ccmCTIDeviceInfoGroupRev4,
ccmNotificationsInfoGroupRev5,
ccmNotificationsGroupRev3,
ccmH323DeviceInfoGroupRev3,
ccmVoiceMailDeviceInfoGroupRev2,
ccmSIPDeviceInfoGroupRev2
}
::= { ciscoCcmMIBCompliances 8 }

```

## Units of Conformance

### **ccmMediaDeviceInfoGroupRev2 OBJECT-GROUP**

OBJECTS {

```

ccmMediaDeviceName,
ccmMediaDeviceDescription,
ccmMediaDeviceStatus,
ccmMediaDeviceDevicePoolIndex,
ccmMediaDeviceInetAddressType,
ccmMediaDeviceInetAddress,
ccmMediaDeviceStatusReason,
ccmMediaDeviceTimeLastStatusUpdt,
ccmMediaDeviceTimeLastRegistered,
ccmMediaDeviceProductTypeIndex,
ccmRegisteredMediaDevices,
ccmUnregisteredMediaDevices,
ccmRejectedMediaDevices
}

```

STATUS current

DESCRIPTION

A collection of objects that provide info about all Media Devices within the scope of the local Cisco Unified CM. It comprises of the MediaDevice table.

```

::= { ciscoCcmMIBGroups 26 }

```

**ccmCTIDeviceInfoGroupRev2 OBJECT-GROUP**

```
OBJECTS {
  ccmCTIDeviceName,
  ccmCTIDeviceDescription,
  ccmCTIDeviceStatus,
  ccmCTIDevicePoolIndex,
  ccmCTIDeviceInetAddressType,
  ccmCTIDeviceInetAddress,
  ccmCTIDeviceStatusReason,
  ccmCTIDeviceTimeLastStatusUpdt,
  ccmCTIDeviceTimeLastRegistered,
  ccmCTIDeviceProductTypeIndex,
  ccmCTIDeviceDirNum,
  ccmRegisteredCTIDevices,
  ccmUnregisteredCTIDevices,
  ccmRejectedCTIDevices,
  ccmCTIDeviceTableStateId,
  ccmCTIDeviceDirNumTableStateId
}
```

STATUS current

## DESCRIPTION

A collection of objects that provide info about all CTI Devices within the scope of the local Cisco Unified CM. It comprises of the ccmCTIDevice and ccmCTIDeviceDirNum tables.

```
::= { ciscoCcmMIBGroups 27 }
```

**ccmInfoGroupRev4 OBJECT-GROUP**

```
OBJECTS {
  ccmGroupName,
  ccmGroupTftpDefault,
  ccmName,
  ccmDescription,
  ccmVersion,
  ccmStatus,
  ccmInetAddressType,
  ccmInetAddress,
```

```

ccmClusterId,
ccmCMGroupMappingCMPriority,
ccmRegionName,
ccmRegionAvailableBandWidth,
ccmTimeZoneName,
ccmTimeZoneOffsetHours,
ccmTimeZoneOffsetMinutes,
ccmDevicePoolName,
ccmDevicePoolRegionIndex,
ccmDevicePoolTimeZoneIndex,
ccmDevicePoolGroupIndex,
ccmProductType,
ccmProductName,
ccmProductCategory,
ccmCallManagerStartTime,
ccmSystemVersion,
ccmInstallationId,
ccmInetAddress2Type,
ccmInetAddress2
}

```

STATUS current

DESCRIPTION

A collection of objects that provide information about all Cisco Unified Communications Managers and its related information within a Cisco Unified CM cluster. It comprises of GroupTable, ccmTable, GroupMappingTable, Region, TimeZone, Device Pool and ProductType tables.

```
::= { ciscoCcmMIBGroups 34 }
```

### **ccmSIPDeviceInfoGroupRev2 OBJECT-GROUP**

OBJECTS {

```

ccmSIPDevName,
ccmSIPDevProductTypeIndex,
ccmSIPDevDescription,
ccmSIPInTransportProtocolType,
ccmSIPInPortNumber,
ccmSIPOutTransportProtocolType,
ccmSIPOutPortNumber,

```

```
ccmSIPDevInetAddressIPv4,  
ccmSIPDevInetAddressIPv6,  
ccmSIPTableEntries  
}
```

STATUS current

DESCRIPTION

A collection of objects that provide information about all SIP Trunk devices within the scope of the local Cisco Unified Communications Manager. It comprises of the SIP Device table.

```
::= { ciscoCcmMIBGroups 37 }
```

### **ccmPhoneInfoGroupRev6 OBJECT-GROUP**

OBJECTS {

```
ccmPhonePhysicalAddress,  
ccmPhoneDescription,  
ccmPhoneUserName,  
ccmPhoneStatus,  
ccmPhoneTimeLastRegistered,  
ccmPhoneE911Location,  
ccmPhoneLoadID,  
ccmPhoneDevicePoolIndex,  
ccmPhoneTimeLastStatusUpdt,  
ccmPhoneProductTypeIndex,  
ccmPhoneProtocol,  
ccmPhoneName,  
ccmPhoneExtn,  
ccmPhoneExtnMultiLines,  
ccmPhoneExtnInetAddressType,  
ccmPhoneExtnInetAddress,  
ccmPhoneExtnStatus,  
ccmRegisteredPhones,  
ccmUnregisteredPhones,  
ccmRejectedPhones,  
ccmPartiallyRegisteredPhones,  
ccmPhoneTableStateId,  
ccmPhoneExtensionTableStateId,
```

```

ccmPhoneInetAddressIPv4,
ccmPhoneInetAddressIPv6,
ccmPhoneIPv4Attribute,
ccmPhoneIPv6Attribute,
ccmPhoneActiveLoadID,
ccmPhoneUnregReason,
ccmPhoneRegFailReason
}

```

STATUS current

DESCRIPTION

A collection of objects that provide information about all phones within the scope of the local Cisco Unified Communications Manager. It comprises of the Phone and Phone Extension tables.

```
::= { ciscoCcmMIBGroups 41 }
```

#### **ccmNotificationsInfoGroupRev5 OBJECT-GROUP**

OBJECTS {

```

ccmAlarmSeverity,
ccmCallManagerAlarmEnable,
ccmFailCauseCode,
ccmPhoneFailures,
ccmPhoneFailedTime,
ccmPhoneFailedMacAddress,
ccmPhoneFailedAlarmInterval,
ccmPhoneFailedStorePeriod,
ccmPhFailedTblLastAddedIndex,
ccmPhoneUpdates,
ccmPhoneStatusPhoneIndex,
ccmPhoneStatusUpdateTime,
ccmPhoneStatusUpdateType,
ccmPhoneStatusUpdateAlarmInterv,
ccmPhoneStatusUpdateStorePeriod,
ccmPhoneStatusUpdateTableStateId,
ccmPhStatUpdtTblLastAddedIndex,
ccmGatewayAlarmEnable,
ccmMediaResourceType,

```

```
ccmMediaResourceListName,  
ccmRouteListName,  
ccmGatewayPhysIfIndex,  
ccmGatewayPhysIfL2Status,  
ccmMaliciousCallAlarmEnable,  
ccmMaliCallCalledPartyName,  
ccmMaliCallCalledPartyNumber,  
ccmMaliCallCalledDeviceName,  
ccmMaliCallCallingPartyName,  
ccmMaliCallCallingPartyNumber,  
ccmMaliCallCallingDeviceName,  
ccmMaliCallTime,  
ccmQualityReportAlarmEnable,  
ccmQualityRprtSourceDevName,  
ccmQualityRprtClusterId,  
ccmQualityRprtCategory,  
ccmQualityRprtReasonCode,  
ccmQualityRprtTime,  
ccmTLSDevName,  
ccmTLSDevInetAddressType,  
ccmTLSDevInetAddress,  
ccmTLSConnFailTime,  
ccmTLSConnectionFailReasonCode,  
ccmPhoneFailedInetAddressIPv4,  
ccmPhoneFailedInetAddressIPv6,  
ccmPhoneFailedIPv4Attribute,  
ccmPhoneFailedIPv6Attribute,  
ccmPhoneFailedRegFailReason,  
ccmPhoneStatusUnregReason,  
ccmPhoneStatusRegFailReason,  
ccmGatewayRegFailCauseCode  
}  
STATUS current  
DESCRIPTION
```

A collection of objects that provide information about all the Notifications generated by the Cisco Unified CM Agent.

```
::= { ciscoCcmMIBGroups 42 }
```

#### **ccmGatewayInfoGroupRev4 OBJECT-GROUP**

OBJECTS {

```
ccmGatewayName,
ccmGatewayDescription,
ccmGatewayStatus,
ccmGatewayDevicePoolIndex,
ccmGatewayInetAddressType,
ccmGatewayInetAddress,
ccmGatewayTimeLastStatusUpdt,
ccmGatewayTimeLastRegistered,
ccmGatewayDChannelStatus,
ccmGatewayDChannelNumber,
ccmGatewayProductTypeIndex,
ccmRegisteredGateways,
ccmUnregisteredGateways,
ccmRejectedGateways,
ccmGatewayTableStateId,
ccmGatewayUnregReason,
ccmGatewayRegFailReason
}
```

STATUS current

DESCRIPTION

A collection of objects that provide information about all Gateways within the scope of the local Cisco Unified CM. It comprises of the Gateway table.

```
::= { ciscoCcmMIBGroups 43 }
```

#### **ccmMediaDeviceInfoGroupRev4 OBJECT-GROUP**

OBJECTS {

```
ccmMediaDeviceName,
ccmMediaDeviceDescription,
ccmMediaDeviceStatus,
ccmMediaDeviceDevicePoolIndex,
```

```

ccmMediaDeviceTimeLastStatusUpdt,
ccmMediaDeviceTimeLastRegistered,
ccmMediaDeviceProductTypeIndex,
ccmRegisteredMediaDevices,
ccmUnregisteredMediaDevices,
ccmRejectedMediaDevices,
ccmMediaDeviceInetAddressIPv4,
ccmMediaDeviceInetAddressIPv6,
ccmMediaDeviceUnregReason,
ccmMediaDeviceRegFailReason
}

```

STATUS current

DESCRIPTION

A collection of objects that provide information about all Media Devices within the scope of the local Cisco Unified Communications Manager. It comprises of the MediaDevice table.

```
::= { ciscoCcmMIBGroups 44 }
```

#### **ccmCTIDeviceInfoGroupRev4 OBJECT-GROUP**

```

OBJECTS {
ccmCTIDeviceName,
ccmCTIDeviceDescription,
ccmCTIDeviceStatus,
ccmCTIDevicePoolIndex,
ccmCTIDeviceTimeLastStatusUpdt,
ccmCTIDeviceTimeLastRegistered,
ccmCTIDeviceProductTypeIndex,
ccmCTIDeviceDirNum,
ccmRegisteredCTIDevices,
ccmUnregisteredCTIDevices,
ccmRejectedCTIDevices,
ccmCTIDeviceTableStateId,
ccmCTIDeviceDirNumTableStateId,
ccmCTIDeviceInetAddressIPv4,
ccmCTIDeviceInetAddressIPv6,
ccmCTIDeviceUnregReason,

```

```
ccmCTIDeviceRegFailReason
}
```

STATUS current

DESCRIPTION

A collection of objects that provide information about all CTI Devices within the scope of the local Cisco Unified CM. It comprises of the ccmCTIDevice and ccmCTIDeviceDirNum tables.

```
::= { ciscoCcmMIBGroups 45 }
```

### **ccmH323DeviceInfoGroupRev3 OBJECT-GROUP**

OBJECTS {

```
ccmH323DevName,
ccmH323DevDescription,
ccmH323DevInetAddressType,
ccmH323DevInetAddress,
ccmH323DevCnfgGKInetAddressType,
ccmH323DevCnfgGKInetAddress,
ccmH323DevAltGK1InetAddressType,
ccmH323DevAltGK1InetAddress,
ccmH323DevAltGK2InetAddressType,
ccmH323DevAltGK2InetAddress,
ccmH323DevAltGK3InetAddressType,
ccmH323DevAltGK3InetAddress,
ccmH323DevAltGK4InetAddressType,
ccmH323DevAltGK4InetAddress,
ccmH323DevAltGK5InetAddressType,
ccmH323DevAltGK5InetAddress,
ccmH323DevActGKInetAddressType,
ccmH323DevActGKInetAddress,
ccmH323DevStatus,
ccmH323DevTimeLastStatusUpdt,
ccmH323DevTimeLastRegistered,
ccmH323DevRmtCM1InetAddressType,
ccmH323DevRmtCM1InetAddress,
ccmH323DevRmtCM2InetAddressType,
ccmH323DevRmtCM2InetAddress,
```

```

ccmH323DevRmtCM3InetAddressType,
ccmH323DevRmtCM3InetAddress,
ccmH323DevProductTypeIndex,
ccmH323TableEntries,
ccmH323DevUnregReason,
ccmH323DevRegFailReason
}

```

STATUS current

DESCRIPTION

A collection of objects that provide information about all H323 devices within the scope of the local Cisco Unified Communications Manager. It comprises of the H323Device table.

```
::= { ciscoCcmMIBGroups 46 }
```

### **ccmVoiceMailDevicelInfoGroupRev2 OBJECT-GROUP**

```
OBJECTS {
```

```

ccmVMailDevName,
ccmVMailDevDescription,
ccmVMailDevStatus,
ccmVMailDevInetAddressType,
ccmVMailDevInetAddress,
ccmVMailDevTimeLastStatusUpdt,
ccmVMailDevTimeLastRegistered,
ccmVMailDevProductTypeIndex,
ccmVMailDevDirNum,
ccmRegisteredVoiceMailDevices,
ccmUnregisteredVoiceMailDevices,
ccmRejectedVoiceMailDevices,
ccmVMailDevUnregReason,
ccmVMailDevRegFailReason
}

```

STATUS current

DESCRIPTION

A collection of objects that provide information about all Voice Messaging Devices within the scope of the local Cisco Unified CM. It comprises of the ccmVoiceMailDevice and ccmVoiceMailDirNum tables.

```
::= { ciscoCcmMIBGroups 47 }
```

**ccmNotificationsGroupRev3 NOTIFICATION-GROUP**

```
NOTIFICATIONS {
```

```
  ccmCallManagerFailed,
```

```
  ccmPhoneFailed,
```

```
  ccmPhoneStatusUpdate,
```

```
  ccmGatewayFailedReason,
```

```
  ccmMediaResourceListExhausted,
```

```
  ccmRouteListExhausted,
```

```
  ccmGatewayLayer2Change,
```

```
  ccmMaliciousCall,
```

```
  ccmQualityReport,
```

```
  ccmTLSConnectionFailure
```

```
}
```

```
STATUS current
```

```
DESCRIPTION
```

A collection of notifications that are generated by the Cisco Unified CM Agent.

```
::= { ciscoCcmMIBGroups 48 }
```

## Cisco Unified CM Managed Services and SNMP Traps

The services that are provided in Cisco Unified Serviceability and the SNMP trap components to which they track are described in the following table.

*Table 2: Cisco Unified CM Managed Services, Alarms/Notifications, and Trap Components*

Cisco Unified CM managed service in CISCO-CCM-MIB	Alarm/Notifications	Trap components
Cisco Unified CM Failure	ccmCallManagerFailed	ccmAlarmSeverity ccmFailCauseCode
Gateway Failure	ccmGatewayFailed <b>Note</b> ccmGatewayFailed is deprecated and replaced by ccmGatewayFailedReason.	ccmAlarmSeverity ccmGatewayName ccmGatewayInetAddressType ccmGatewayInetAddress ccmGatewayFailCauseCode
Cisco Unified CM Phones	ccmPhoneFailed	ccmAlarmSeverity ccmPhoneFailures

Cisco Unified CM managed service in CISCO-CCM-MIB	Alarm/Notifications	Trap components
Cisco Unified CM Media Resources	ccmMediaResourceListExhausted	ccmAlarmSeverity ccmMediaResourceType ccmMediaResourceListName
Cisco Unified CM Route List	ccmRouteListExhausted	
Gateway Layer 2 Change	ccmGatewayLayer2Change	
Malicious Call Status	ccmMaliciousCall	
Quality Report	ccmQualityReport	
TLS Connection Failure	ccmTLSConnectionFailure	

## Cisco Unified CM Alarms to Enable

Enabling the `ccmCallManagerAlarmEnable` object in the CISCO-CCM-MIB allows the Cisco Unified CM agent to generate traps and send the following alarms:

- `ccmCallManagerFailed`
- `ccmGatewayFailed`
- `ccmPhoneFailed`
- `ccmMediaResourceListExhausted`
- `ccmRouteListExhausted`
- `ccmGatewayLayer2Change`
- `ccmMaliciousCall`
- `ccmQualityReport`
- `ccmTLSConnectionFailure`

## Traps to Monitor

The following are Cisco Unified CM traps to monitor:

- `ccmCallManagerFailed`. This trap means that Cisco Unified CM has detected a failure in one of its critical subsystems. It can also be detected from a heartbeat/event monitoring process. The OID is 1.3.6.1.4.1.9.9.156.2.0.1. The trap components are `ccmAlarmSeverity` and `ccmFailCauseCode`.
  - `ccmAlarmSeverity` OID is 1.3.6.1.4.1.9.9.156.1.10.1. The values are:
    - 1—Emergency
    - 2—Alert
    - 3—Critical

4—Error

5—Warning

6—Notice

7—Informational

- `ccmFailCauseCode` is derived from a monitoring thread in the Cisco Unified CM or from a heartbeat monitoring process. OID is 1.3.6.1.4.1.9.9.156.1.10.2. The values are:

1—Unknown

2—Heart Beat Stopped

3—Router Thread Died

4—Timer Thread Died

5—Critical Thread Died

6—Device MgrInit Failed

7—Digit Analysis Init Failed

8—Call Control Init Failed

9—Link Mgr Init Failed

10—DB Mgr Init Failed

11—Msg Translator Init Failed

12—Supp Services Init Failed

- Cisco Phone Failures—`CISCO-CCM-MIB::ccmPhoneFailed`. This notification is generated in the intervals specified in `ccmPhoneFailedAlarmInterval` if there is at least one entry in the `ccmPhoneFailedTable`. The OID is 1.3.6.1.4.1.9.9.156.2.0.2. The trap components are `ccmAlarmSeverity` and `ccmPhoneFailures`. See `ccmAlarmSeverity` for more information. The `ccmPhoneFailures` OID is 1.3.6.1.4.1.9.9.156.1.10.3 and the `ccmPhoneFailedTable` should be checked for phone initialization and communication failures.
- Cisco Unified CM Gateway Failure—`CISCO-CCM-MIB::ccmGatewayFailed`. This notification indicates that at least one gateway has attempted to register or communicate with the Cisco Unified CM and failed. The OID is 1.3.6.1.4.1.9.9.156.2.0.4. The trap components are:
  - `ccmAlarmSeverity` OID is 1.3.6.1.4.1.9.9.156.1.10.1. The values are:
    - 1—Emergency
    - 2—Alert
    - 3—Critical
    - 4—Error
    - 5—Warning
    - 6—Notice
    - 7—Informational
  - `ccmGatewayFailCauseCode` OID is 1.3.6.1.4.1.9.9.156.1.10.5. The type is `CcmDevFailCauseCode` and contains the following values:

- 0—No Error
- 1—Unknown
- 2—No Entry In Database
- 3—Database Configuration Error
- 4—Device Name Unresolveable
- 5—Max Dev Reg Reached
- 6—Connectivity Error
- 7—Initialization Error
- 8—Device Initiated Reset
- 9—Cisco Unified CM Reset
- 10—Authentication Error
- 11—Invalid X509 Name In Certificate
- 12—Invalid TLS Cipher
- 13—Directory Number Mismatch
- 14—Malformed Register Msg



---

**Note** CcmDevFailCauseCode is deprecated and replaced by CcmDevRegFailCauseCode and CcmDevUnregCauseCode.

---

- Cisco Unified CM Media Resource Exhausted—CISCO-CCM-MIB::ccmMediaResourceListExhausted. This notification indicates that Cisco Unified CM has run out of a certain specified type of resource. The OID is 1.3.6.1.4.1.9.9.156.2.0.5. The critical trap components are:
  - ccmAlarmSeverity OID is 1.3.6.1.4.1.9.9.156.1.10.1. The values are:
    - 1—Emergency
    - 2—Alert
    - 3—Critical
    - 4—Error
    - 5—Warning
    - 6—Notice
    - 7—Informational
  - ccmMediaResourceType OID is 1.3.6.1.4.1.9.9.156.1.10.6. The values are:
    - 1—Unknown
    - 2—Media Termination Point
    - 3—Transcoder
    - 4—Conference Bridge

## 5—Music On Hold

- 1.3.6.1.4.1.9.9.156.2.0.6 ccmRouteListExhausted
- 1.3.6.1.4.1.9.9.156.2.0.7 ccmGatewayLayer2Change
- 1.3.6.1.4.1.9.9.156.2.0.8 ccmMaliciousCall
- 1.3.6.1.4.1.9.9.156.2.0.9 ccmQualityReport
- 1.3.6.1.4.1.9.9.156.2.0.10 ccmTLSConnectionFailure

## Dynamic Table Objects

The following table lists the objects that are populated only if the Cisco Unified Communications Manager service is up and running or the local Cisco Unified Communications Manager service in the case of a Cisco Unified Communications Manager cluster configuration.

**Table 3: CISCO-CCM-MIB Dynamic Tables**

Object	Content
ccmTable	This table stores the version and installation ID for the local Cisco Unified CM. The table also stores information about all the Cisco Unified CMs in a cluster that the local Cisco Unified CM knows about but shows “unknown” for the version detail. If the local Cisco Unified CM is down, the table remains empty, except for the version and installation ID values.
ccmPhoneFailed, ccmPhoneStatusUpdate, ccmPhoneExtn, ccmPhone, ccmPhoneExtension	For the Cisco Unified IP Phone, the number of registered phones in ccmPhoneTable should match Cisco Unified Communications Manager/RegisteredHardware Phones perfmon counter. The ccmPhoneTable includes one entry for each registered, unregistered, or rejected Cisco Unified IP Phone. The ccmPhoneExtnTable uses a combined index, ccmPhoneIndex and ccmPhoneExtnIndex, for relating the entries in the ccmPhoneTable and ccmPhoneExtnTable.
ccmCTIDevice, ccmCTIDeviceDirNum	The ccmCTIDeviceTable stores each CTI device as one device. Based on the registration status of the CTI Route Point or CTI Port, the ccmRegisteredCTIDevices, ccmUnregisteredCTIDevices, and ccmRejectedCTIDevices counters in the Cisco Unified Communications Manager MIB get updated.
ccmSIPDevice	The CCMSIPDeviceTable stores each SIP trunk as one device.

Object	Content
ccmH323Device	<p>The ccmH323DeviceTable contains the list of H323 devices for which Cisco Unified Communications Manager contains information (or the local Cisco Unified Communications Manager in the case of a cluster configuration). For H.323 phones or H.323 gateways, the ccmH.323DeviceTable contains one entry for each H.323 device. (The H.323 phone and gateway do not register with Cisco Unified Communications Manager. Cisco Unified Communications Manager generates the H.323Started alarm when it is ready to handle calls for the indicated H.323 phone and gateway.) The system provides the gatekeeper information as part of the H323 trunk information.</p>
ccmVoiceMailDevice, ccmVoiceMailDirNum	<p>For Cisco uOne, ActiveVoice, the ccmVoiceMailDeviceTable includes one entry for each voice-messaging device. Based on the registration status, the ccmRegisteredVoiceMailDevices, ccmUnregisteredVoiceMailDevices, and ccmRejectedVoiceMailDevices counters in the Cisco Unified Communications Manager MIB get updated.</p>
ccmGateway	<p>The ccmRegisteredGateways, ccmUnregistered gateways, and ccmRejectedGateways keep track of the number of registered gateway devices or ports, number of unregistered gateway devices or ports, and number of rejected gateway devices or ports, respectively.</p> <p>Cisco Unified Communications Manager generates alarms at the device or port level. The ccmGatewayTable, based on Cisco Unified CM alarms, contains device- or port-level information. Each registered, unregistered, or rejected device or port has one entry in ccmGatewayTable. The VG200 with two FXS ports and one T1 port has three entries in ccmGatewayTable. The ccmActiveGateway and ccmInActiveGateway counters track number of active (registered) and lost contact with (unregistered or rejected) gateway devices or ports.</p> <p>Based on the registration status, ccmRegisteredGateways, ccmUnregisteredGateways, and ccmRejectedGateways counters get updated.</p>
ccmMediaDeviceInfo	<p>The table contains a list of all media devices that have tried to register with the local Cisco Unified CM at least once.</p>

Object	Content
ccmGroup	This tables contains the Cisco Unified CM groups in a Cisco Unified Communications Manager cluster.
ccmGroupMapping	This table maps all Cisco Unified CMs in a cluster to a Cisco Unified CM group. The table remains empty when the local Cisco Unified CM node is down

## Static Table Objects

The following table lists the objects that get populated when the Cisco Unified Communications Manager SNMP Service is running.

**Table 4: CISCO-CCM-MIB Static Tables**

Object	Content
ccmProductType	The table contains the list of product types that are supported with Cisco Unified Communications Manager (or cluster, in the case of a Cisco Unified Communications Manager cluster configuration), including phone types, gateway types, media device types, H323 device types, CTI device types, voice-messaging device types, and SIP device types.
ccmRegion, ccmRegionPair	ccmRegionTable contains the list of all geographically separated regions in a Cisco Communications Network (CCN) system. The ccmRegionPairTable contains the list of geographical region pairs for a Cisco Unified Communications Manager cluster. Geographical region pairs are defined by Source region and Destination region.
ccmTimeZone	The table contains the list of all time zone groups in a Cisco Unified Communications Manager cluster.
ccmDevicePool	The tables contains the list of all device pools in a Cisco Unified Communications Manager cluster. Device pools are defined by Region, Date/Time Group, and Cisco Unified CM Group.

## Troubleshoot SNMP

### General Tips

The following are general troubleshooting tips:

- Check the community string or snmp user is properly configured on the system using the SNMP configuration web pages

- Check if Cisco Unified CM SNMP Service is activated and running by checking the ccmService window and clicking **Tools > Service Activation/ ControlCenter - Feature Services**.
- Check if SNMP Master Agent is running by checking the ccmService window and clicking **Tools > Service Activation/ ControlCenter - Network Services**
- Check if Cisco Unified CM is running.
- If Cisco Unified CM is not running, only the following MIB tables respond:
  - ccmGroupTable
  - ccmRegionTable
  - ccmRegionPairTable
  - ccmDevicePoolTable
  - ccmProductTypeTable
  - ccmQualityReportAlarmConfigInfo
  - ccmGlobalInfo
- For the rest of the tables to respond Cisco Unified CM needs to be running.
- Set the debug trace level to detailed for Cisco CallManager SNMP Service. Go to the Serviceability web window and click **Trace > Configuration > <select serverCisco> Performance and Monitoring Services > CallManager SNMP Service**.
- Execute the CLI command: **utils snmp walk 2c <community> <ipaddress> 1.3.6.1.4.1.9.9.156** or execute the walk from any other management application on this OID.
- Get the Cisco Unified Communication Manager release details, Cisco SNMP CallManager Service trace, and SNMP Master agent traces after the testing above for troubleshooting reference.

Review this section for Cisco CallManager SNMP Service Troubleshooting tips:

- Be sure to set the trace setting to detailed for Cisco CallManager SNMP Service (see the “SNMP Trace Configuration” chapter of the Cisco Unified Serviceability Administration Guide).
- Execute the command: **snmp walk -c<community> -v2c<ipaddress> 1.3.6.1.4.1.9.9.156.1.1.2**
- Get the Cisco Unified Communications Manager version details
- Collect the following logs and information:
  - SNMP Master Agent (path: platform/snmp/snmpdm/\*) and Cisco CallManager SNMP Service (path: cm/trace/ccmmib/sdi/\*) by using TLC in RTMT or this CLI command: **file get activelog**
  - SNMP package version by using this CLI command: **show packages active snmp**
  - MMF Spy output for phone by using this CLI command: **show risdb query phone**
- Send the trace logs and MMFSpy data for further analysis

The following table provides procedures for verifying that CISCO-CCM-MIB SNMP traps get sent.

Table 5: How to Check CISCO-CCM-MIB SNMP Traps

Trap	Verification procedure
ccmPhoneStatusUpdate	<ol style="list-style-type: none"> <li>1. Set MaxSeverity=Info in CiscoSyslog-&gt;dogBasic MIB table.</li> <li>2. Set PhoneStatusUpdateAlarmInterv=30 or higher in ccmAlarmConfigInfo MIB table.</li> <li>3. Disconnect a Cisco Unified CM server that your phones point to.</li> <li>4. Phones will unregister.</li> <li>5. Connect the Cisco Unified CM server again.</li> <li>6. Phones will re-register.</li> <li>7. Check that the ccmPhoneStatusUpdate trap is generated.</li> </ol>
ccmPhoneFailed	<ol style="list-style-type: none"> <li>1. Set MaxSeverity=Info in CiscoSyslog-&gt;clogBasic MIB table.</li> <li>2. Set PhoneFailedAlarmInterv=30 or higher in ccmAlarmConfigInfo MIB table.</li> <li>3. Make a phone fail. Delete a phone Cisco Unified CM Administration and register the phone again.</li> <li>4. Check that the ccmPhoneFailed trap is generated.</li> </ol>
MediaResourceListExhausted	<ol style="list-style-type: none"> <li>1. Create a Media Resource Group (MRG) that contains one of the standard Conference Bridge resources (CFB-2).</li> <li>2. Create a Media Resource Group List (MRGL) that contains the MRG just created.</li> <li>3. In the Phone Configuration window (for actual phones), set MRGL as the phone Media Resource Group List.</li> <li>4. Stop the IPVMS, which makes the Conference Bridge resource(CFB-2) stop working.</li> <li>5. If you make conference calls with phones that use the media list, you will see "No Conference Bridge available" in the phone screen.</li> <li>6. Check that a MediaListExhausted Alarm/Alert/Trap is generated</li> </ol>

Trap	Verification procedure
RouteListExhausted	<ol style="list-style-type: none"> <li>1. Create a Route Group (RG) that contains one gateway.</li> <li>2. Create a Route Group List (RGL) that contains the RG that was just created.</li> <li>3. Create a Route Pattern (9.XXXX) that routes a 9XXXX call through the RGL.</li> <li>4. Unregister the gateway.</li> <li>5. Dial 9XXXX on one of the phones.</li> <li>6. Check that a RouteListExhausted Alarm/Alert/Trap is generated.</li> </ol>
MaliciousCallFailed	<ol style="list-style-type: none"> <li>1. Similar to QRT, create a softkey template. In the template, add all available “MaliciousCall” softkey to the phone different status.</li> <li>2. Assign the new softkey template to actual phones; reset the phones.</li> <li>3. Make some calls and select the “MaliciousCall” softkey in the phone screen during or after the call.</li> <li>4. Check that a “MaliciousCallFailed” Alarm/Alert/Trap is generated.</li> </ol>

## Logs and Analytical Information for Linux and Cisco Unified CM Releases 5.x 6.x 7.x

Collect the following logs and information for analysis:

- SNMP Master Agent (Path : /platform/snmp/snmpdm/\*)
- Cisco CallManager SNMP Service (Path : /cm/trace/ccmmib/sdi/\*)
- The files can be collected using TLC ( Real Time Monitoring Tool (RTMT) ) or CLI by using the following command: **file get activelog <path mentioned above>**.
- All the files in /usr/local/Snmpri/conf folder. (This is possible only if ROOT/REMOTE login is available)
- The 'ls -l' listing of the above folder. (This is possible only if ROOT/REMOTE login is available)
- Collect Perfmon logs. Execute the following CLI command: **file get activelog /cm/log/ris/csv/**.
- Details of the set of actions performed that resulted in the issue.
- Ccmservice logs. Execute the following CLI command: **file get activelog /tomcat/logs/ccmservice/log4j/**.

- Collect the SNMP package version. Use the **show packages activesnmp** CLI command.
- Get the MMF Spy output for Phone. Use the **show risdb query phone** CLI command.

## Logs and Analytical Information for Windows and Cisco Unified CM Version 4.x

Collect the following logs for analysis:

- Set the Alarm level from the ccmervice Alarm Configuration window for Cisco Unified CM to Detailed.
- Set the RIS Trace configuration from the ccmervice window to Detailed.
- Do a snmpwalk on the ccm MIB from the network management application or execute command from any linux box by using the **snmpwalk -c <community>-v2c <ipaddress> 1.3.6.1.4.1.9.9.156**.
- Capture the output of the snmpwalk.
- Collect the logs under `C:\Program Files\Cisco\Trace\RIS\CCMSNMP_*.log`.
- Collect the logs under `C:\Program Files\Cisco\Trace\DBL\ DBL_SNMP*.txt`.
- Event logs (both application and system).
- mmfSpy output for 'misc', 'CMnode' tables.
- MMFSpy tool to dump registration status (`C:\Program Files\Cisco\Bin\MMFSpy.exe`, gives different options). Usage: "`mmfSpy -j > OutputFileName`".

CISCO-CCM-MIB only supports a limited amount of configuration information about a device. For more complete configuration information, the AXL interface accessing the data in DB serves the purpose.

The list of MMFs that are created by the Cisco Unified CM Agent are as follows:

- cmnode
- cmgroup
- cmgroupmember
- region
- regionmatrix
- timezone
- devicepool
- phonefailed
- phonestatsupd
- cmproduct
- cmmodel

## Limitations

If multiple OIDs are specified in the SNMP request and if the variables are pointing to empty tables in CISCO-CCM-MIB, then the request will take longer. In case the getbulk/getnext/getmany request has multiple OIDs in its request PDU with the subsequent tables being empty in the CISCO-CCM-MIB, the responses may be NO\_SUCH\_NAME for SNMP v1 version or GENERIC\_ERROR for SNMP v2c or v3 version.

- Reason—This timeout occurs due to the code added to enhance the performance of the CCMAgent and throttle when it gets a large number of queries thus protecting the priority of Cisco Unified CM callprocessing engine.
- Workaround:
  - Use the available scalar variables (1.3.6.1.4.1.9.9.156.1.5) to determine the table size before accessing the table. Or do the get operation on the desired table first and then query the non empty tables.
  - Reduce the number of variables queried in a single request. For example, for empty tables, if Management application has timeout set at 3 sec, then recommendations is to specify no more than 1 OID. For non-empty tables it takes 1 second to retrieve 1 row of data.
  - Increase the response timeout.
  - Reduce the number of retries.
  - Avoid using getbulk SNMP API. Getbulk API gets number of records specified by MaxRepetitions. This means even if the next object goes outside the table or MIB, it gets those objects. So if the CISCO-CCM -MIB has empty tables then it goes to next MIB and so will more time to respond. Use getbulk API when it is known that the table is not empty, and also know the number of records. Under this condition limit the max repetition counts to 5 to get response within 5 sec.
  - Structured SNMP queries to adapt to current limits.
  - Avoid doing a number of getbulks on the PhoneTable in case there are a number of phones registered to the Cisco Unified CM, walking it periodically may not be optimal. In such a scenario whenever there is an update, ccmPhoneStatusUpdateTable will be updated, use this information to decide whether to walk the PhoneTable.

## Frequently Asked Questions

- Q. Not getting any SNMP traps from the Cisco Unified Communication Manager node for the CISCO-CCM-MIB.
- A. For receiving SNMP traps in CISCO-CCM-MIB, you need to ensure that the value of the following MIB OIDs are set to appropriate values: ccmPhoneFailedAlarmInterval (1.3.6.1.4.1.9.9.156.1.9.2) and ccmPhoneStatusUpdateAlarmInterv (1.3.6.1.4.1.9.9.156.1.9.4) are set between 30 and 3600. The default is set to 0.

Execute the following commands from any Linux machine:

- `snmpset -c <Community String> -v 2c <transmitter ip address> 1.3.6.1.4.1.9.9.156.1.9.2.0 i <value>`
- `snmpset -c<Community String> -v 2c <transmitter ip address> 1.3.6.1.4.1.9.9.156.1.9.4.0 i <value>`

These are related to registration/deregistration/failure of phones.

You need to ensure that notification destinations are configured. This can be done from the Serviceability Web window. There is a menu for **SNMP > Notification destination**.

Before you configure notification destination, verify that the required SNMP services are activated and running (SNMP Master Agent and Cisco CallManager SNMP Services). Also, make sure that you configured the privileges for the community string/user correctly which should contain Notify permissions as well.

If still Traps are not generated check if corresponding alarms are generated. Since these traps are generated based on the alarm events, ensure that SNMP agents are getting these alarm events. Enable 'Local Syslog', setup the Cisco Unified CM Alarm configuration to 'Informational' level for 'Local Syslog' destination from the Alarm configuration available on **Cisco Unified CM Serviceability web page->Alarm->Configuration**. Then repro the traps and see if corresponding alarms are logged in CiscoSyslog file.

Receiving syslog messages as traps—To receive syslog messages above a particular severity as traps, set the following 2 MIB objects in the clogBasic table:

- `clogNotificationsEnabled` (1.3.6.1.4.1.9.9.41.1.1.2)—Set this to true(1) to enable syslog trap notification. Default value is false (2). For example, `snmpset -c <Community String> -v 2c <transmitter ip address> 1.3.6.1.4.1.9.9.41.1.1.2.0 i <value>`.
- `clogMaxSeverity` (1.3.6.1.4.1.9.9.41.1.1.3)—Set the severity level above which traps are desired. Default value is warning (5). All syslog messages with alarm severity lesser than or equal to configured severity level will be sent as traps if notification is enabled. For example, `snmpset -c <Community String> -v 2c <transmitter ip address> 1.3.6.1.4.1.9.9.41.1.1.3.0 i <value>`.

**Q.** What is the additional .1 and .2 in SNMP walk output?

**A.** The additional .1 and .2 are generic numbers that show the order in an output.

For example, while performing SNMP for OID 1.3.6.1.4.1.9.9.156.1.1.1.1.2(ccmGroupName); the following output will be shown for Cisco Unified Communications Manager, which has two cm group name “Default” and “Cluster1”.

```
iso.3.6.1.4.1.9.9.156.1.1.1.1.2.1 = STRING: "Default"
iso.3.6.1.4.1.9.9.156.1.1.1.1.2.2 = STRING: "Cluster1"
```

**Q.** What are the different traps defined for Cisco Unified Communication Manager?

**A.** The CISCO-CCM-MIB contains the traps related information. Following are the list of defined traps defined:

- `ccmCallManagerFailed`—Indication that the CallManager process detects a failure in one of its critical subsystems. It can also be detected from a heartbeat/event monitoring process.
- `ccmPhoneFailed`—Notification that the intervals specified in `ccmPhoneFailedAlarmInterval` indicate at least one entry in the `ccmPhoneFailedTable`.
- `ccmPhoneStatusUpdate`—Notification that is generated in the intervals specified in `ccmPhoneStatusUpdateInterv` if there is at least one entry in the `ccmPhoneStatusUpdateTable`.
- `ccmGatewayFailed`—Indication that at least one gateway has attempted to register or communicate with the CallManager and failed.




---

**Note** `ccmGatewayFailed` is deprecated and replaced by `ccmGatewayFailedReason`.

---

- `ccmMediaResourceListExhausted`—Indication that the CallManager has run out a certain specified type of resource
- `ccmRouteListExhausted`—Indication that the CallManager could not find an available route in the indicated route list.
- `ccmGatewayLayer2Change`—Sent when the D-Channel/Layer 2 of an interface in a skinny gateway that has registered with the CallManager changes state.
- `ccmMaliciousCall`—Sent when a user registers a call as malicious with the local Cisco Unified CM
- `ccmQualityReport`—Sent when a user reports a quality problem using the Quality Report Tool
- `ccmTLSConnectionFailure`—Sent when CallManager fails to open TLS connection for the indicated device

The mapping of the traps to alarms is as follows:

- `ccmCallManagerFailed`—`CallManagerFailure`
- `ccmPhoneFailed`—`DeviceTransientConnection`
- `ccmPhoneStatusUpdate`
- `ccmGatewayFailed`—`DeviceTransientConnection`
- `ccmMaliciousCall`—`MaliciousCall`
- `ccmMediaResourceListExhausted`—`MediaResourceListExhausted`
- `ccmQualityReportRequest`—`QRTRequest`
- `ccmRouteListExhausted`—`RouteListExhausted`
- `ccmGatewayLayer2Change`—`DChannelOOS`, `DChannelISV`

**Q.** How can different SNMP traps from Cisco Unified Communication Manager be checked?

**A.** Following is the procedure for triggering few traps:

- `ccmPhoneStatusUpdate` trap
  - Set `ccmPhoneStatusUpdateAlarmInterv` (1.3.6.1.4.1.9.9.156.1.9.4) to 30 or higher in `ccmAlarmConfigInfo` MIB table.
  - Disconnect a ccm server that your phones are pointing to.
  - Phones will unregister.
  - Connect the ccm server again.
  - Phones will re-register.
  - Will get the `ccmPhoneStatusUpdate` trap.
- `ccmPhoneFailed` trap
  - Set `ccmPhoneFailedAlarmInterval` (1.3.6.1.4.1.9.9.156.1.9.2) to 30 or higher in `ccmAlarmConfigInfo` MIB table.
  - Make a phone fail. Delete a phone from CM and register the phone again.

- For phone failed traps two different scenarios can be tried:

Set the phone to point to tftp/ccm server A. plugin the phone to ccm server B on different switch. The phone status is unknown. Will see following: 2007-10-31:2007-10-31 14:53:40 Local7.Debug 172.19.240.221 community=public, enterprise=1.3.6.1.4.1.9.9.156.2.0.2, enterprise\_mib\_name=ccmPhoneFailed, uptime=7988879, agent\_ip=128.107.143.68, version=Ver2, ccmAlarmSeverity=error, ccmPhoneFailures=1.

Register a 7960 phone as 7940 phone in the Cisco Unified CM and thus cause the db issue that makes the phone fail trap.

- MediaResourceListExhausted trap
  - Create a Media Resource Group (MRG), have it contains one of the standard ConferenceBridge resource (CFB-2).
  - Create a Media Resource Group List (MRGL), have it contains the MRG just created.
  - In the Phone Configuration page for real phones, set MRGL as the phone Media Resource Group List.
  - Stop the IPVMS which make the ConferenceBridge resource (CFB-2) stop working.
  - Make conference calls with phones that using the media list, you will see “No Conference Bridge available” in the phone screen.
  - Then check if a “MediaListExhausted” Alarm/Alert/Trap is generated.
- RouteListExhausted trap
  - Create a Route Group (RG), have it contains one Gateway.
  - Create a Route Group List (RGL), have it contains the RG just created.
  - Create a Route Pattern (9.XXXX) that reroute a 9XXXX call through the RGL.
  - Unregister the gateway.
  - Dial 9XXXX in one of the phone.
  - Then check if a “RouteListExhausted” Alarm/Alert/Trap is generated.
- MaliciousCallFailed trap
  - Similar as QRT, create a softkey template. In the template, add all available “MaliciousCall” softkey to the phone's different status.
  - Assign the new softkey template to real phones, reset the phones.
  - Making calls, select the “MaliciousCall” in the phone screen during or after the call.
  - Then check if a “MaliciousCallFailed” Alarm/Alert/Trap is generated
- GatewayFailed trap (Method 1)
  - Remove the configuration of the gateway from the database through Web Admin (or) Change the MAC address of the gateway to some invalid value and update.
  - Reboot the gateway

- Another way is to restart the Cisco Unified CM service to which the gateway is connected.
- GatewayFailed trap (Method 2)
  - Set GatewayAlarmEnable=true in ccmAlarmConfigInfo mib table
  - In ccm serviceability->Snmp configuration page, make sure you have SNMP community string and trap destination set correctly.
  - Create a gateway failure event and the trap will be seen on the trap receiver.
  - To cause a gateway fail, Restart Cisco Unified CM service which will cause gateway failover to the redundant ccm manager server. On that server, the gateway should not be configured in the database.
- ccmGatewayLayer2Change trap
  - ccmGatewayLayer2Change trap is triggered during DChannelOOS(D Channel Out of service) or DChannelISV (D Channel Inservice) from Cisco Unified CM. Please check if any such events can be triggered to test it out
- ccmCallManagerFailed trap
  - The CallManager Failed Alarm is generated when an internal error is encountered. These include an internal thread dying due to lack of CPU, timer issues and a couple others. This trap would be something that is hard to reproduce unless the CallManager team give a friendly that intentionally causes one of these occurrences.

- Q.** If the Cisco Unified CM Agent consumes high CPU continuously, what needs to be done?
- A.** Collect the logs as mentioned above (under Troubleshooting) for analysis and refer to defect CSCsm74316 to check if it is being hit. Verify if the fix for the defect has gone into the Cisco Unified CM version used by the customer.
- Q.** If the CTI Routepoint is deleted from Cisco Unified CM Admin UI, an entry exists for that in ccmCTIDeviceTable mib. Why?
- A.** There is service parameter called “RIS Unused Cisco CallManager Device Store Period” which defines how long Unregistered devices (when a registered device is removed from db, it unregisters) will remain in RISDB and hence in the MIB. The ccmadmin page and the SNMP MIB WALK may or may not be in sync, since the ccmadmin page shows the info from the database however SNMP uses the RISDB.
- Q.** When ccmPhoneType is queried from ccmPhoneTable in Cisco-CCM-MIB, no information is returned. Why?
- A.** The ccmPhoneType has been made obsolete. The same information can be retrieved from ccmPhoneProductTypeIndex against CcmProductTypeEntry. In the table, the indexes correspond to the index and name as listed in that table.

Some of other obsolete and alternate OIDs to be referred:

- ccmGatewayType is obsolete and need to refer ccmGateWayProductTypeIndex.
- ccmMediaDeviceType is obsolete and need to refer to ccmMediaDeviceProductTypeIndex
- ccmCTIDeviceType is obsolete and need to refer to ccmCTIDeviceProductTypeIndex

- Q.** A query on `ccmPhoneProductTypeIndex` returns zero. Why?
- A.** Verify that the Cisco Unified CM release that you are using has this capability.
- Q.** While performing a WALK on `ccmPhoneTable`, `ccmPhoneUserName` is not returning any value. How are usernames associated to the IP Phones?
- A.** Create an end user and then go to the phone that has been registered and associate the Owner User ID. Once this is done, the user will be shown by the OID in the SNMP Walk.
- Q.** How do I get the firmware versions of each phone by using SNMP?
- A.** `ccmPhoneLoadID` object in the `ccmPhoneTable` will give the firmware version of each phone. But this value may differ if new image download failed. In case of 7.x versions SNMP will expose both configured firmware ID (`ccmPhoneLoadID`) and the actual running firmware (`ccmPhoneActiveLoad`).
- Q.** CCM MIB returns `ccmVersion` as 5.0.1, which is the incorrect.
- A.** Verify the Cisco Unified CM release that you are using has this capability. If it does not, upgrade.
- Q.** CCM MIB returns incorrect `ccmPhoneLoadID`
- A.** `ccmPhoneLoadID` values are picked up from RISDB which is populated based on the alarm received during Phone registration. Perform the following steps and collect the logs for further analysis:
1. Go to **Serviceability web page > Alarm > Configuration > Service Group (CM Services) > Service (Cisco CallManager)**.
  2. Check Local Syslog, SDI Trace, SDL Trace. Ensure the Alarm Event Level for these selected destinations is set to Informational.
  3. Set the Cisco CallManager trace level to Detailed.
  4. Reset the phones showing incorrect LoadID.
  5. Collect the Syslog and Cisco CallManager traces.
  6. Collect the phone details.
- Q.** How Cisco Unified CM status (START/STOP) monitored?
- A.** For service monitoring we have following options:
- SYSAPPL MIB
  - HOST-RESOURCE-MIB
  - CISCO-CCM-MIB (`ccmStatus`)
  - SOAP interface
  - Real-TimeMonitoringTool (RTMT) alerts

There is a `ccmCallManagerFailed` trap for Cisco Unified CM service failure. But this does not cover normal service stop and unknown crashes.

- Q. The device pool information seems incorrect for any device polled for. The OID used is `ccmPhoneDevicePoolIndex`.
- A. As stated in the CISCO-CCM-CAPABILITY MIB, `ccmPhoneDevicePoolIndex` is not supported, hence it returns 0. The CallManager device registration alarm currently does not contain the devicepool information.

## CISCO-CCM-CAPABILITY



---

**Note** This is a reformatted version of CISCO-CCM-CAPABILITY. Download and compile all of the MIBs in this section from <http://tools.cisco.com/Support/SNMP/do/BrowseMIB.do?local=en&step=2>.

---



---

**Note** This MIB is not meant to perform SNMP queries like MIB walk as there is no agent supporting this MIB. It is only used as documentation supplement to the CISCO-CCM-MIB.

---

Before you can compile CISCO-CCM-CAPABILITY, you need to compile the MIBs listed below in the order listed.

1. SNMPv2-SMI
2. SNMPv2-TC
3. SNMPv2-CONF
4. SNMPv2-MIB
5. IANAifType-MIB
6. IF-MIB
7. CISCO-SMI
8. SNMP-FRAMEWORK-MIB
9. RMON-MIB
10. CISCO-TC
11. CISCO-VTP-MIB
12. RFC1155-SMI
13. RFC-1212
14. SNMPv2-TC-v1
15. CISCO-CDP-MIB
16. CISCO-CCM-CAPABILITY

Additional downloads are:

- OID File: CISCO-CCM-CAPABILITY.OID

## CISCO-CCM-CAPABILITY Revisions

The following table lists the revisions to this MIB beginning with the latest revision first.

**Table 6: History of Revisions**

Date	Action	Description
10-03-2003	Added	Agent capability for CISCO-CCM-MIB
10-03-2003	Added	Agent capabilities for Cisco Call Manager 4.0 release
03-21-2002	Added	DESCRIPTION Added the agent capabilities for Cisco Call Manager 3.3 release.
07-02-2001	Added	DESCRIPTION Added the agent capabilities for Cisco Call Manager 3.0 release.
06-19-2001	Initial Version	::= { ciscoAgentCapability 211 }

## CISCO-CCM-CAPABILITY Definitions

The following definitions are imported for CISCO-CCM-CAPABILITY:

- MODULE-IDENTITY
- From SNMPv2-SMI—AGENT-CAPABILITIES
- From SNMPv2-CONF—ciscoAgentCapability
- From CISCO-SMI—ciscoCCMCapability MODULE-IDENTITY

## CISCO-CCM-CAPABILITY Agent Capabilities

### **ciscoCCMCapabilityV3R00 AGENT-CAPABILITIES**

PRODUCT RELEASE Cisco Call Manager 3.0

STATUS Current

DESCRIPTION Cisco Call Manager Agent Capabilities

SUPPORTS Cisco-ccm-mib

INCLUDES { ccmInfoGroup, ccmPhoneInfoGroup, ccmGatewayInfoGroup }

VARIATION ccmPhoneE911Location

ACCESS not-implemented

DESCRIPTION ccmPhoneE911Location is not supported  
 VARIATION ccmPhoneLastError  
 ACCESS not-implemented  
 DESCRIPTION ccmPhoneLastError is not supported  
 VARIATION ccmPhoneTimeLastError  
 ACCESS not-implemented  
 DESCRIPTION ccmPhoneTimeLastError is not supported  
 VARIATION ccmPhoneDevicePoolIndex  
 ACCESS not-implemented  
 DESCRIPTION ccmPhoneDevicePoolIndex is not supported  
 VARIATION ccmGatewayDevicePoolIndex  
 ACCESS not-implemented  
 DESCRIPTION ccmGatewayDevicePoolIndex is not supported  
 VARIATION ccmGatewayTrunkIndex  
 ACCESS not-implemented  
 DESCRIPTION ccmGatewayTrunkIndex is not supported  
 VARIATION ccmGatewayTrunkType  
 ACCESS not-implemented  
 DESCRIPTION ccmGatewayTrunkType is not supported  
 VARIATION ccmGatewayTrunkName  
 ACCESS not-implemented  
 DESCRIPTION ccmGatewayTrunkName is not supported  
 VARIATION ccmTrunkGatewayIndex  
 ACCESS not-implemented  
 DESCRIPTION ccmTrunkGatewayIndex is not supported  
 VARIATION ccmGatewayTrunkStatus  
 ACCESS not-implemented  
 DESCRIPTION ccmGatewayTrunkStatus is not supported  
 ::= { ciscoCCMCapability 1 }

### **ciscoCCMCapabilityV3R01 AGENT-CAPABILITIES**

PRODUCT-RELEASE Cisco Call Manager 3.1

STATUS current  
 DESCRIPTION Cisco Call Manager Agent capabilities  
 SUPPORTS CISCO-CCM-MIB  
 INCLUDES { ccmInfoGroupRev1, ccmPhoneInfoGroupRev1, ccmGatewayInfoGroupRev1,  
 ccmMediaDeviceInfoGroup, ccmGatekeeperInfoGroup, ccmCTIDeviceInfoGroup,  
 ccmNotificationsInfoGroup, ccmNotificationsGroup }  
 VARIATION ccmPhoneE911Location  
 ACCESS not-implemented  
 DESCRIPTION ccmPhoneE911Location is not supported  
 VARIATION ccmPhoneLastError  
 ACCESS not-implemented  
 DESCRIPTION ccmPhoneLastError is not supported  
 VARIATION ccmPhoneTimeLastError  
 ACCESS not-implemented

DESCRIPTION ccmPhoneTimeLastError is not supported  
 VARIATION ccmPhoneDevicePoolIndex  
 ACCESS not-implemented  
 DESCRIPTION ccmPhoneDevicePoolIndex is not supported  
 VARIATION ccmGatewayDevicePoolIndex  
 VARIATION ccmGatewayDevicePoolIndex  
 ACCESS not-implemented  
 DESCRIPTION ccmGatewayDevicePoolIndex is not supported  
 VARIATION ccmMediaDeviceDevicePoolIndex  
 ACCESS not-implemented  
 DESCRIPTION ccmMediaDeviceDevicePoolIndex is not supported  
 VARIATION ccmGatekeeperDevicePoolIndex  
 ACCESS not-implemented  
 DESCRIPTION ccmGatekeeperDevicePoolIndex is not supported  
 VARIATION ccmCTIDevicePoolIndex  
 ACCESS not-implemented  
 DESCRIPTION ccmCTIDevicePoolIndex is not supported  
 VARIATION ccmCTIDeviceAppInfo  
 ACCESS not-implemented  
 DESCRIPTION ccmCTIDeviceAppInfo is not supported  
 VARIATION ccmPhonePhysicalAddress  
 SYNTAX MacAddress  
 DESCRIPTION Represents the MAC address of the phone  
 ::= { ciscoCCMCapability 2 }

### **ciscoCCMCapabilityV3R03 AGENT-CAPABILITIES**

PRODUCT-RELEASE Cisco Call Manager 3.3

STATUS obsolete and superseded by ciscoCCMCapabilityV3R03Rev1  
 DESCRIPTION Cisco Call Manager Agent capabilities  
 SUPPORTS CISCO-CCM-MIB  
 INCLUDES { ccmInfoGroupRev2, ccmPhoneInfoGroupRev2, ccmGatewayInfoGroupRev2,  
 ccmMediaDeviceInfoGroupRev1, ccmCTIDeviceInfoGroupRev1, ccmNotificationsInfoGroupRev1,  
 ccmNotificationsGroup, ccmH323DeviceInfoGroup, ccmVoiceMailDeviceInfoGroup }  
 VARIATION ccmPhoneE911Location  
 ACCESS not-implemented  
 DESCRIPTION ccmPhoneE911Location is not supported  
 VARIATION ccmPhoneDevicePoolIndex  
 ACCESS not-implemented  
 DESCRIPTION ccmPhoneDevicePoolIndex is not supported  
 VARIATION ccmGatewayDevicePoolIndex  
 ACCESS not-implemented  
 DESCRIPTION ccmGatewayDevicePoolIndex is not supported  
 VARIATION ccmMediaDeviceDevicePoolIndex  
 ACCESS not-implemented  
 DESCRIPTION ccmMediaDeviceDevicePoolIndex is not supported  
 VARIATION ccmCTIDevicePoolIndex

ACCESS not-implemented

DESCRIPTION ccmCTIDevicePoolIndex is not supported

VARIATION ccmPhoneFailedTable

DESCRIPTION The table containing the list of all phones which attempted to register with the local call manager and failed. The entries which have not been updated and kept at least for the duration specified in the ccmPhoneFailedStorePeriod will be deleted. Reasons for these failures could be due to configuration error, maximum number of phones has been reached, lost contact, etc.

VARIATION ccmPhoneStatusUpdateTableStateId

DESCRIPTION The current state of ccmPhoneStatusUpdateTable. The initial value of this object is 0 and it will be incremented everytime when there is a change (addition/deletion/modification) to the ccmPhoneStatusUpdateTable. This value and sysUpTime should be used together to find if the table has changed or not. When the SNMP service is restarted this value will be reset to 0.

VARIATION ccmPhStatUpdtTblLastAddedIndex

SYNTAX CcmIndexOrZero

DESCRIPTION The ccmPhoneStatusUpdateIndex value of the last entry that was added to the ccmPhoneStatusUpdateTable. This value together with sysUpTime can be used by the manager applications to identify the new entries in the ccmPhoneStatusUpdateTable since their last poll. This value need not be the same as the highest index in the ccmPhoneStatusUpdateTable as the index could have wrapped around. The initial value of this object is 0 which indicates that there has been no entries added to this table. When the SNMP service is restarted this value will be reset to 0.

VARIATION ccmPhFailedTblLastAddedIndex

SYNTAX CcmIndexOrZero

DESCRIPTION The ccmPhoneFailedIndex value of the last entry that was added to the ccmPhoneFailedTable. This value together with sysUpTime can be used by the manager applications to identify the new entries in the ccmPhoneFailedTable since their last poll. This value need not be the same as the highest index in the ccmPhoneFailedTable as the index could have wrapped around. The initial value of this object is 0 which indicates that there has been no entries added to this table. When the SNMP service is restarted this value will be reset to 0.

VARIATION ccmPhoneFailedStorePeriod

DESCRIPTION The time duration for storing each entry in the ccmPhoneFailedTable. The entries which have not been updated and kept at least this period will be deleted. This value should ideally be set to a higher value than the ccmPhoneFailedAlarmInterval object. The default value is 1800 seconds.

::= { ciscoCCMCapability 3 }

### **ciscoCCMCapabilityV3R03Rev1 AGENT-CAPABILITIES**

PRODUCT-RELEASE Cisco Call Manager 3.3

STATUS current

DESCRIPTION Cisco Call Manager Agent capabilities

SUPPORTS CISCO-CCM-MIB

INCLUDES { ccmInfoGroupRev2, ccmPhoneInfoGroupRev2, ccmGatewayInfoGroupRev2, ccmMediaDeviceInfoGroupRev1, ccmCTIDeviceInfoGroupRev1, ccmNotificationsInfoGroupRev1, ccmNotificationsGroup, ccmH323DeviceInfoGroup, ccmVoiceMailDeviceInfoGroup }

VARIATION ccmPhoneE911Location

ACCESS not-implemented

DESCRIPTION ccmPhoneE911Location is not supported

VARIATION ccmPhoneDevicePoolIndex

ACCESS not-implemented

DESCRIPTION ccmPhoneDevicePoolIndex is not supported  
 VARIATION ccmGatewayDevicePoolIndex  
 ACCESS not-implemented  
 DESCRIPTION ccmGatewayDevicePoolIndex is not supported  
 VARIATION ccmMediaDeviceDevicePoolIndex  
 ACCESS not-implemented  
 DESCRIPTION ccmMediaDeviceDevicePoolIndex is not supported  
 VARIATION ccmCTIDevicePoolIndex  
 ACCESS not-implemented  
 DESCRIPTION ccmCTIDevicePoolIndex is not supported  
 ::= { ciscoCCMCapability 4 }

### ciscoCCMCapabilityV4R00 AGENT-CAPABILITIES

PRODUCT-RELEASE Cisco Call Manager 4.0

STATUS current  
 DESCRIPTION Cisco Call Manager Agent capabilities  
 SUPPORTS CISCO-CCM-MIB  
 INCLUDES { ccmInfoGroupRev3, ccmPhoneInfoGroupRev3, ccmGatewayInfoGroupRev3,  
 ccmMediaDeviceInfoGroupRev2, ccmCTIDeviceInfoGroupRev2, ccmNotificationsInfoGroupRev2,  
 ccmNotificationsGroupRev1, ccmH323DeviceInfoGroupRev1, ccmVoiceMailDeviceInfoGroupRev1,  
 ccmSIPDeviceInfoGroup }  
 VARIATION ccmPhoneE911Location  
 ACCESS not-implemented  
 DESCRIPTION ccmPhoneE911Location is not supported  
 VARIATION ccmPhoneDevicePoolIndex  
 ACCESS not-implemented  
 DESCRIPTION ccmPhoneDevicePoolIndex is not supported  
 VARIATION ccmGatewayDevicePoolIndex  
 ACCESS not-implemented  
 DESCRIPTION ccmGatewayDevicePoolIndex is not supported  
 VARIATION ccmMediaDeviceDevicePoolIndex  
 ACCESS not-implemented  
 DESCRIPTION ccmMediaDeviceDevicePoolIndex is not supported  
 VARIATION ccmCTIDevicePoolIndex  
 ACCESS not-implemented  
 DESCRIPTION ccmCTIDevicePoolIndex is not supported  
 ::= { ciscoCCMCapability 5 }

## CISCO-CDP-MIB



### Note

This is a reformatted version of CISCO-CDP-MIB. Download and compile all of the MIBs in this section from <http://tools.cisco.com/Support/SNMP/do/BrowseMIB.do?local=en&step=2>.

This MIB is for the management of the Cisco Discovery Protocol (CDP) in Cisco devices. Before you can compile CISCO-CDP-MIB, you need to compile the MIBs listed below in the order listed.

1. SNMPv2-SMI
2. SNMPv2-TC
3. SNMPv2-CONF
4. SNMPv2-MIB
5. IANAifType-MIB
6. IF-MIB
7. CISCO-SMI
8. SNMP-FRAMEWORK-MIB
9. RMON-MIB
10. CISCO-TC
11. CISCO-VTP-MIB
12. RFC1155-SMI
13. RFC-1212
14. SNMPv2-TC-v1
15. CISCO-CDP-MIB

Additional downloads are:

- OID File: CISCO-CDP-MIB.oid
- Capability File: CISCO-CDP-CAPABILITY

## CISCO-CDP-MIB Revisions

The following table lists the revision to this MIB beginning with the latest revision.

**Table 7: History of Revisions**

Date	Action	Description
11-23-2001	Added	cdpInterfaceExtTable which contains the following objects: cdpInterfaceExtendedTrust, cdpInterfaceCosForUntrustedPort
04-23-2001	Added	cdpGlobalDeviceIdFormatCpb, cdpGlobalDeviceIdFormatCpb, cdpGlobalDeviceIdFormat

Date	Action	Description
11-22-2000	Added	cdpCacheApplianceID, cdpCacheVlanID, cdpCachePowerConsumption, cdpCacheMTU, cdpCachePrimaryMgmtAddrType cdpCachePrimaryMgmtAddrType, cdpCachePrimaryMgmtAddr, cdpCacheSecondaryMgmtAddrType cdpCacheSecondaryMgmtAddrType, cdpCacheSecondaryMgmtAddr, cdpCacheLastChange, cdpCachePhysLocation, cdpCacheSysName, cdpCacheSysObjectID, cdpGlobalLastChange
12-10-1998	Added	cdpGlobalDeviceId
09-16-1998	Added	These objects to cdpCacheTable: cdpCacheVTPMgmtDomain, cdpCacheNativeVLAN, cdpCacheDuplex
07-08-1996	Obsoleted and defined cdpGlobal	cdpInterfaceMessageInterval
08-15-1995	—	Specified a correct (non-negative) range for several index objects
07-27-1995	—	Corrected range of cdpInterfaceMessageInterval
01-25-1995	Moved from ciscoExperiment to ciscoMgmt OID subtree ::= { ciscoMgmt 23 }	ciscoCdpMIBObjects OBJECT IDENTIFIER ::= { ciscoCdpMIB 1 } cdpInterface OBJECT IDENTIFIER ::= { ciscoCdpMIBObjects 1 } cdpCache OBJECT IDENTIFIER ::= { ciscoCdpMIBObjects 2 } cdpGlobal OBJECT IDENTIFIER ::= { ciscoCdpMIBObjects 3 }

## CISCO-CDP-MIB Definitions

The following definitions are imported for CISCO-CDP-MIB:

- MODULE-IDENTITY, OBJECT-TYPE, Integer32
- From SNMPv2-SMI—MODULE-COMPLIANCE, OBJECT-GROUP
- From SNMPv2-CONF—TruthValue, DisplayString, TimeStamp

- From SNMPv2-TC—ciscoMgmt
- From CISCO-SMI—CiscoNetworkProtocol, CiscoNetworkAddress, Unsigned32
- From CISCO-TC —VlanIndex
- From CISCO-VTP-MIB—ifIndex
- From IF-MIB—ciscoCdpMIB MODULE-IDENTITY

## CDP Interface Group

### **cdpInterfaceTable OBJECT-TYPE**

SYNTAX SEQUENCE OF CdpInterfaceEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

The (conceptual) table containing the status of CDP on the device interfaces.

::= { cdpInterface 1 }

### **cdpInterfaceEntry OBJECT-TYPE**

SYNTAX CdpInterfaceEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

An entry (conceptual row) in the cdpInterfaceTable, containing the status of CDP on an interface.

INDEX { cdpInterfaceIfIndex }

::= { cdpInterfaceTable 1 }

CdpInterfaceEntry ::= SEQUENCE {  
 cdpInterfaceIfIndex Integer32,  
 cdpInterfaceEnableTruthValue,  
 cdpInterfaceMessageInterval INTEGER,  
 cdpInterfaceGroup Integer32,  
 cdpInterfacePort Integer32

}

### **cdpInterfaceIfIndex OBJECT-TYPE**

SYNTAX Integer32 (0..2147483647)

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

The ifIndex value of the local interface. For 802.3 Repeaters on which the repeater ports do not have ifIndex values assigned, this value is a unique value for the port, and greater than any ifIndex value supported by the repeater; in this case, the specific port is indicated by corresponding values of cdpInterfaceGroup and cdpInterfacePort, where these values correspond to the group number and port number values of RFC 1516.

```
::= { cdpInterfaceEntry 1 }
```

#### **cdpInterfaceEnable OBJECT-TYPE**

SYNTAX TruthValue

MAX-ACCESS read-write

STATUS current

DESCRIPTION

An indication of whether the Cisco Discovery Protocol is currently running on this interface. This variable has no effect when CDP is disabled (cdpGlobalRun = FALSE).

```
::= { cdpInterfaceEntry 2 }
```

#### **cdpInterfaceMessageInterval OBJECT-TYPE**

SYNTAX INTEGER (5..254)

UNITS seconds

MAX-ACCESS read-write

STATUS obsolete and replaced by cdpGlobalMessageInterval. This object should be applied to the whole system instead of per interface.

DESCRIPTION

The interval at which CDP messages are to be generated on this interface. The default value is 60 seconds.

```
::= { cdpInterfaceEntry 3 }
```

#### **cdpInterfaceGroup OBJECT-TYPE**

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object is only relevant to interfaces which are repeater ports on 802.3 repeaters. In this situation, it indicates the RFC1516 group number of the repeater port which corresponds to this interface.

```
::= { cdpInterfaceEntry 4 }
```

#### **cdpInterfacePort OBJECT-TYPE**

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object is only relevant to interfaces which are repeater ports on 802.3 repeaters. In this situation, it indicates the RFC1516 port number of the repeater port which corresponds to this interface.

```
::= { cdpInterfaceEntry 5 }
```

#### **cdpInterfaceExtTable OBJECT-TYPE**

SYNTAX SEQUENCE OF CdpInterfaceExtEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

This table contains the additional CDP configuration on the device interfaces.

```
::= { cdpInterface 2 }
```

#### **cdpInterfaceExtEntry OBJECT-TYPE**

SYNTAX CdpInterfaceExtEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

An entry in the cdpInterfaceExtTable contains the values configured for Extended Trust TLV and COS (Class of Service) for Untrusted Ports TLV on an interface which supports the sending of these TLVs.

INDEX { ifIndex }

```
::= { cdpInterfaceExtTable 1 }
```

```
CdpInterfaceExtEntry ::= SEQUENCE {
    cdpInterfaceExtendedTrust INTEGER,
    cdpInterfaceCosForUntrustedPort Unsigned32
}
```

#### **cdpInterfaceExtendedTrust OBJECT-TYPE**

SYNTAX INTEGER { trusted(1), noTrust(2) }

MAX-ACCESS read-write

STATUS current

DESCRIPTION

Indicates the value to be sent by Extended Trust TLV. If trusted(1) is configured, the value of Extended Trust TLV is one byte in length with its least significant bit equal to 1 to indicate extended trust. All other bits are 0. If noTrust(2) is configured, the value of Extended Trust TLV is one byte in length with its least significant bit equal to 0 to indicate no extended trust. All other bits are 0.

```
::= { cdpInterfaceExtEntry 1 }
```

#### **cdpInterfaceCosForUntrustedPort OBJECT-TYPE**

SYNTAX Unsigned32 (0..7)

MAX-ACCESS read-write

STATUS current

DESCRIPTION

Indicates the value to be sent by COS for Untrusted Ports TLV.

::= { cdpInterfaceExtEntry 2 }

## CDP Address Cache Group

### **cdpCacheTable OBJECT-TYPE**

SYNTAX SEQUENCE OF CdpCacheEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

The (conceptual) table containing the cached information obtained via receiving CDP messages.

::= { cdpCache 1 }

### **cdpCacheEntry OBJECT-TYPE**

SYNTAX CdpCacheEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

An entry (conceptual row) in the cdpCacheTable, containing the information received via CDP on one interface from one device. Entries appear when a CDP advertisement is received from a neighbor device. Entries disappear when CDP is disabled on the interface, or globally.

INDEX { cdpCacheIfIndex, cdpCacheDeviceIndex }

::= { cdpCacheTable 1 }

CdpCacheEntry ::= SEQUENCE {

cdpCacheIfIndex Integer32,

cdpCacheDeviceIndex Integer32,

cdpCacheAddressType CiscoNetworkProtocol,

cdpCacheAddressCiscoNetworkAddress,

cdpCacheVersionDisplayString,

cdpCacheDeviceIdDisplayString,

cdpCacheDevicePort DisplayString,

cdpCachePlatformDisplayString,

cdpCacheCapabilitiesOCTET STRING,

cdpCacheVTPMgmtDomain DisplayString,

cdpCacheNativeVLAN VlanIndex,

```

cdpCacheDuplex INTEGER,
cdpCacheApplianceID Unsigned32,
cdpCacheVlanID Unsigned32,
cdpCachePowerConsumptionUnsigned32,
cdpCacheMTUUnsigned32,
cdpCacheSysNameDisplayString,
cdpCacheSysObjectID OBJECT IDENTIFIER,
cdpCachePrimaryMgmtAddrType CiscoNetworkProtocol,
cdpCachePrimaryMgmtAddr CiscoNetworkAddress,
cdpCacheSecondaryMgmtAddrType CiscoNetworkProtocol,
cdpCacheSecondaryMgmtAddr CiscoNetworkAddress,
cdpCachePhysLocationDisplayString,
cdpCacheLastChange TimeStamp
}

```

**cdpCacheIfIndex OBJECT-TYPE**

SYNTAX Integer32 (0..2147483647)

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

Normally, the ifIndex value of the local interface. For 802.3 repeaters for which the repeater ports do not have ifIndex values assigned, this value is a unique value for the port, and greater than any ifIndex value supported by the repeater; the specific port number in this case, is given by the corresponding value of cdpInterfacePort.

::= { cdpCacheEntry 1 }

**cdpCacheDeviceIndex OBJECT-TYPE**

SYNTAX Integer32 (0..2147483647)

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

A unique value for each device from which CDP messages are being received.

::= { cdpCacheEntry 2 }

**cdpCacheAddressType OBJECT-TYPE**

SYNTAX CiscoNetworkProtocol

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

An indication of the type of address contained in the corresponding instance of `cdpCacheAddress`.

::= { `cdpCacheEntry 3` }

**cdpCacheAddress OBJECT-TYPE**

SYNTAX `CiscoNetworkAddress`

MAX-ACCESS `read-only`

STATUS `current`

## DESCRIPTION

The (first) network-layer address of the device's SNMP-agent as reported in the Address TLV of the most recently received CDP message. For example, if the corresponding instance of `cacheAddressType` had the value 'ip(1)', then this object would be an IP-address.

::= { `cdpCacheEntry 4` }

**cdpCacheVersion OBJECT-TYPE**

SYNTAX `DisplayString`

MAX-ACCESS `read-only`

STATUS `current`

DESCRIPTION The Version string as reported in the most recent CDP message. The zero-length string indicates no Version field (TLV) was reported in the most recent CDP message.

::= { `cdpCacheEntry 5` }

**cdpCacheDeviceId OBJECT-TYPE**

SYNTAX `DisplayString`

STATUS `current`

## DESCRIPTION

The Device-ID string as reported in the most recent CDP message. The zero-length string indicates no Device-ID field (TLV) was reported in the most recent CDP message.

MAX-ACCESS `read-only`

::= { `cdpCacheEntry 6` }

**cdpCacheDevicePort OBJECT-TYPE**

SYNTAX `DisplayString`

MAX-ACCESS `read-only`

STATUS `current`

## DESCRIPTION

The Port-ID string as reported in the most recent CDP message. This will typically be the value of the `ifName` object (e.g. `Ethernet0`). The zero-length string indicates no Port-ID field (TLV) was reported in the most recent CDP message.

::= { `cdpCacheEntry 7` }

**cdpCachePlatform OBJECT-TYPE**

SYNTAX DisplayString

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The Device Hardware Platform as reported in the most recent CDP message. The zero-length string indicates that no Platform field (TLV) was reported in the most recent CDP message.

::= { cdpCacheEntry 8 }

**cdpCacheCapabilities OBJECT-TYPE**

SYNTAX OCTET STRING (SIZE (0..4))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The Device Functional Capabilities as reported in the most recent CDP message. For latest set of specific values, see the latest version of the CDP specification. The zero-length string indicates no Capabilities field (TLV) was reported in the most recent CDP message.

REFERENCE Cisco Discovery Protocol Specification, 10/19/94.

::= { cdpCacheEntry 9 }

**cdpCacheVTPMgmtDomain OBJECT-TYPE**

SYNTAX DisplayString (SIZE (0..32))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The VTP Management Domain for the remote device interface, as reported in the most recently received CDP message. This object is not instantiated if no VTP Management Domain field (TLV) was reported in the most recently received CDP message.

REFERENCE managementDomainName in CISCO-VTP-MIB

::= { cdpCacheEntry 10 }

**cdpCacheNativeVLAN OBJECT-TYPE**

SYNTAX VlanIndex

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The remote device interface native VLAN, as reported in the most recent CDP message. The value 0 indicates no native VLAN field (TLV) was reported in the most recent CDP message.

::= { cdpCacheEntry 11 }

**cdpCacheDuplex OBJECT-TYPE**

SYNTAX INTEGER { unknown(1), halfduplex(2), fullduplex(3) }

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The remote device interface duplex mode, as reported in the most recent CDP message. The value unknown(1) indicates no duplex mode field (TLV) was reported in the most recent CDP message.

::= { cdpCacheEntry 12 }

**cdpCacheApplianceID OBJECT-TYPE**

SYNTAX Unsigned32 (0..255)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The remote device Appliance ID, as reported in the most recent CDP message. This object is not instantiated if no Appliance VLAN-ID field (TLV) was reported in the most recently received CDP message.

::= { cdpCacheEntry 13 }

**cdpCacheVlanID OBJECT-TYPE**

SYNTAX Unsigned32 (0..4095)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The remote device VoIP VLAN ID, as reported in the most recent CDP message. This object is not instantiated if no Appliance VLAN-ID field (TLV) was reported in the most recently received CDP message.

::= { cdpCacheEntry 14 }

**cdpCachePowerConsumption OBJECT-TYPE**

SYNTAX Unsigned32

UNITS milliwatts

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The amount of power consumed by remote device, as reported in the most recent CDP message. This object is not instantiated if no Power Consumption field (TLV) was reported in the most recently received CDP message.

::= { cdpCacheEntry 15 }

**cdpCacheMTU OBJECT-TYPE**

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

Indicates the size of the largest datagram that can be sent/received by remote device, as reported in the most recent CDP message. This object is not instantiated if no MTU field (TLV) was reported in the most recently received CDP message.

::= { cdpCacheEntry 16 }

**cdpCacheSysName OBJECT-TYPE**

SYNTAX DisplayString (SIZE (0..255))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

Indicates the value of the remote device sysName MIB object. By convention, it is the device fully qualified domain name. This object is not instantiated if no sysName field (TLV) was reported in the most recently received CDP message.

::= { cdpCacheEntry 17 }

**cdpCacheSysObjectID OBJECT-TYPE**

SYNTAX OBJECT IDENTIFIER

MAX-ACCESS read-only

STATUS current

DESCRIPTION

Indicates the value of the remote device sysObjectID MIB object. This object is not instantiated if no sysObjectID field (TLV) was reported in the most recently received CDP message.

::= { cdpCacheEntry 18 }

**cdpCachePrimaryMgmtAddrType OBJECT-TYPE**

SYNTAX CiscoNetworkProtocol

MAX-ACCESS read-only

STATUS current

DESCRIPTION

An indication of the type of address contained in the corresponding instance of cdpCachePrimaryMgmtAddress.

::= { cdpCacheEntry 19 }

**cdpCachePrimaryMgmtAddr OBJECT-TYPE**

SYNTAX CiscoNetworkAddress

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object indicates the (first) network layer address at which the device will accept SNMP messages as reported in the most recently received CDP message. If the corresponding instance of `cdpCachePrimaryMgmtAddrType` has the value `ip(1)`, then this object would be an IP-address. If the remote device is not currently manageable via any network protocol, this object has the special value of the IPv4 address `0.0.0.0`. If the most recently received CDP message did not contain any primary address at which the device prefers to receive SNMP messages, then this object is not instantiated.

::= { cdpCacheEntry 20 }

#### **cdpCacheSecondaryMgmtAddrType OBJECT-TYPE**

SYNTAX CiscoNetworkProtocol

MAX-ACCESS read-only

STATUS current

DESCRIPTION

An indication of the type of address contained in the corresponding instance of `cdpCacheSecondaryMgmtAddress`.

::= { cdpCacheEntry 21 }

#### **cdpCacheSecondaryMgmtAddr OBJECT-TYPE**

SYNTAX CiscoNetworkAddress

MAX-ACCESS read-only

STATUS current

DESCRIPTION

This object indicates the alternate network layer address (other than the one indicated by `cdpCachePrimaryMgmtAddr`) at which the device will accept SNMP messages as reported in the most recently received CDP message. If the corresponding instance of `cdpCacheSecondaryMgmtAddrType` has the value `ip(1)`, then this object would be an IP-address. If the most recently received CDP message did not contain such an alternate network layer address, then this object is not instantiated.

::= { cdpCacheEntry 22 }

#### **cdpCachePhysLocation OBJECT-TYPE**

SYNTAX DisplayString

MAX-ACCESS read-only

STATUS current

DESCRIPTION

Indicates the physical location, as reported by the most recent CDP message, of a connector which is on, or physically connected to, the remote device's interface over which the CDP packet is sent. This object is not instantiated if no Physical Location field (TLV) was reported by the most recently received CDP message.

::= { cdpCacheEntry 23 }

**cdpCacheLastChange OBJECT-TYPE**

SYNTAX TimeStamp

MAX-ACCESS read-only

STATUS current

DESCRIPTION

Indicates the time when this cache entry was last changed. This object is initialised to the current time when the entry gets created and updated to the current time whenever the value of any (other) object instance in the corresponding row is modified.

::= { cdpCacheEntry 24 }

## CDP Global Group

**cdpGlobalRun OBJECT-TYPE**

SYNTAX TruthValue

MAX-ACCESS read-write

STATUS current

DESCRIPTION

An indication of whether the Cisco Discovery Protocol is currently running. Entries in cdpCacheTable are deleted when CDP is disabled.

DEFVAL { true }

::= { cdpGlobal 1 }

**cdpGlobalMessageInterval OBJECT-TYPE**

SYNTAX INTEGER (5..254)

UNITS seconds

MAX-ACCESS read-write

STATUS current

DESCRIPTION

The interval at which CDP messages are to be generated. The default value is 60 seconds.

DEFVAL { 60 }

::= { cdpGlobal 2 }

**cdpGlobalHoldTime OBJECT-TYPE**

SYNTAX INTEGER (10..255)

UNITS seconds

MAX-ACCESS read-write

STATUS current

DESCRIPTION

The time for the receiving device holds CDP message. The default value is 180 seconds.

```
DEFVAL { 180 }
::= { cdpGlobal 3 }
```

#### **cdpGlobalDeviceId OBJECT-TYPE**

```
SYNTAX DisplayString
MAX-ACCESS read-only
STATUS current
DESCRIPTION
```

The device ID advertised by this device. The format of this device id is characterized by the value of cdpGlobalDeviceIdFormat object.

```
::= { cdpGlobal 4 }
```

#### **cdpGlobalLastChange OBJECT-TYPE**

```
SYNTAX TimeStamp
MAX-ACCESS read-only
STATUS current
DESCRIPTION
```

Indicates the time when the cache table was last changed. It is the most recent time at which any row was last created, modified or deleted.

```
::= { cdpGlobal 5 }
```

#### **cdpGlobalDeviceIdFormatCpb OBJECT-TYPE**

```
SYNTAX BITS { serialNumber(0), macAddress(1), other (2) }
MAX-ACCESS read-only
STATUS current
DESCRIPTION
```

Indicates the Device-Id format capability of the device. The serialNumber(0) indicates that the device supports using serial number as the format for its DeviceId. The macAddress(1) indicates that the device supports using layer 2 MAC address as the format for its DeviceId. The other(2) indicates that the device supports using its platform specific format as the format for its DeviceId.

```
::= { cdpGlobal 6 }
```

#### **cdpGlobalDeviceIdFormat OBJECT-TYPE**

```
SYNTAX INTEGER { serialNumber(1), macAddress(2), other(3) }
MAX-ACCESS read-write
STATUS current
DESCRIPTION
```

An indication of the format of Device-Id contained in the corresponding instance of cdpGlobalDeviceId. User can only specify the formats that the device is capable of as denoted in cdpGlobalDeviceIdFormatCpb

object. The serialNumber(1) indicates that the value of cdpGlobalDeviceId object is in the form of an ASCII string contain the device serial number. The macAddress(2) indicates that the value of cdpGlobalDeviceId object is in the form of Layer 2 MAC address. The other(3) indicates that the value of cdpGlobalDeviceId object is in the form of a platform specific ASCII string contain info that identifies the device. For example: ASCII string contains serialNumber appended/prepended with system name.

```
::= { cdpGlobal 7 }
```

## CDP MIB Conformance Information

```
ciscoCdpMIBConformance OBJECT IDENTIFIER ::= { ciscoCdpMIB 2 }
```

```
ciscoCdpMIBCompliances OBJECT IDENTIFIER ::= { ciscoCdpMIBConformance 1 }
```

```
ciscoCdpMIBGroups OBJECT IDENTIFIER ::= { ciscoCdpMIBConformance 2 }
```

## CDP MIB Compliance Statements

### ciscoCdpMIBCompliance MODULE-COMPLIANCE

STATUS obsoleted and superseded by ciscoCdpMIBComplianceV11R01

DESCRIPTION

The compliance statement for the CDP MIB.

MODULE This module

MANDATORY-GROUPS { ciscoCdpMIBGroup }

```
::= { ciscoCdpMIBCompliances 1 }
```

### ciscoCdpMIBComplianceV11R01 MODULE-COMPLIANCE

STATUS obsoleted and superseded by ciscoCdpMIBComplianceV11R02

DESCRIPTION

The compliance statement for the CDP MIB.

MANDATORY-GROUPS { ciscoCdpMIBGroupV11R01 }

```
::= { ciscoCdpMIBCompliances 2 }
```

### ciscoCdpMIBComplianceV11R02 MODULE-COMPLIANCE

STATUS obsoleted and superseded by ciscoCdpMIBComplianceV12R02

DESCRIPTION

The compliance statement for the CDP MIB.

MANDATORY-GROUPS { ciscoCdpMIBGroupV11R02 }

```
::= { ciscoCdpMIBCompliances 3 }
```

### ciscoCdpMIBComplianceV12R02 MODULE-COMPLIANCE

STATUS current

DESCRIPTION

The compliance statement for the CDP MIB.

```
MANDATORY-GROUPS { ciscoCdpMIBGroupV12R02 }
::= { ciscoCdpMIBCompliances 4 }
```

## CDP MIB Units of Conformance

### **ciscoCdpMIBGroup OBJECT-GROUP**

```
OBJECTS { cdpInterfaceEnable, cdpInterfaceMessageInterval,
cdpCacheAddressType>cdpCacheAddressType, cdpCacheAddress, cdpCacheVersion, cdpCacheDeviceId,
cdpCacheDevicePort, cdpCacheCapabilities, cdpCachePlatform
```

```
}
```

STATUS obsoleted and superseded by ciscoCdpMIBGroupV11R01

DESCRIPTION

A collection of objects for use with the Cisco Discovery Protocol.

```
::= { ciscoCdpMIBGroups 1 }
```

### **ciscoCdpMIBGroupV11R01 OBJECT-GROUP**

```
OBJECTS { cdpInterfaceEnable, cdpInterfaceMessageInterval, cdpInterfaceGroup, cdpInterfacePort,
cdpCacheAddressType, cdpCacheAddressType, cdpCacheAddress, cdpCacheVersion, cdpCacheDeviceId,
cdpCacheDevicePort,
```

```
cdpCacheCapabilities, cdpCachePlatform
```

```
}
```

STATUS obsoleted and superseded by ciscoCdpMIBGroupV11R02

DESCRIPTION

A collection of objects for use with the Cisco Discovery Protocol.

```
::= { ciscoCdpMIBGroups 2 }
```

### **ciscoCdpMIBGroupV11R02 OBJECT-GROUP**

```
OBJECTS { cdpInterfaceEnable, cdpInterfaceGroup, cdpInterfacePort, cdpCacheAddressType,
cdpCacheAddressType, cdpCacheAddress, cdpCacheVersion, cdpCacheDeviceId, cdpCacheDevicePort,
cdpCacheCapabilities, cdpCachePlatform, cdpGlobalRun, cdpGlobalMessageInterval, cdpGlobalHoldTime
```

```
}
```

STATUS obsoleted and superseded by ciscoCdpMIBGroupV12R02

DESCRIPTION

A collection of objects for use with the Cisco Discovery Protocol.

```
::= { ciscoCdpMIBGroups 3 }
```

### **ciscoCdpMIBGroupV12R02 OBJECT-GROUP**

```
OBJECTS { cdpInterfaceEnable, cdpInterfaceGroup, cdpInterfacePort, cdpCacheAddressType,
cdpCacheAddressType, cdpCacheAddress, cdpCacheVersion, cdpCacheDeviceId, cdpCacheDevicePort,
cdpCacheCapabilities, cdpCachePlatform, cdpCacheVTPMgmtDomain, cdpCacheNativeVLAN,
```

```
cdpCacheDuplex, cdpGlobalRun, cdpGlobalMessageInterval, cdpGlobalHoldTime, cdpGlobalDeviceId
```

```
}
```

STATUS current

DESCRIPTION

A collection of objects for use with the Cisco Discovery Protocol.

::= { ciscoCdpMIBGroups 5 }

#### **ciscoCdpV2MIBGroup OBJECT-GROUP**

OBJECTS { cdpCacheApplianceID, cdpCacheVlanID, cdpCachePowerConsumption, cdpCacheMTU, cdpCacheSysName, cdpCacheSysObjectID, cdpCacheLastChange, cdpCachePhysLocation, cdpCachePrimaryMgmtAddrType, cdpCachePrimaryMgmtAddr, cdpCacheSecondaryMgmtAddrType, cdpCacheSecondaryMgmtAddr, cdpGlobalLastChange, cdpGlobalDeviceIdFormatCpb, cdpGlobalDeviceIdFormat }

STATUS current

DESCRIPTION

A collection of objects for use with the Cisco Discovery Protocol version 2.

::= { ciscoCdpMIBGroups 6 }

#### **ciscoCdpV2IfExtGroup OBJECT-GROUP**

OBJECTS { cdpInterfaceExtendedTrust, cdpInterfaceCosForUntrustedPort }

STATUS current

DESCRIPTION

A collection of objects for use with the Cisco Discovery Protocol version 2 to configure the value for Extended Trust TLV and COS for Untrusted Port TLV.

::= { ciscoCdpMIBGroups 7 }

## **Troubleshoot CDP MIB for Linux and Cisco Unified CM Release 5.x, 6.x, 7.x**

For Linux and Cisco Unified CM Release 5.x, 6.x, 7.x., collect the following logs and information for analysis:

- Use the **set trace enable Detailed cdpmib** CLI set the detailed trace for cdpAgt ().
- Restart the Cisco CDP Agent service from the serviceability Web Page (**Tools > Controlcenter- Network Services**) and wait for some time.
- Collect the following trace files:
  - Enable the Cisco CDP Agent traces by using the file get activelog cm/trace/cdpmib/sdi command and Cisco CDP daemon traces using the file get activelog cm/trace/cdp/sdi command.
  - Enable the Cisco CDP Agent and daemon traces by using the Real-Time Monitoring Tool (RTMT) > Trace & Log Central > Collect Files > Cisco CallManager SNMP Service > Cisco CDP Agent and Cisco CDP.
- Once the logs are collected, reset the trace setting by using the set trace disable cdpmib command.

For Windows and Cisco Unified CM Release 4.x, perform the following to collect logs for analysis.

- Set TraceEnabled to true under the registry HKEY\_LOCAL\_MACHINE\SOFTWARE\Cisco Systems, Inc.\SnmpCDPAgent and restart SNMP service.

- After restarting SNMP service, another option TraceLevel displays. Set this to value 3.
- Restart SNMP service again.
- Do the walk on CDP MIB.
- Collect the log file from location C:\Program Files\Cisco\bin\SnmppCDPImpl.log.
- Collect the output of c:\utils\tlist.exe snmp.exe and output of dir c:\program files\cisco\bin.

## Frequently Asked Questions for CDP MIBs

- [The CDP interface table and globalinfo tables are blank.](#)
- [How is the MessageInterval value set in the Interface table as well as Global table in CDP MIB?](#)

---

### The CDP interface table and globalinfo tables are blank.

Verify that you Cisco Unified CM release that you are using has this capability. If not, upgrade.

### How is the MessageInterval value set in the Interface table as well as Global table in CDP MIB?

Check to see if the HoldTime value is greater than MessageInterval value. If it is less, then the MessageInterval value can not be set from both Interface table as well as Global table.

## CISCO-SYSLOG-MIB




---

**Note** This is a reformatted version of CISCO-SYSLOG-MIB. Download and compile all of the MIBs in this section from <http://tools.cisco.com/Support/SNMP/do/BrowseMIB.do?local=en&step=2>.

This MIB provides a means to gather syslog messages generated by the Cisco IOS. Various textual messages are generated by the Cisco IOS. Cisco IOS can be configured such that these messages are sent to a syslog server. With this MIB these same messages can also be received via the SNMP. These messages are hereupon referred to as syslog messages in this document.




---

**Note** Messages generated as a result of entering CLI debug commands are not made available via the SNMP at this time.

All Cisco IOS syslog messages have timestamps (optional), facility names (where the message came from), severity, message name, and message text. The following example is often seen: %SYS-5-CONFIG\_I: configured from console where facility=SYS, severity=5, message name=CONFIG\_I.

Before you can compile CISCO-SYSLOG-MIB, you need to compile the MIBs listed below in the order listed.

1. SNMPv2-SMI
2. SNMPv2-TC

3. SNMPv2-CONF
4. CISCO-SMI
5. INET-ADDRESS-MIB
6. SNMP-FRAMEWORK-MIB
7. RFC1155-SMI
8. RFC-1212
9. RFC-1215
10. SNMPv2-TC-v1
11. CISCO-SYSLOG-MIB

Additional downloads are:

- OID File: CISCO-SYSLOG-MIB.oid
- Capability File: CISCO-SYSLOG-CAPABILITY

## CISCO-SYSLOG-MIB Revisions

The following table lists the revisions to the MIB beginning with the latest revision.

*Table 8. History of Revisions*

Date	Action	Description
08-07-1995	Initial Version	The MIB module describes how to store the system messages generated by the Cisco IOS software. ::= { ciscoMgmt 41 }

## CISCO-SYSLOG-MIB Definitions

The following definitions are imported for CISCO-SYSLOG-MIB:

- MODULE-IDENTITY, NOTIFICATION-TYPE, OBJECT-TYPE, Integer32, Counter32
- From SNMPv2-SMI—TEXTUAL-CONVENTION, DisplayString, TimeStamp, TruthValue
- From SNMPv2-TC—MODULE-COMPLIANCE, OBJECT-GROUP
- From SNMPv2-CONF—ciscoMgmt
- From CISCO-SMI—ciscoSyslogMIB MODULE-IDENTITY

**ciscoSyslogMIBObjects OBJECT IDENTIFIER ::= { ciscoSyslogMIB 1 }**

## CISCO-SYSLOG-MIB Object Identifiers

```
clogBasicOBJECT IDENTIFIER ::= { ciscoSyslogMIBObjects 1 }
clogHistoryOBJECT IDENTIFIER ::= { ciscoSyslogMIBObjects 2 }
```

## Syslog MIB Textual Conventions

**SyslogSeverity ::= TEXTUAL-CONVENTION**

STATUS current

DESCRIPTION

The severity of a syslog message. The enumeration values are equal to the values that syslog uses + 1. For example, with syslog, emergency=0.

SYNTAX INTEGER { emergency(1), alert(2), critical(3), error(4), warning(5), notice(6), info(7), debug(8) }

## Basic Syslog Objects

**clogNotificationsSent OBJECT-TYPE**

SYNTAX Counter32

UNITS notifications

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The number of clogMessageGenerated notifications that have been sent. This number may include notifications that were prevented from being transmitted due to reasons such as resource limitations and/or non-connectivity. If one is receiving notifications, one can periodically poll this object to determine if any notifications were missed. If so, a poll of the clogHistoryTable might be appropriate.

::= { clogBasic 1 }

**clogNotificationsEnabled OBJECT-TYPE**

SYNTAX TruthValue

MAX-ACCESS read-write

STATUS current

DESCRIPTION

Indicates whether clogMessageGenerated notifications will or will not be sent when a syslog message is generated by the device. Disabling notifications does not prevent syslog messages from being added to the clogHistoryTable.

DEFVAL { false }

::= { clogBasic 2 }

**clogMaxSeverity OBJECT-TYPE**

SYNTAX SyslogSeverity

MAX-ACCESS read-write

STATUS current

DESCRIPTION

Indicates which syslog severity levels will be processed. Any syslog message with a severity value greater than this value will be ignored by the agent.




---

**Note** Severity numeric values increase as their severity decreases, e.g. error(4) is more severe than debug(8).

---

DEFVAL { warning }

::= { clogBasic 3 }

**clogMsgIgnores OBJECT-TYPE**

SYNTAX Counter32

UNITS messages

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The number of syslog messages which were ignored. A message will be ignored if it has a severity value greater than clogMaxSeverity.

::= { clogBasic 4 }

**clogMsgDrops OBJECT-TYPE**

SYNTAX Counter32

UNITS messages

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The number of syslog messages which could not be processed due to lack of system resources. Most likely this will occur at the same time that syslog messages are generated to indicate this lack of resources. Increases in this object's value may serve as an indication that system resource levels should be examined via other mib objects. A message that is dropped will not appear in the history table and no notification will be sent for this message.

::= { clogBasic 5 }

## Syslog MIB Message History Table

### **clogHistTableMaxLength OBJECT-TYPE**

SYNTAX Integer32 (0..500)

UNITS entries

MAX-ACCESS read-write

STATUS current

DESCRIPTION

The upper limit on the number of entries that the clogHistoryTable may contain. A value of zero prevents any history from being retained. When this table is full, the oldest entry will be deleted and a new one will be created.

DEFVAL { 1 }

::= { clogHistory 1 }

### **clogHistMsgsFlushed OBJECT-TYPE**

SYNTAX Counter32

UNITS messages

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The number of entries that have been removed from the clogHistoryTable in order to make room for new entries. This object can be utilized to determine whether your polling frequency on the history table is fast enough and/or the size of your history table is large enough such that you are not missing messages.

::= { clogHistory 2 }

### **clogHistoryTable OBJECT-TYPE**

SYNTAX SEQUENCE OF ClogHistoryEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

A table of syslog messages generated by this device. All 'interesting' syslog messages (i.e. severity <= clogMaxSeverity) are entered into this table.

::= { clogHistory 3 }

### **clogHistoryEntry OBJECT-TYPE**

SYNTAX ClogHistoryEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

A syslog message that was previously generated by this device. Each entry is indexed by a message index.

```
INDEX { clogHistIndex }
 ::= { clogHistoryTable 1 }
```

```
ClogHistoryEntry ::= SEQUENCE { clogHistIndex Integer32, clogHistFacility DisplayString,
clogHistSeverity SyslogSeverity, clogHistMsgName DisplayString, clogHistMsgText DisplayString,
clogHistTimestamp TimeStamp }
```

#### **clogHistIndex OBJECT-TYPE**

```
SYNTAX Integer32 (1..2147483647)
```

```
MAX-ACCESS not-accessible
```

```
STATUS current
```

```
DESCRIPTION
```

A monotonically increasing integer for the sole purpose of indexing messages. When it reaches the maximum value the agent flushes the table and wraps the value back to 1.

```
 ::= { clogHistoryEntry 1 }
```

#### **clogHistFacility OBJECT-TYPE**

```
SYNTAX DisplayString (SIZE (1..20))
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

Name of the facility that generated this message. For example: 'SYS'.

```
 ::= { clogHistoryEntry 2 }
```

#### **clogHistSeverity OBJECT-TYPE**

```
SYNTAX SyslogSeverity
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

The severity of the message.

```
 ::= { clogHistoryEntry 3 }
```

#### **clogHistMsgName OBJECT-TYPE**

```
SYNTAX DisplayString (SIZE (1..30))
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

A textual identification for the message type. A facility name in conjunction with a message name uniquely identifies a message type.

::= { clogHistoryEntry 4 }

**clogHistMsgText OBJECT-TYPE**

SYNTAX DisplayString (SIZE (1..255))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The text of the message. If the text of the message exceeds 255 bytes, the message will be truncated to 254 bytes and a '\*' character will be appended indicating that the message has been truncated.

::= { clogHistoryEntry 5 }

**clogHistTimestamp OBJECT-TYPE**

SYNTAX TimeStamp

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The value of sysUpTime when this message was generated.

::= { clogHistoryEntry 6 }

## Syslog MIB Notifications

**ciscoSyslogMIBNotificationPrefix OBJECT IDENTIFIER**

::= { ciscoSyslogMIB 2 }

**ciscoSyslogMIBNotifications OBJECT IDENTIFIER**

::= { ciscoSyslogMIBNotificationPrefix 0 }

**clogMessageGenerated NOTIFICATION-TYPE**

OBJECTS { clogHistFacility, clogHistSeverity, clogHistMsgName, clogHistMsgText, clogHistTimestamp }

STATUS current

DESCRIPTION

When a syslog message is generated by the device a clogMessageGenerated notification is sent. The sending of these notifications can be enabled/disabled via the clogNotificationsEnabled object.

::= { ciscoSyslogMIBNotifications 1 }

## Syslog MIB Conformance Information

**ciscoSyslogMIBConformance OBJECT IDENTIFIER**

::= { ciscoSyslogMIB 3 }

**ciscoSyslogMIBCompliances OBJECT IDENTIFIER**

```
::= { ciscoSyslogMIBConformance 1 }
```

**ciscoSyslogMIBGroups OBJECT IDENTIFIER**

```
::= { ciscoSyslogMIBConformance 2 }
```

## Syslog MIB Compliance Statements

**ciscoSyslogMIBCompliance MODULE-COMPLIANCE**

STATUS current

DESCRIPTION

The compliance statement for entities which implement the Cisco syslog MIB.

MANDATORY-GROUPS { ciscoSyslogMIBGroup }

```
::= { ciscoSyslogMIBCompliances 1 }
```

## Syslog MIB Units of Conformance

**ciscoSyslogMIBGroup OBJECT-GROUP**

OBJECTS { clogNotificationsSent, clogNotificationsEnabled, clogMaxSeverity, clogMsgIgnores, clogMsgDrops, clogHistTableMaxLength, clogHistMsgsFlushed, clogHistFacility, clogHistSeverity, clogHistMsgName, clogHistMsgText, clogHistTimestamp }

STATUS current

DESCRIPTION

A collection of objects providing the syslog MIB capability.

```
::= { ciscoSyslogMIBGroups 1 }
```

## Troubleshoot Syslog Traps

Syslog has standard buffer size while generating a SNMP trap message; the data is trimmed to the specified field size (255). This avoids any errors caused by data that is too large for the field. For example, if you have specified the message text field to be 255 bytes, but a message arrives that is 300 bytes, the data will be truncated to 255 bytes before being logged.

### Trap Setup

To configure the traps, set `clogsNotificationEnabled` (1.3.6.1.4.1.9.9.41.1.1.2) to TRUE(1) by using SNMP set operation in any SNMP management application. Set the severity using `clogMaxSeverity` (1.3.6.1.4.1.9.9.41.1.1.3) by using any SNMP management application. This object indicates the syslog severity level that needs to be processed. Any syslog message with a severity value greater than this value will be ignored by the agent. Severity numeric values increase as their severity decreases.

Collect the following logs and information:

- Set the detailed trace for `CiscoSyslogAgent` with the **set trace enable Detailed syslogmib** command.

- Restart the Cisco Syslog Agent service from the serviceability Web window **Tools > Control Center - Network Services** and wait for some time.
- Collect the Cisco Syslog Agent trace files by:
  - Using the file **get activelog** cm/trace/syslogmib/sdi/ command.
  - Using **RTMT Trace & Log Central > Collect Files > Cisco CallManager SNMP Service > Cisco Syslog Agent**.
- Once the logs are collected, reset the trace settings by using the **set trace disable syslogmib** command.

## Frequently Asked Questions for Syslog

- Q.** How is a remote syslog server configured?
- A.** You can configure a remote syslog server from Cisco Unified CM Administration **System > Enterprise Parameters** plus the following:
- Remote Syslog Server Name—You can enter the name or IP address of the remote Syslog server that you want to use to accept Syslog messages. If the server name is not specified, Cisco Unified Serviceability does not send the Syslog messages. Do not specify a Cisco Unified Communications Manager server as the destination because the Cisco Unified Communications Manager server does not accept Syslog messages from another server.
    - Maximum length: 255
    - Allowed values: Provide a valid remote syslog server name that comprises (A-Z,a-z,0-9,.,-)
  - Syslog Severity For Remote Syslog messages—You can select the desired Syslog messages severity for remote syslog server. The system sends all the syslog messages with selected or higher severity levels to the remote syslog. If the remote server name is not specified, Cisco Unified Serviceability does not send the Syslog messages.
- Q.** How is a remote syslog server configured to redirect alarms specific to a particular service?
- A.** You can configure a remote syslog server from **Cisco Unified Serviceability** window **Alarm > Configuration**:
- Select the Service Group and Service from drop down list for the particular server.
  - Enable Alarm for Remote Syslogs and set the desired Alarm Event Level. Enter the remote syslog server name or IP address for redirection.

- The system sends all the syslog messages for the particular service with selected or higher severity levels to the remote syslog.

**Q.** How are messages captured in the configured remote server?

**A.** Kiwi Syslog Daemon is a freeware tool which can be installed in the remote server to capture the syslog messages.

**Q.** What happens if the same remote server is configured from Enterprise Parameters and Alarm Configuration page?

- A.**
- Enterprise parameters configuration of remote syslog redirects all the syslog messages which have severity equal to or higher than configured severity. There is no classification done for different types of syslog messages. It is just a plain redirection of all the syslog messages generated.
  - Alarm configuration sends the specific service syslog messages to the configured remote server based on the severity.
  - Enterprise Parameters configuration is used by the Cisco Syslog Agent to send the messages. Corresponding application Alarm configuration will use the alarm interface to send to remote syslog server configured.
  - If the “Local Syslogs” Alarm is enabled in Alarm page, there will be duplication of the service specific messages, incase the same remote server is configured in both pages (provided the severity conditions are matched). For example: Enterprise window has severity level as “Error”, Alarm page has severity “Debug” and “Local syslogs” alarm is enabled. If a syslog message of a particular service configured via alarm page, has a severity higher than 'Debug' and 'Error', then it will be duplicated.

**Q.** Does the SysLog subagent generate traps for the alarms in Syslog automatically? Is there any configuration?

- A.** Syslog subagent can be configured to generate traps for the syslog alarms. Some limitations are:
- Traps are sent out based on selected severity. If the given alarm is of low severity then the management application needs to set the severity threshold lower to capture this low severity alarm/trap. In other words mgmt apps need to deal with flooding of other low severity traps.
  - SNMP Trap message size limited to 255 and not enabled by default. i.e. by default `clogNotificationEnabled (1.3.6.1.4.1.9.9.41.1.1.2)` is set to FALSE (2).

## CISCO-SYSLOG-EXT-MIB



**Note** This is a reformatted version of CISCO-SYSLOG-EXT-MIB. Download and compile all of the MIBs in this section from <http://tools.cisco.com/Support/SNMP/do/BrowseMIB.do?local=en&step=2>.

Before you can compile CISCO-SYSLOG-EXT-MIB, you need to download and compile the MIBs listed below in the order listed.

1. SNMPv2-SMI

2. SNMPv2-TC
3. SNMPv2-CONF
4. CISCO-SMI
5. INET-ADDRESS-MIB
6. SNMP-FRAMEWORK-MIB
7. CISCO-SYSLOG-MIB
8. RFC1155-SMI
9. RFC-1212
10. SNMPv2-TC-v1
11. CISCO-SYSLOG-EXT-MIB

Additional downloads are:

- OID File: CISCO-SYSLOG-EXT-MIB.oid
- Capability File: CISCO-SYSLOG-EXT-CAPABILITY

## CISCO-SYSLOG-EXT-MIB Revisions

The following table lists the revisions to the MIB beginning with the latest revision.

*Table 9: History of Revisions*

Date	Action	Description
12/15/2003	Added	New enumerations. MIB module for configuring and monitoring System Log related management parameters as defined by RFC 3164.
11/13/2002	Added	cseSyslogServerFacility to cseSyslogServerTable. Added two TCs SyslogFacility and SyslogExFacility.
10/04/2002	Initial Version	:= { ciscoMgmt 301 }

## CISCO-SYSLOG-EXT-MIB Definitions

The following definitions are imported for CISCO-SYSLOG-EXT-MIB

- From MODULE-IDENTITY, OBJECT-TYPE, Unsigned32
- From SNMPv2-SMI—MODULE-COMPLIANCE, OBJECT-GROUP

- From SNMPv2-CONF—TruthValue, RowStatus, TEXTUAL-CONVENTION
- From SNMPv2-TC—snmpAdminString
- From SNMP-FRAMEWORK-MIB—inetAddressType, InetAddress
- From INET-ADDRESS-MIB—ciscoMgmt
- From CISCO-SMI—syslogSeverity
- From CISCO-SYSLOG-MIB

**ciscoSyslogExtMIBObjects OBJECT IDENTIFIER**

::= { ciscoSyslogExtMIB 1 }

**cseSyslogConfigurationGroup OBJECT IDENTIFIER**

::= { ciscoSyslogExtMIBObjects 1 }

## Syslog Ext MIB Textual Conventions

**SyslogFacility ::= TEXTUAL-CONVENTION**

STATUS current

DESCRIPTION

The Syslog standard facilities.

REFERENCE

- RFC 3014—The BSD Syslog protocol, Section 4.

SYNTAX INTEGER { kernel (0),-- Kernel user (8), -- User Level mail (16), -- Mail System daemon(24),-- System Daemon auth (32),-- Security/Authorization syslog (40),-- Internal Syslogd lpr (48), -- Line Printer subsystem news (56), -- Network New subsystem uucp (64), -- UUCP subsystem cron (72), -- Clock Daemon authPriv (80), -- Security/Auth(private) ftp (88), -- FTP Daemon local0 (128), -- Reserved local use local1 (136), -- Reserved local use local2 (144), -- Reserved local use local3 (152), -- Reserved local use local4 (160), -- Reserved local use local5 (168), -- Reserved local use local6 (176), -- Reserved local use local7 (184)-- Reserved local use }

**SyslogExFacility ::= TEXTUAL-CONVENTION**

STATUS current

DESCRIPTION

The Syslog facilities including both standard and proprietary facilities.

REFERENCE

- RFC 3014—The BSD Syslog protocol, Section 4.

SYNTAX INTEGER { kernel (0),-- Kernel user (8), -- User Level mail (16), -- Mail System daemon(24), -- System Daemon auth (32),-- Security/Authorization syslog (40),-- Internal Syslogd lpr (48), -- Line Printer subsystem news (56), -- Network New subsystem uucp (64), -- UUCP subsystem cron (72), -- Clock Daemon authPriv (80), -- Security/Auth(private) ftp (88), -- FTP Daemon local0 (128), -- Reserved local use local1 (136), -- Reserved local use local2 (144), -- Reserved local use local3 (152), -- Reserved local use local4 (160), -- Reserved local use local5 (168), -- Reserved local use local6 (176), -- Reserved

local use local7 (184), -- Reserved local use vsanMgr (200), -- VSAN Manager fspf (208), -- FSPF domainMgr (216), -- Domain Manager mtsDaemon (224), -- MTS Daemon linecardMgr (232), -- Line Card Mgr sysMgr (240), -- System Manager sysMgrLib (248), -- System Mgr Library zoneServer (256), -- Zone Server virtualIfMgr (264), -- VirtualInterface Mgr ipConfMgr (272), -- IP Config Manager ipfc (280), -- IP Over FC xBarMgr (288), -- Xbar Manager fcDns (296), -- Fibre Channel DNS fabricConfMgr (304), -- Fabric Config Server aclMgr (312), -- AccessControlList Mgr tlPortMgr (320), -- TL Port Manager portMgr (328), -- Port Manager fportServer (336), -- FPort Server portChMgr (344), -- Port Channel Mgr mpls (352), -- MPLS tftpLib (360), -- TFTP Library wwnMgr (368), -- WWN Mgr fcc (376), -- FCC Process qosMgr (384), -- QOS Mgr vhba (392), -- VHBA procMgr (400), -- Proc Mgr vedbMgr (408), -- VEBD Mgr span (416), -- SPANvrrpMgr (424), -- VRRP Mgr fcfwd (432), -- FCFWD ntp (440), -- NTP pltmfmMgr (448), -- Platform Mgr xbarClient (456), -- XBAR Client vrrpEngine (464), -- VRRP Engine callhome (472), -- Callhome ipsMgr (480), -- IPS Mgr fc2 (488), -- FC2 debugLib (496), -- Debug Library vpm (504), -- VPM mcast (512), -- Multicast rdl (520), -- RDL rscn (536), -- RSCN bootvar (552), -- BootVar pss (576), -- Persistent Storage -- System snmp (584), -- SNMP security (592), -- Security vhead (608), -- VHEAD dns (648), -- DNS rib (656), -- RIB vshd (672), -- VSH Daemon fvpd (688), -- Fabric Virtual Port -- Daemon mplsTunnel (816), -- MPLS Tunnel cdpd (848), -- CDP Daemon ohmsd (920), -- OHMs Daemon portSec (960), -- Port Security Manager ethPortMgr (976), -- Ethernet Port Manager ipaclMgr (1016), -- IP ACL Manager ficonMgr (1064), -- FICON Manager ficonContDev (1096), -- Ficon Control Device rlir (1128), -- RLIR Module fdmi (1136), -- Fabric Device -- Management Interface licmgr (1152), -- License Manager fcspmgr (1160), -- FCSP Manager confCheck (1192), -- Configuration Check ivr (1232), -- Inter-VSAN Routing aaad (1240), -- AAA Daemon tacacsd (1248), -- TACACS Daemon radiusd (1256), -- Radius Daemon fc2d (1320), -- FC2 Daemon lcohmsd (1336), -- LC Ohms Daemon ficonStat (1352), -- FICON Statistics, featureMgr (1360), -- Feature Manager ltt (1376) -- LTT Daemon }

## Syslog Setup Group

This group provides the System log (Syslog) configuration options.

### **cseSyslogConsoleEnable OBJECT-TYPE**

SYNTAX TruthValue

MAX-ACCESS read-write

STATUS current

DESCRIPTION

Indicate whether the Syslog messages should be sent to the console.

DEFVAL { false }

::= { cseSyslogConfigurationGroup 1 }

### **cseSyslogConsoleMsgSeverity OBJECT-TYPE**

SYNTAX SyslogSeverity

MAX-ACCESS read-write

STATUS current

DESCRIPTION

Minimum severity of the message that are sent to the Console.

DEFVAL { debug }

::= { cseSyslogConfigurationGroup 2 }

**cseSyslogLogFileName OBJECT-TYPE**

SYNTAX SnmpAdminString (SIZE (0..255))

MAX-ACCESS read-write

STATUS current

DESCRIPTION

Name of file to which the Syslog messages are logged. Set operation with a zero length will fail.

DEFVAL { "messages" }

::= { cseSyslogConfigurationGroup 3 }

**cseSyslogLogFileMsgSeverity OBJECT-TYPE**

SYNTAX SyslogSeverity

MAX-ACCESS read-write

STATUS current

DESCRIPTION

Minimum severity of the message that are sent to the log file (cseSyslogLogFileName).

DEFVAL { debug }

::= { cseSyslogConfigurationGroup 4 }

**cseSyslogFileLoggingDisable OBJECT-TYPE**

SYNTAX Integer { true (1), noOp (2) }

MAX-ACCESS read-write

STATUS current

DESCRIPTION

Indicates whether the Syslog messages should be sent to the file indicated by cseSyslogLogFileName. Once this object is set to 'true', the Syslog messages are no longer sent to the file. The value of 'cseSyslogLogFileName' is set to zero length string. To restart the file logging, the cseSyslogLogFileName should be set to a valid file name.

No action is taken if this object is set to 'noOp'. The value of the object when read is always 'noOp'.

::= { cseSyslogConfigurationGroup 5 }

**cseSyslogServerTableMaxEntries OBJECT-TYPE**

SYNTAX Unsigned32 (0..65535)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

The maximum number of entries that the agent supports in the cseSyslogServerTable.

::= { cseSyslogConfigurationGroup 6 }

## cseSyslogServerTable

### cseSyslogServerTable OBJECT-TYPE

SYNTAX Sequence of CseSyslogServerEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

This table contains all the Syslog servers which are configured.

::= { cseSyslogConfigurationGroup 7 }

### cseSyslogServerEntry OBJECT-TYPE

SYNTAX CseSyslogServerEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

An entry containing information about a Syslog server.

INDEX { cseSyslogServerIndex }

::= { cseSyslogServerTable 1 }

CseSyslogServerEntry ::=

SEQUENCE { cseSyslogServerIndex Unsigned32, cseSyslogServerAddressType InetAddressType, cseSyslogServerAddress InetAddress, cseSyslogServerMsgSeverity SyslogSeverity, cseSyslogServerStatus RowStatus, cseSyslogServerFacility SyslogFacility }

### cseSyslogServerIndex OBJECT-TYPE

SYNTAX Unsigned32 (1..65535)

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

An arbitrary integer value, greater than zero, and less than and equal to cseSyslogServerTableMaxEntries, which identifies a Syslog server row in this table.

::= { cseSyslogServerEntry 1 }

### cseSyslogServerAddressType OBJECT-TYPE

SYNTAX InetAddressType

MAX-ACCESS read-create

STATUS current

DESCRIPTION

The type of the address of the Syslog server which is given by the corresponding value of cseSyslogServerAddress.

::= { cseSyslogServerEntry 2 }

#### **cseSyslogServerAddress OBJECT-TYPE**

SYNTAX InetAddress

MAX-ACCESS read-create

STATUS current

DESCRIPTION

The address of the Syslog server.

::= { cseSyslogServerEntry 3 }

#### **cseSyslogServerMsgSeverity OBJECT-TYPE**

SYNTAX SyslogSeverity

MAX-ACCESS read-create

STATUS current

DESCRIPTION

Minimum severity of the message that are sent to this Syslog server.

DEFVAL { debug }

::= { cseSyslogServerEntry 4 }

#### **cseSyslogServerStatus OBJECT-TYPE**

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION

The status of this row. A row can not become 'active' until the values for cseSyslogServerAddressType and cseSyslogServerAddress in that row have both been set. A row cannot be created until corresponding instances of following objects are instantiated.

- cseSyslogServerAddressType
- cseSyslogServerAddress

The following objects may not be modified while the value of this object is active (1):

- cseSyslogServerAddressType
- cseSyslogServerAddress.

::= { cseSyslogServerEntry 5 }

#### **cseSyslogServerFacility OBJECT-TYPE**

SYNTAX SyslogFacility

MAX-ACCESS read-create

STATUS current

## DESCRIPTION

The facility to be used when sending Syslog messages to this server.

DEFVAL {local7}

::= { cseSyslogServerEntry 6 }

## cseSyslogMessageControlTable

### cseSyslogMessageControlTable OBJECT-TYPE

SYNTAX Sequence of CseSyslogMessageControlEntry

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

This table contains the information about what system log messages should be sent to Syslog host, console, log file, and/or logged into the internal buffer.

::= { cseSyslogConfigurationGroup 8 }

### cseSyslogMessageControlEntry OBJECT-TYPE

SYNTAX cseSyslogMessageControlEntry

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

A system log message control table entry. Each entry specifies a severity for a particular 'facility' which generates Syslog messages. Any generated message which is at least as severe as the specified severity will be logged.

INDEX { cseSyslogMessageFacility }

::= { cseSyslogMessageControlTable 1 }

CseSyslogMessageControlEntry ::=

SEQUENCE { cseSyslogMessageFacility SyslogExFacility, cseSyslogMessageSeverity SyslogSeverity }

### cseSyslogMessageFacility OBJECT-TYPE

SYNTAX SyslogExFacility

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

System log message facility.

::= { cseSyslogMessageControlEntry 1 }

### cseSyslogMessageSeverity OBJECT-TYPE

SYNTAX SyslogSeverity

MAX-ACCESS read-write

STATUS current

DESCRIPTION

Minimum severity of the message that are generated by this Syslog message facility.

::= { cseSyslogMessageControlEntry 2 }

#### **cseSyslogTerminalEnable OBJECT-TYPE**

SYNTAX TruthValue

MAX-ACCESS read-write

STATUS current

DESCRIPTION

Indicate whether the Syslog messages should be sent to the terminals.

DEFVAL { false }

::= { cseSyslogConfigurationGroup 9 }

#### **cseSyslogTerminalMsgSeverity OBJECT-TYPE**

SYNTAX SyslogSeverity

MAX-ACCESS read-write

STATUS current

DESCRIPTION

Minimum severity of the message that are sent to the terminals.

DEFVAL { debug }

::= { cseSyslogConfigurationGroup 10 }

#### **cseSyslogLinecardEnable OBJECT-TYPE**

SYNTAX TruthValue

MAX-ACCESS read-write

STATUS current

DESCRIPTION

Indicate whether the Syslog messages should be generated at the line cards.

DEFVAL { false }

::= { cseSyslogConfigurationGroup 11 }

#### **cseSyslogLinecardMsgSeverity OBJECT-TYPE**

SYNTAX SyslogSeverity

MAX-ACCESS read-write

STATUS current

DESCRIPTION

Minimum severity of the message that are sent from linecards.

DEFVAL { debug }

::= { cseSyslogConfigurationGroup 12 }

## Syslog Ext MIB Conformance

### **ciscoSyslogExtMIBConformance**

OBJECT IDENTIFIER ::= { ciscoSyslogExtMIB 2 }

### **ciscoSyslogExtMIBCompliances**

OBJECT IDENTIFIER ::= { ciscoSyslogExtMIBConformance 1 }

### **ciscoSyslogExtMIBGroups**

OBJECT IDENTIFIER ::= { ciscoSyslogExtMIBConformance 2 }

### **ciscoSyslogExtMIBCompliance MODULE-COMPLIANCE**

STATUS current

DESCRIPTION

The compliance statement for entities which implement the CISCO-SYSLOG-EXT-MIB.

### **MODULE MANDATORY-GROUPS { ciscoSyslogExtGroup }**

OBJECT cseSyslogServerAddressType

SYNTAX Integer { ipv4 (1), dns (16) }

DESCRIPTION

Only dns and ipv4 addresses are need to be supported.

OBJECT cseSyslogServerStatus

SYNTAX Integer { active (1), createAndGo (4), destroy (6) }

DESCRIPTION

Only three values 'createAndGo', 'destroy' and 'active' need to be supported.

OBJECT cseSyslogLinecardEnable

MIN-ACCESS read-only

DESCRIPTION

Write access is not required.

OBJECT cseSyslogLinecardMsgSeverity

MIN-ACCESS read-only

DESCRIPTION

Write access is not required.

OBJECT cseSyslogMessageFacility

SYNTAX SyslogFacility

## DESCRIPTION

Only the standard facilities need to be supported.

::= { ciscoSyslogExtMIBCompliances 1 }

## Syslog Ext MIB Units of Conformance

### ciscoSyslogExtGroup OBJECT-GROUP

OBJECTS { cseSyslogConsoleEnable, cseSyslogLogFileName, cseSyslogFileLoggingDisable, cseSyslogConsoleMsgSeverity, cseSyslogLogFileMsgSeverity, cseSyslogServerTableMaxEntries, cseSyslogServerAddress, cseSyslogServerAddressType, cseSyslogServerMsgSeverity, cseSyslogServerStatus, cseSyslogServerFacility, cseSyslogMessageSeverity, cseSyslogTerminalEnable, cseSyslogTerminalMsgSeverity, cseSyslogLinecardEnable, cseSyslogLinecardMsgSeverity }

STATUS current

## DESCRIPTION

A collection of objects for Syslog management.

::= { ciscoSyslogExtMIBGroups 1 }

