Cisco Unified Communications Manager Call Detail Records Administration Guide, Release 10.0(1)

First Published: 2013-12-03

Last Modified: 2017-11-03

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Text Part Number: OL-29843-01

CONTENTS

Preface

Change History

Change	Date
Information on CDR OnDemand Service AXIS 1.4 Compatibility With Cisco Unified Communications Manager Release 12.0	March 22, 2018

Purpose

The *Cisco Unified Communications Manager Call Detail Records Administration Guide* describes how to configure call detail records (CDRs) and call management records (CMRs) and provides examples of these records. Use this guide in conjunction with the following documents:

- *CDR Analysis and Reporting Administration Guide*—This document describes how to configure and use Cisco Unified Communications Manager CDR Analysis and Reporting (CAR), a tool that is used to create user, system, device, and billing reports.
- *Cisco Unified Serviceability Administration Guide*—This document provides descriptions and procedures for configuring alarms, traces, SNMP, and so on, through Cisco Unified Serviceability.
- *Real-Time Monitoring Tool Administration Guide* —This document describes how to use Real-Time Monitoring Tool (RTMT), a tool that allows you to monitor many aspects of the system (critical services, alerts, performance counters, and so on).

- *Cisco Unity Connection Serviceability Administration Guide*—This document provides descriptions and procedures for using alarms, traces, reports, and so on, through Cisco Unity Connection Serviceability.

Audience

The *Cisco Unified Communications Manager Call Detail Records Administration Guide* provides information for administrators who are responsible for managing and supporting CDRs. Network engineers, system administrators, or telecom engineers use this guide to learn the content and structure of CDR and CMR records to import them into billing programs and other third-party programs. CAR administrators, managers, and end users use this guide to analyze the information that is generated in certain CAR reports.

Related Documentation

See the *Cisco Unified Communications Manager Documentation Guide* for moreUnified Communications Manager documentation. The following URL shows an example of the path to the documentation guide:

http://www.cisco.com/en/US/docs/voice_ip_comm/cucm/docguide/8_0_1/dg801.html

For more Cisco Unity Connection documentation, see the *Cisco Unity Connection Documentation Guide* at http://www.cisco.com/en/US/products/ps6509/products_documentation_roadmaps_list.html.

Organization

The following table shows how this guide is organized:

Chapter	Description
Overview	
Cisco Call Detail Records, on page 1	Provides an overview of call detail records and an understanding of CDR management.
CDR Processing, on page 5	Describes the procedures for how CDRs are processed.
Call Information Record Types, on page 7	Provides information on call information records.
Call Detail Records	
CDR Examples, on page 15	Provides examples of call detail records.
Cisco Call Detail Records Field Descriptions, on page 127	Describes all call detail record fields.
Cisco Call Detail Records Codes, on page 171	Provides information on all CDR codes, including call termination cause codes, codec type codes, redirect reason codes, and onbehalfof codes.
Call Management Records	
Call Management Records, on page 187	Provides an overview of call management records (CMRs).

Chapter	Description
Cisco Call Management Record Field Descriptions, on page 191	Describes CMR fields.
Cisco Call Management Records K-Factor Data, on page 205	Describes K-Factor data information in the CMR record.
Example Cisco Call Management Records, on page 209	Provides examples of CMRs.

Conventions

This document uses the following conventions:

Convention	Description
boldface font	Commands and keywords are in **boldface**.
italic font	Arguments for which you supply values are in *italics*.
[ ]	Elements in square brackets are optional.
{ x | y | z }	Alternative keywords are grouped in braces and separated by vertical bars.
[x | y | z]	Optional alternative keywords are grouped in brackets and separated by vertical bars.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.
`screen` font	Terminal sessions and information the system displays are in `screen` font.
`boldface screen` font	Information you must enter is in `boldface screen` font.
italic screen font	Arguments for which you supply values are in *italic screen* font.
^	The symbol ^ represents the key labeled Control—for example, the key combination ^D in a screen display means hold down the Control key while you press the D key.
< >	Nonprinting characters, such as passwords, are in angle brackets.

Notes use the following conventions:

Note Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the publication.

Timesavers use the following conventions:

Timesaver Means *the described action saves time*. You can save time by performing the action described in the paragraph.

Tips use the following conventions:

Tip Means *the information contains useful tips.*

Cautions use the following conventions:

Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Warnings use the following conventions:

Warning This warning symbol means danger. You are in a situation that could cause bodily injury. Before you work on any equipment, you must be aware of the hazards involved with electrical circuitry and familiar with standard practices for preventing accidents.

Obtain Documentation and Submit Service Requests

For information on obtaining documentation, submitting a service request, and gathering additional information, see the monthly *What's New in Cisco Product Documentation*, which also lists all new and revised Cisco technical documentation, at:

http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html

Subscribe to the *What's New in Cisco Product Documentation* as a Really Simple Syndication (RSS) feed and set content to be delivered directly to your desktop using a reader application. The RSS feeds are a free service and Cisco currently supports RSS version 2.0.

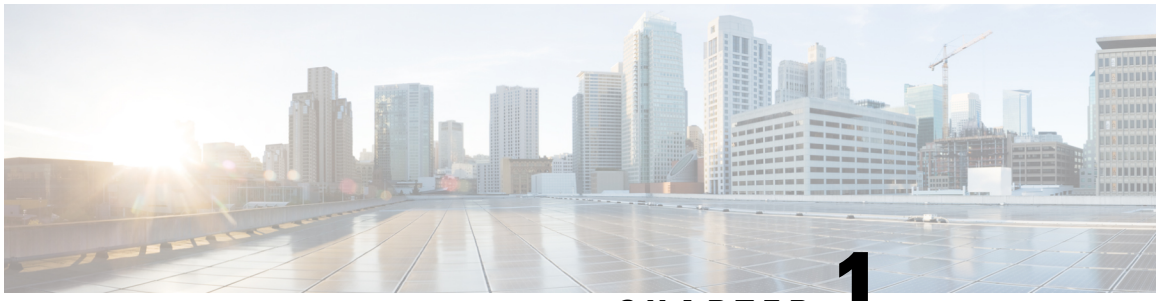
Cisco Product Security Overview

This product contains cryptographic features and is subject to United States and local country laws governing import, export, transfer and use. Delivery of Cisco cryptographic products does not imply third-party authority to import, export, distribute or use encryption. Importers, exporters, distributors and users are responsible for compliance with U.S. and local country laws. By using this product you agree to comply with applicable laws and regulations. If you are unable to comply with U.S. and local laws, return this product immediately.

Further information regarding U.S. export regulations may be found at http://www.access.gpo.gov/bis/ear/ear_data.html.

PART I

Cisco Call Detail Records



CHAPTER 1

Cisco Call Detail Records

This chapter provides information about the format and logic of the call detail records (CDRs) that the Unified Communications Manager system generates. You can use this information for post-processing activities such as generating billing records and network analysis.

When you install your system, the system enables CDRs by default. Call management records (CMRs) remain disabled by default. You can enable or disable CDRs or CMRs at any time that the system is in operation. You do not need to restart Unified Communications Manager for the change to take effect. The system responds to all changes within a few seconds. The system enables CMR or diagnostic data separately from CDR data.

CDR Management

The CDR Management (CDRM) feature, a background application, supports the following capabilities:

- Collects the CDR/CMR files from the Unified Communications Manager server or node to the CDR Repository server or node.
- Collects and maintains the CDR/CMR files on the server where you configure CAR.
- Maintains the CDR/CMR files on the CDR Repository node or CDR server.
- Allows third-party applications to retrieve CDR/CMR files on demand through a SOAP interface.
- Accepts on-demand requests for searching file names.
- Pushes CDR/CMR files from individual nodes within a cluster to the CDR Repository server or node.
- Sends CDR/CMR files to up to three customer billing servers via FTP/SFTP.
- Monitors disk usage of CDR/CMR files on the server where you configure CAR or on the CDR Repository server or node.
- Periodically deletes CDR/CMR files that were successfully delivered. You can configure the amount of storage that is used to store flat files. Predefined storage limits exist. If the storage limits are exceeded, the CDR Repository Manager deletes old files to reduce the disk usage to the preconfigured low water mark. The post-processing applications can later retrieve the buffered historical data to re-get any lost,

corrupted, or missing data. The CDRM feature, which is not aware of the flat file format, does not manipulate the file contents.

Note The CDRM feature handles CDR files and CMR files in the same manner.

CDRM comprises two default services, the CDR Agent and the CDR Repository Manager, and one activate service, CDR onDemand Service.

Related Topics

Call Information Record Types, on page 7
CDR Agent, on page 2
CDR onDemand Service, on page 3
CDR Processing, on page 5
CDR Repository Manager, on page 2
CMR Processing, on page 187

CDR Agent

As part of the CDRM feature, a resident component on the server or node within a Unified Communications Manager installation acts as the CDR Agent. On the server or node where both Unified Communications Manager and the CDR Agent are running, Unified Communications Manager writes the CDRs into CDR flat files in comma separated value (CSV) format. A special control character ("_") that is prefixed to the filename by the call processing module that indicates that the file is not available for transfer. If this control character is not present, the system assumes that the file is available for transfer, and the CDR Agent then SFTPs those files to the designated CDR repository node. Upon successful transfer, the system deletes the local copy of the file.

Reliability gets the highest priority for the CDRM feature. CDRs comprise very important financial data, so the goal of this feature is to guarantee that no CDR is lost. The Unified Communications Manager continuously writes CDRs to flat files, closes existing flat files, and opens new ones. The number of records that are written varies by the type of call and the significant changes that occur during a call: such as, ending the call, transferring the call, redirecting the call, splitting the call, or joining the call.

Note On Linux platforms, the CDR Agent collects the CDR/CMR flat files that the Unified Communications Manager generates and sends these files to the publisher through SFTP. The Windows versions of do not support SFTP. On Windows platforms, the CDR Agent copies the files directly from the subscriber disk to the shared publisher disk.

CDR Repository Manager

Within a Unified Communications Manager server or cluster, one instance of the CDR Repository Manager runs on the CDR Repository server or node. It manages CDR files that are received from the Unified Communications Manager nodes and periodically sends the files to the specified customer/third-party billing servers via FTP/SFTP.

When the file arrives on the CDR Repository server or node, the CDR Repository Manager detects it. The system archives the file in a directory that is dedicated to the date that is indicated by the UTC timestamp that was placed in the file name when the file was created.

If any external billing server is specified in the CDRM configuration, the system creates an empty file in each of the corresponding folders for CAR and the billing servers, if CAR or the corresponding billing server is activated. The CDR Agent monitors new CDR/CMR files that are generated on CallManager servers or nodes by the call processing component. It sends the files to the CDR Repository node and then deletes the local copy after the file is pushed out. The file sender component of the CDR Repository Manager detects these empty files and sends the file to the destination with the specified method. If the delivery is successful, the system removes the empty file in the destination directory.

Every Unified Communications Manager can generate one CDR file and one CMR file every minute for up to 1 hour. You can configure the maximum disk space that is used for storage of CDR files in the CDR Repository through provisioning.

The File Manager component of the CDR Repository Manager runs hourly. When the File Manager runs, it deletes files with dates outside the configured preservation duration. It also checks whether disk usage has exceeded the high water mark. If so, the system deletes the processed CDR files until the low water mark is reached, starting with the oldest files. However, if any CDR file to be deleted was not successfully sent to the specified billing server, the system leaves it in the CDR Repository and raises a notification or alarm. The system creates a flag file during the configured maintenance window, which denies access to the CDR files for the CDR onDemand Service. The system removes the flag file after the maintenance window expires.

For detailed procedures on configuring the CDR Repository Manager and customer billing servers, see the "CDR Repository Manager Configuration" section in the *Cisco Unified Serviceability Administration Guide*.

CDR onDemand Service

The CDR onDemand Service, is a SOAP/HTTPS-based service, that runs on the CDR Repository server or node. It receives SOAP requests for CDR filename lists based on a user-specified time interval (up to a maximum of 1 hour) and returns all lists that fit the duration that the request specifies.

The CDR onDemand Service can also handle requests for delivering a specific CDR file to a specified destination through an SFTP API. All SFTP connections require userid and password information for each session setup. A separate SFTP session gets set up for every file that is sent, and the session is closed after the file has been sent. The system can activate the CDR onDemand service on the CDR Repository node because it has to access the CDR files in the repository. The system prohibits service during the maintenance window. For detailed information on the CDR onDemand Service, see the *Cisco Unified Communications Manager* Developers Guide.

For Cisco Unified Communications Manager Release 10.x and later releases, CDR onDemand Service is not enabled by default. If you want to enable the CDR onDemand service, the service should be activated manually. Execute the following command at the root level to activate the CDR onDemand service:
`/usr/local/cm/bin/soapservicecontrol2.sh CDRonDemandService CDRonDemand deploy 8443`.

Cisco Unified Communications Manager Upgrades and CDR Data

When you upgrade from an earlier version of Unified Communications Manager to a later version of Unified Communications Manager, you may not be able to upgrade all your CDR data. For additional information

about the limitations that affect the amount of CDR data that may be available after upgrade, see the section titled “Upgrading the CAR Database” in the *CDR Analysis and Reporting Administration Guide*. You may also need to refer to the latest *Data Migration Assistant User Guide* and the latest upgrade documentation.

CDR Database Backup and Recovery

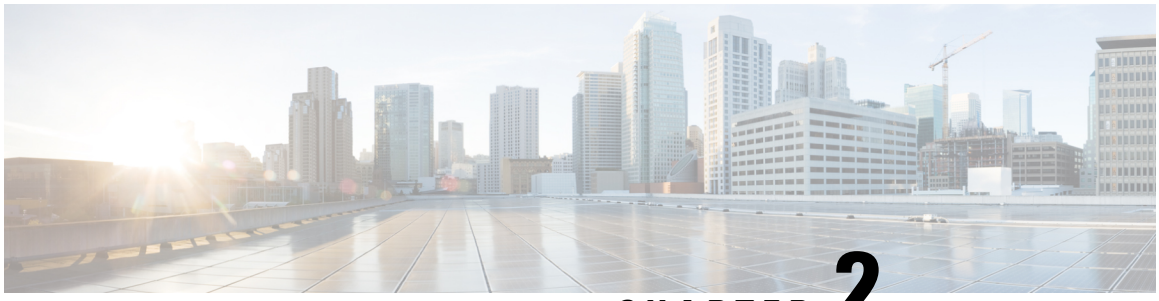
Be sure that the CAR and CDR Disaster Recovery Service (DRS) is integrated into the Cisco Unified Communications Manager DRS.

See the latest release of the *Disaster Recovery System Administration Guide*.

Documentation Related to CDR

The following documents contain additional information related to CDRs:

- *Cisco Unified Serviceability Administration Guide*.
- See the “Configuring the CDR Repository Manager” chapter in the *Cisco Unified Serviceability Administration Guide*.
- *CDR Analysis and Reporting Administration Guide*.
- See the “Activating CAR” section in the Configuring CDR Analysis and Reporting Tool chapter found in the *CDR Analysis and Reporting Administration Guide*.
- *Cisco Unified Communications Manager Developers Guide*
- *Disaster Recovery System Administration Guide*.



CHAPTER 2

CDR Processing

This chapter provides information about how CDRs are processed.

Record Processing

Unified Communications Manager generates two different types of call information records: CDRs and CMRs. The CDR records store information about a call. The CMR records store information about the quality of the streamed audio of the call. The CDR records relate to the CMR records by way of two GlobalCallID columns: Global CallID callManagerId and GlobalCallID Called. Depending upon the call scenario, more than one CMR may exist for each CDR.

When Unified Communications Manager places or receives a call, the system generates a CDR record when the call terminates. The system writes the CDR to a flat file (text file). Inside the Unified Communications Manager, the Call Control process generates CDR records. The system writes records when significant changes occur to a given call, such as ending the call, transferring the call, redirecting the call, splitting the call, joining a call, and so forth.

When CDR records are enabled, Call Control generates one or more CDR records for each call. The system sends these records to EnvProcessCdr, where they are written to the flat files. The number of records that are written varies by type of call and the call scenario. When Diagnostics are enabled, the device generates CMR records for each call. The system writes one CMR record for each IP phone that is involved in the call or for each Media Gateway Control Protocol (MGCP) gateway. The system also sends these records to EnvProcessCdr where they get written to flat files.

The Unified Communications Manager generates CDR and CMR records but does not perform any post processing on the records. The system writes the records to comma-delimited flat files and periodically passes them to the CDR Repository. The CDR and CMR files represent a specific filename format within the flat file.

Filename Format

The following example shows the full format of the filename:
`tag_clusterId_nodeId_datetime_seqNumber`

- **`tag`**—Identifies the type of file, either CDR or CMR.
- **`clusterId`**—Identifies the cluster or server where the Unified Communications Manager database resides.

- **nodeId**—Identifies the node
- **datetime**—UTC time in yyyymmddhhmm format
- **seqnumber**—Sequence number

Two examples of the filenames follow:

```
cdr_Cluster1_01_200404021658_1
cmr_Cluster1_02_200404061011_6125
```

Flat File Format

The CDR and CMR flat files have the following format:

- Line 1—List of field names comma separated
- Line 2—List of field type comma separated
- Line 3—Data comma separated
- Line 4—Data comma separated

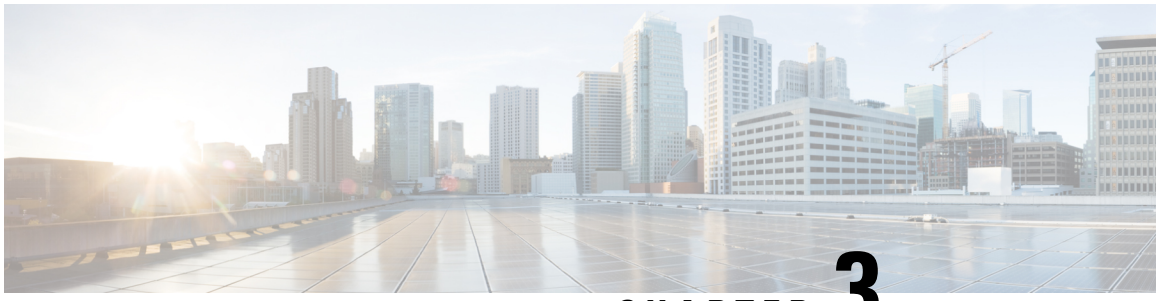
The following example shows a flat file:

```
Line1-"cdrRecordType","globalCallID_callManagerId","globalCallID_callId","origLegCallIdentifier",...
Line2-INTEGER,INTEGER,INTEGER,INTEGER,...
Line3-1,1,388289,17586046,...
Line4-1,1,388293,17586054,...
```

Note If the value of the CDR Log Calls With Zero Duration Flag parameter is True, the system writes all calls to a flat file. See the "Configuring CDR Service Parameters" section in the *CDR Analysis and Reporting Administration Guide* for additional information about this parameter.

Related Topics

Cisco Call Detail Records, on page 1
Cisco Call Management Record Field Descriptions, on page 191
Call Information Record Types, on page 7
Documentation Related to CDR, on page 4



CHAPTER 3

Call Information Record Types

This chapter describes the two types of call information records that Cisco Unified Communications Manager generates: Call Detail Records (CDRs) and Call Management Records (CMRs), also called call diagnostic records.

Call Information Record Types

Unified Communications Manager generates two different types of call information records: Call Detail Records (CDRs) and Call Management Records (CMRs), also called call diagnostic records. CDRs store information about the endpoints of the call and other call control/routing aspects. CMRs contain diagnostic information about the quality of the streamed audio of the call. More than one CMR can exist per CDR.

CMRs are supported by Cisco Unified IP Phones, Cisco 7960 series phones, and Media Gateway Control Protocol (MGCP) gateways. If one of these endpoints is involved in a call, it will generate a CMR record after the call terminates. Each endpoint in the call generates a separate CMR record. If the call involves endpoints that do not support call diagnostics, no record gets generated for that endpoint. A call from a Cisco 7960 phone to an H.323 gateway generates one CMR record (from the Cisco 7960 phone).

CDRs relate to the CMRs via two globalCallID columns:

- globalCallID_callManagerId
- globalCallId_callId

When the Call Diagnostics service parameter is set to True, the system generates up to two CMRs for each call. Each type of call, such as conference calls, call transfers, forwarded calls, and calls through gateways, produce a set of records that gets written to ASCII files at the end of the call. Only completed calls and failed calls generate CDRs and CMRs. Unified Communications Manager does not perform any post processing on CDRs or CMRs.

Related Topics

Call Management Records, on page 187
CDR Processing, on page 5
Cisco Call Detail Records, on page 1
Documentation Related to CDR, on page 4

Global Call Identifier

The Unified Communications Manager allocates a global call identifier (GlobalCallID_callId) each time that a Cisco Unified IP Phone is taken off hook or a call is received from a gateway. The GlobalCallID_callId is allocated sequentially on a Unified Communications Manager server, independent of calls running on other call servers in the cluster. Unified Communications Manager writes the GlobalCallID_callId value to a disk file for every 1,000th call. When Unified Communications Manager restarts for any reason, it assigns the next 1000th number to the next GlobalCallID_callId.

For example, when a successful call gets made, the GlobalCallID_callId value in the CDR specifies 1001. For the next call, the GlobalCallID_callId value specifies 1002, and so on. When Unified Communications Manager restarts, the value for the next call in the CDR gets assigned 2001. The numbers continue sequentially from there until Unified Communications Manager restarts again. For the next restart, the GlobalCallID_callId value specifies 3001.

Note The maximum value that gets assigned to the GlobalCallID_callId is limited to 24 bits. When this limitation occurs, the GlobalCallID_callId value gets reset to 1.

The GlobalCallID_callIds in the CDR file may not be in sequential order in the CDR flat file. If a call with GlobalCallID_callId = 1 lasts longer than the call with GlobalCallID_callId = 2, then the CDR records for GlobalCallId_callId = 2 are written before GlobalCallId_callId = 1. GlobalCallID_callIds may be completely missing from the CDR flat file. If the first CDR record has GlobalCallID_callId = 1, and the second CDR has GlobalCallID_callId = 3, that does not mean that the CDR for GlobalCallID_callId = 2 is missing. GlobalCallID_callId = 2 did not meet the criteria to generate a CDR. The failure to generate a CDR can occur because while the first and third call were successful, the second call was never completed; or, GlobalCallID_callId = 2 could be part of a conference call. Each call leg in a conference call is assigned a GlobalCallID_callId that is overwritten in the conference GlobalCallID_callId. The original GlobalCallID_callId may not appear in the CDR flat file.

If the GlobalCallID_callId field is missing from the CDR record, CAR generates an error for that particular record. Additional information on CDR errors is available in the “Configuring CDR Error Reports” chapter of the *CDR Analysis and Reporting Administration Guide*.

Note For Unified Communications Manager Release 5.x and later releases, the value in the GlobalCallId CDR field survives over Unified Communications Manager restarts. In Release 4.x and earlier releases, even though the GlobalCallId field is time-based, the field gets reused under conditions of heavy traffic. Because of this behavior, problems can occur with customer billing applications and the ability of CAR to correlate CMRs with CDRs and to correlate conference call CDRs. For Release 5.x and later releases, GlobalCallId redesign ensures the field retains a unique value, at least for a certain number of days. Now, the last used globalCallId_callId value gets written to disk periodically (for every x number of calls). The value gets retrieved after a Unified Communications Manager restart, and the new globalCallId_callId value begins with this number plus x.

Number Translations

The Unified Communications Managercan perform translations on the digits that a user dials. The translated number, not the actual dialed digits, appears in the CDR.

For example, many companies translate “911” calls to “9-911,” so the caller does not need to dial an outside line in an emergency. In these cases, the CDR contains “9911” although the user dials “911.”

Note Gateways can perform further modifications to the number before the digits are actually output through the gateway. The CDR does not reflect these modifications.

Partitions and Numbers

Within a CDR, a combination of extension number and partitions identifies each phone that is referenced, if partitions are defined. When partitions exist, fully identifying a phone requires both values because extension numbers may not be unique.

The Partition field stays empty when a call ingresses through a gateway. When a call egresses through a gateway, the Partition field shows the partition to which the gateway belongs.

If the dial plan allows callers to use the # key for speed dialing, the # key goes into the database when it is used. For example, the Called Party Number field may contain a value such as “902087569174#.”

The Party Number fields may include SIP URIs instead of the traditional calling/called party number.

CDRs use the Partition/Extension Numbers that are shown in the following table:

Table 1: Partition/Extension Numbers in CDRs

Phone Number	Description
callingPartyNumber	This party placed the call. For transferred calls, the transferred party becomes the calling party.
originalCalledPartyNumber	This number designates the originally called party, after any digit translations have occurred.

Phone Number	Description
finalCalledPartyNumber	For forwarded calls, this number designates the last party to receive the call. For non-forwarded calls, this field shows the original called party.
lastRedirectDn	For forwarded calls, this field designates the last party to redirect the call. For non-forwarded calls, this field shows the last party to redirect (such as transfer and conference) the call.
callingPartyNumberPartition	This number identifies the partition name that is associated with the CallingPartyNumber field. This field uniquely identifies this number because the Unified Communications Manager supports multiple Cisco Unified IP Phones with the same extension number in different partitions. For calls that ingress through a gateway, this field remains blank.
originalCalledPartyNumberPartition	This number identifies the partition name that is associated with the OriginalCalledPartyNumber field. This field uniquely identifies this number because the Unified Communications Manager supports multiple Cisco Unified IP Phones with the same extension number in different partitions. For calls that egress through a gateway, this field specifies the partition name that is associated with the route pattern that pointed to the gateway.
finalCalledPartyNumberPartition	This number identifies the partition name that is associated with the FinalCalledPartyNumber field. This field uniquely identifies this number because the Unified Communications Manager supports multiple Cisco Unified IP Phones with the same extension number in different partitions. For calls that egress through a gateway, this field specifies the partition name that is associated with the route pattern that pointed to the gateway.

Phone Number	Description
lastRedirectDnPartition	This number identifies the partition name that is associated with the LastRedirectDn field. This field uniquely identifies this number because the Unified Communications Managersupports multiple Cisco Unified IP Phones with the same extension number in different partitions. For calls that egress through a gateway, this field specifies the partition name that is associated with the route pattern that pointed to the gateway.
outpulsedCallingPartyNumber	The calling party number outpulsed from the device.
outpulsedCalledPartyNumber	The called party number outpulsed from the device.

Timestamps

Timestamps within a CDR appear in Universal Coordinated Time (UTC). This value remains independent of daylight saving time changes.

Unsigned 32-bit integers represent all time values. This unsigned integer value displays from the database as a single integer. The field specifies a time_t value that is obtained from the operating system.

The following table displays the UTC timestamps that get included in the CDR.

Table 2: UTC Timestamps in CDRs

Field	Format	Description
dateTimeOrigination	UTC	For outgoing calls, this field designates the time that the device goes off hook. For incoming calls, this field designates the time that the SETUP message is received. This field always gets populated.
dateTimeConnect	UTC	This field designates the time that the devices connect. This field shows a zero if the call never connects.
dateTimeDisconnect	UTC	This field designates the time that the call disconnects. This field gets set even if the call never connects. The time gets stored as UTC. This field always gets populated.

Call Clearing Causes

The CDR includes two call clearing cause codes: OrigCause and DestCause. When the originating party releases the call, the OrigCause gets populated. When the terminating party releases the call, or the call is rejected, the DestCause gets populated. When unpopulated, the cause code value shows zero.

Table 12: Call Termination Cause Codes, on page 174 lists the call clearing cause code values per ITU specification Q.850. For On Net call legs, the Unified Communications Manager determines the cause code value. For Off Net call legs, the far-end switch determines the cause code value.

Convert Signed Decimal Value to IP Address

The system stores IP addresses as unsigned integers. The CDR file displays IP addresses as signed integers. To convert the signed decimal value to an IP address, first convert the value to a hex number, taking into consideration that it is really an unsigned number. The 32-bit hex value represents four bytes in reverse order (Intel standard). To determine the IP address, reverse the order of the bytes and convert each byte to a decimal number. The resulting four bytes represent the four-byte fields of the IP address in dotted decimal notation.

Note The file displays a negative number when the low byte of the IP address has the most significant bit set.

For example, the IP address 192.168.18.188 displays as -1139627840. To convert this IP address, perform the following procedure:

Procedure

Step 1 Convert the database display (-1139627840) to a hex value.

The hex value equals 0xBC12A8C0.

Step 2 Reverse the order of the hex bytes, as shown below:

CO A8 12 BC

Step 3 Convert the four bytes from hex to decimal, as shown below:

192 168 18 188

Step 4 The IP address displays in the dotted decimal format:

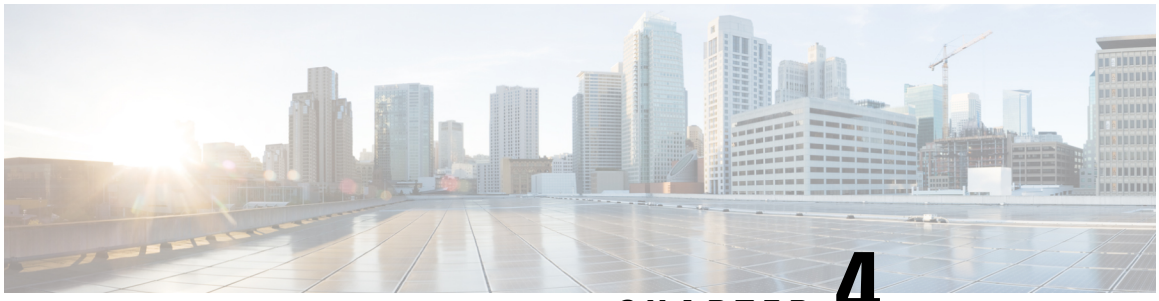
192.168.18.188

What to do next

When working with CDRs, you may want to read other tables in the CAR database to obtain information about the type of device in each CDR because the correlation between devices in the device table and the IP address that is listed in the CDR is not straightforward.

PART II

Call Detail Records



CHAPTER 4

CDR Examples

This chapter provides examples of the call detail records (CDRs) that the Unified Communications ManagerRelease system generates for all call types. You can use this information for post-processing activities such as generating billing records and network analysis.

When you install your system, the system enables CDRs by default. You can enable or disable CDRs at any time that the system is in operation. You do not need to restart Unified Communications Manager for the change to take effect. The system responds to all changes within a few seconds.

AAC Calls

Advanced Audio Coding-Low Delay (AAC-LD) is a super-wideband codec that provides excellent speech and music quality at various bit rates. The audio quality scales up with the bit rate. Two mutually incompatible RTP payload formats are supported: mpeg4-generic and MP4A-LATM.

For AAC-LD (mpeg4-generic) calls, the codec type (payload capability) value 42 is used.

For AAC-LD (MP4A-LATM) calls, a separate codec type value is used for each supported bit rate. The codec type values are 43 (128K), 44 (64K), 45 (56K), 46 (48K), 47 (32K), and 48 (24K).

The system adds an audio bandwidth field to the CDR for AAC-LD calls.

Field Names	Definitions
origMediaCap_bandwidth	This integer field contains the audio bandwidth.
destMediaCap_bandwidth	This integer field contains the audio bandwidth.

The system populates the bandwidth fields based on the following table:

Codec	Bandwidth
G711Alaw64k	64

Codec	Bandwidth
G711Alaw56k	56
G711mu-law64k	64
G711mu-law56k	56
G722 64k	64
G722 56k	56
G722 48k	48
G7231	7
G728	16
G729	8
G729AnnexA	8
Is11172AudioCap	0
Is13818AudioCap	0
G729AnnexB	8
G729AnnexAwAnnexB	8
GSM Full Rate	13
GSM Half Rate	7
GSM Enhanced Full Rate	13
Wideband 256K	256
Data 64k	64
Data 56k	56
G7221 32K	32
G7221 24K	24
AAC-LD (mpeg4-generic)	256
AAC-LD (MP4A-LATM) 128K	128
AAC-LD (MP4A-LATM) 64K	64
AAC-LD (MP4A-LATM) 56K	56
AAC-LD (MP4A-LATM) 48K	48
AAC-LD (MP4A-LATM) 32K	32

Codec	Bandwidth
AAC-LD (MP4A-LATM) 24K	24
GSM	13
iLBC	15 or 13
iSAC	32
XV150 MR 729A	8
NSE VBD 729A	8

AAC-LD (mpeg4-generic) Calls CDR Example

This example applies to a call with AAC-LD (mpeg4-generic) codec:

Field Names	AAC CDR
globalCallID_callId	121
origLegCallIdentifier	101
destLegCallIdentifier	102
callingPartyNumber	51234
originalCalledPartyNumber	57890
finalCalledPartyNumber	57890
lastRedirectDn	57890
origCause_Value	0
dest_CauseValue	16
origMediaCap_payloadCapability	42
origMediaCap_Bandwidth	256
destMediaCap_payloadCapability	42
destMediaCap_Bandwidth	256

Related Topics

Cisco Call Detail Records Field Descriptions, on page 127
Cisco Call Detail Records Codes, on page 171
Example Cisco Call Management Records, on page 209

Abandoned Calls

The logging of calls with zero duration represents an optional action. If logging calls with zero duration is enabled, the following actions occur:

- All calls generate a CDR.
- If the call is abandoned, such as when a phone is taken off hook and placed back on hook, various fields do not contain data. In this case, the originalCalledPartyNumber, finalCalledPartyNumber, the partitions that are associated with them, the destIpAddr, and the dateTimeConnect fields all remain blank. All calls that are not connected have a duration of 0 second. When a call is abandoned, the cause code contains **0**.
- If the user dials a directory number and abandons the call before it connects, the FirstDest and FinalDest fields and their associated partitions contain the directory number and the partition to which the call would have been extended. The DestIp field remains blank, and the duration specifies 0 second.

Note **You must enable the CDR Log Calls With Zero Duration Flag service parameter to log calls with zero duration.** This parameter enables or disables the logging of CDRs for calls which lasted less than 1 second. See the "Configuring CDR Service Parameters" section in the *CDR Analysis and Reporting Administration Guide* for more information.

Examples of Abandoned Calls

1. Extension 2001 goes off hook, then on hook.

Field Names	CDR
globalCallID_callId	1
origLegCallIdentifier	100
destLegCallIdentifier	0
callingPartyNumber	2001
originalCalledPartyNumber	
finalCalledPartyNumber	
lastRedirectDn	
origCause_Value	16
dest_CauseValue	0
duration	0

2. Extension 2001 calls 2309, but 2001 hangs up (abandons) the call before it is answered.

Field Names	CDR
globalCallID_callId	2
origLegCallIdentifier	200
destLegCallIdentifier	201
callingPartyNumber	2001
originalCalledPartyNumber	2309
finalCalledPartyNumber	2309
lastRedirectDn	2309
origCause_Value	16
dest_CauseValue	0
duration	0

Ad Hoc Conference Linking

The advanced ad hoc conference linking feature allows you to link multiple ad hoc conferences together by adding an ad hoc conference to another ad hoc conference as if it were an individual participant. You can also use the methods that are available for adding individual participants to an ad hoc conference to add another conference to an ad hoc conference.

CDRs that the advanced ad hoc conference linking feature generates include a field called OrigConversationId. This field associates the conference bridges that are involved in a linked conference. The Comment field of the CDR adds the ConfRequestorDN and ConfRequestorDeviceName tags to indicate add/drop of participants of the conference by a non-controller of the conference.

The following scenarios show some of the various CDRs:

Related Topics

Conference Linking Using Join, on page 20
Conference Linking Using Transfer or Direct Transfer, on page 22
Linked Conference Party Removal, on page 24
Linked Conference Party (Controller) Removal, on page 26
Linked Conference Removal, on page 28

Conference Linking Using Join

The direction of the call between the bridges depends upon which of the two calls that involve Carol is primary. The primary call survives, and the secondary call gets redirected to the conference.

Alice calls Bob, and Bob conferences in Carol (Conference 1). Dave calls Carol, and conferences in Ed (Conference 2). Two separate conferences get created. Carol exists in both conferences. At this point, CDR1, CDR2, CDR3, and CDR4 get generated.

Carol joins the two conferences. At this point, CDR5 gets generated.

When the remaining parties hang up, the remaining CDRs get generated in the order that the parties leave the conference.

Conference Linking Using Join Example

Field Names	CDR1: Alice -> Bob (original call)	CDR2: Bob -> Carol (consultation call)	CDR3: Dave -> Carol (original call)	CDR4: Dave -> Ed (consultation call)	CDR5: Carol -> Conference Bridge (conference call)	CDR6: Dave -> Conference Bridge (conference call)
globalCallID_callId	1	2	3	4	3	3
origLegCallIdentifier	11	13	21	23	22	22
destLegCallIdentifier	12	14	22	24	25	26
callingPartyNumber	1000	1001	1003	1003	1002	1003
originalCalledPartyNumber	1001	1002	1002	1004	b0029901222	b0029901222
finalCalledPartyNumber	1001	1002	1002	1004	b0029901222	b0029901222
lastRedirectDn	1001	1002	1002	1004	1003	1003
origTerminationOnBehalfOf	4	4	4	4	4	4
destTerminationOnBehalfOf	4	4	4	4	4	4
lastRedirectRedirectReason	0	0	0	0	98	98
lastRedirectRedirectOnBehalfOf	0	0	0	0	4	4
origConversationID	0	0	0	0	0	0
destConversationID	0	0	0	0	2222	2222
Comment					`ConfControllerDn=1001;ConfControllerDeviceName=SEP0003E333FEBD;ConfRequestorDn-1001;ConfRequestorDeviceName=SEP0003E333FEBD`	`ConfControllerDn=1003;ConfControllerDeviceName=SEP0003E333FAD1;ConfRequestorDn-1003;ConfRequestorDeviceName=SEP0003E333FAD1`

Field Names	CDR7: Dave -> Conference Bridge (conference call)	CDR8: Ed -> Conference Bridge (conference call)	CDR9: Conference Bridge (conference call)	CDR10: Alice -> Conference Bridge (conference call)	CDR11: Bob -> Conference Bridge (conference call)
globalCallID_callId	3	3	1	1	1

Field Names	CDR7: Dave -> Conference Bridge (conference call)	CDR8: Ed -> Conference Bridge (conference call)	CDR9: Conference Bridge (conference call)	CDR10: Alice -> Conference Bridge (conference call)	CDR11: Bob -> Conference Bridge (conference call)
origLegCallIdentifier	21	24	17	11	12
destLegCallIdentifier	26	27	28	15	16
callingPartyNumber	1003	1004	b0029901001	1000	1001
originalCalledPartyNumber	b0029901222	b0029901222	b0029901222	b0029901001	b0029901001
finalCalledPartyNumber	b0029901222	b0029901222	b0029901222	b0029901001	b0029901001
lastRedirectDn	1003	1003	1002	1001	1001
origTerminationOnBehalfOf	0	0	0	0	0
destTerminationOnBehalfOf	0	0	0	0	0
lastRedirectRedirectReason	98	98	4	98	98
lastRedirectRedirectOnBehalfOf	4	4	10	4	4
origConversationID	0	0	0	0	0
destConversationID	2222	2222	2222	2222	2222
Comment	ConfControllerDn=1003;ConfControllerDeviceName=SEP0003E333FAD1;ConfRequestorDn-1003;ConfRequestorDeviceName=SEP0003E333FAD1	ConfControllerDn=1003;ConfControllerDeviceName=SEP0003E333FAD1;ConfRequestorDn-1003;ConfRequestorDeviceName=SEP0003E333FAD1	ConfControllerDn=1003;ConfControllerDeviceName=SEP0003E333FAD1;ConfRequestorDn-1003;ConfRequestorDeviceName=SEP0003E333FAD1	ConfControllerDn=1001;ConfControllerDeviceName=SEP0003E333FEBD;ConfRequestorDn-1001;ConfRequestorDeviceName=SEP0003E333FEBD	ConfControllerDn=1001;ConfControllerDeviceName=SEP0003E333FEBD;ConfRequestorDn-1001;ConfRequestorDeviceName=SEP0003E333FEBD

Conference Linking Using Transfer or Direct Transfer

Alice calls Bob, and Bob conferences Carol (Conference 1). Dave calls Carol and conferences in Ed (Conference 2). Two separate conferences get created; Carol exists in both conferences. At this point, CDR1, CDR2, CDR3, and CDR4 get generated.

Carol presses the **Direct Transfer** (DirTrfr) softkey on the call to the first conference. Alice and Bob exist in Conference 1 while Dave and Ed are in Conference 2. When the remaining parties hang up, the remaining CDRs get generated in the order in which the parties leave the conference.

Note The direction of the call between the bridges depends on which of the two calls that involve Carol is the primary call. The primary call side represents the calling party of the transferred call.

Conference Linking Using Transfer or Direct Transfer Example

Field Names	CDR1: Alice -> Bob (original call)	CDR2: Bob -> Carol (consultation call)	CDR3: Dave -> Carol (original call)	CDR4: Dave -> Carol (consultation call)	CDR5: Carol -> Conference Bridge (conference call)	CDR6: Carol -> Conference Bridge (conference call)
globalCallID_callId	1	2	3	4	1	3
origLegCallIdentifier	11	13	21	23	14	22
destLegCallIdentifier	12	14	22	24	17	25
callingPartyNumber	1000	1001	1003	1003	1002	1003
originalCalledPartyNumber	1001	1002	1002	1004	b0029901001	b0029901222
finalCalledPartyNumber	1001	1002	1002	1004	b0029901001	b0029901222
lastRedirectDn	1001	1002	1002	1004	1001	1003
origTerminationOnBehalfOf	4	4	4	4	10	10
destTerminationOnBehalfOf	4	4	4	4	10	10
lastRedirectRedirectReason	0	0	0	0	98	98
lastRedirectRedirectOnBehalfOf	0	0	0	0	4	4
origConversationID	0	0	0	0	0	0
destConversationID	0	0	0	0	1111	2222
Comment					`ConfControllerDn=1001;ConfControllerDeviceName=SEP0003E333FEBD;ConfRequestorDn-1001;ConfRequestorDeviceName=SEP0003E333FEBD`	`ConfControllerDn=1003;ConfControllerDeviceName=SEP0003E333FAD1;ConfRequestorDn-1003;ConfRequestorDeviceName=SEP0003E333FAD1`

Field Names	CDR7: Dave -> Conference Bridge (conference call)	CDR8: Ed -> Conference Bridge (conference call)	CDR9: Conference Bridge (conference call)	CDR10: Alice -> Conference Bridge (conference call)	CDR11: Bob -> Conference Bridge (conference call)
globalCallID_callId	3	3	1	1	1
origLegCallIdentifier	21	24	17	11	12
destLegCallIdentifier	26	27	28	15	16
callingPartyNumber	1003	1004	b0029901001	1000	1001

Field Names	CDR7: Dave -> Conference Bridge (conference call)	CDR8: Ed -> Conference Bridge (conference call)	CDR9: Conference Bridge (conference call)	CDR10: Alice -> Conference Bridge (conference call)	CDR11: Bob -> Conference Bridge (conference call)
originalCalledPartyNumber	b0029901222	b0029901222	b0029901222	b0029901001	b0029901001
finalCalledPartyNumber	b0029901222	b0029901222	b0029901222	b0029901001	b0029901001
lastRedirectDn	1003	1003	1002	1001	1001
origTerminationOnBehalfOf	0	0	0	0	0
destTerminationOnBehalfOf	0	0	0	0	0
lastRedirectRedirectReason	98	98	4	98	98
lastRedirectRedirectOnBehalfOf	4	4	10	4	4
origConversationID	0	0	111	0	0
destConversationID	2222	2222	2222	1111	1111
Comment	ConfControllerDn=1003;ConfControllerDeviceName=SEP0003E333FAD1;ConfRequestorDn-1003;ConfRequestorDeviceName=SEP0003E333FAD1	ConfControllerDn=1003;ConfControllerDeviceName=SEP0003E333FAD1;ConfRequestorDn-1003;ConfRequestorDeviceName=SEP0003E333FAD1	ConfControllerDn=1003;ConfControllerDeviceName=SEP0003E333FAD1;ConfRequestorDn-1003;ConfRequestorDeviceName=SEP0003E333FAD1	ConfControllerDn=1001;ConfControllerDeviceName=SEP0003E333FEBD;ConfRequestorDn-1001;ConfRequestorDeviceName=SEP0003E333FEBD	ConfControllerDn=1001;ConfControllerDeviceName=SEP0003E333FEBD;ConfRequestorDn-1001;ConfRequestorDeviceName=SEP0003E333FEBD

Linked Conference Party Removal

CDRs get generated in the order in which the parties leave a conference. When the remaining conference only has two parties, the two parties get joined directly together.

Alice calls Bob, and Bob conferences Carol (Conference 1). Dave calls Carol, and conferences in Ed (Conference 2). Two separate conferences get created; Carol participates in both conferences. At this point, CDR1, CDR2, CDR3, and CDR4 get generated.

Carol presses the **Direct Transfer** (DirTrfr) softkey on the call to the first conference. Alice and Bob exist in Conference 1 while Dave and Ed are in Conference 2. Conference 1 and Conference 2 get transferred together. Carol hangs up and leaves only two parties in Conference 1.

Because only two parties exist in the conference, Bob and the conference link get joined together. At this point, CDR7, CDR8, and CDR9 get generated. Because Bob is the controller in Conference 1, Bob represents the calling party in the call between Bob and Conference 2. When the remaining parties hang up, the remaining CDRs get generated in the order in which the parties leave the conference.

Note If Bob is not a controller and the chaining occurs before Bob joins Conference 1, the call between Bob and Conference 2 gets generated in the opposite direction from what is shown in the CDRs.

The direction of the call between the final two parties of a conference depends on who has been in the conference the longest. The party that has been in the conference the longest becomes the calling party.

Removing a Party From a Linked Conference Example

Field Names	CDR1: Alice -> Bob (original call)	CDR2: Bob -> Carol (consultation call)	CDR3: Dave -> Carol (original call)	CDR4: Dave -> Carol (consultation call)	CDR5: Carol -> Conference Bridge (conference call)	CDR6: Carol -> Conference Bridge (conference call)
globalCallID_callId	1	2	3	4	1	3
origLegCallIdentifier	11	13	21	23	14	22
destLegCallIdentifier	12	14	22	24	17	25
callingPartyNumber	1000	1001	1003	1003	1002	1002
originalCalledPartyNumber	1001	1002	1002	1004	b0029901001	b0029901222
finalCalledPartyNumber	1001	1002	1002	1004	b0029901001	b0029901222
lastRedirectDn	1001	1002	1002	1004	1001	1003
origTerminationOnBehalfOf	4	4	4	4	10	10
destTerminationOnBehalfOf	4	4	4	4	10	10
lastRedirectRedirectReason	0	0	0	0	98	98
lastRedirectRedirectOnBehalfOf	0	0	0	0	4	4
origConversationID	0	0	0	0	0	0
destConversationID	0	0	0	0	1111	2222
Comment					`ConfControllerDn=1001;ConfControllerDeviceName=SEP0003E333FEBD;ConfRequestorDn-1001;ConfRequestorDeviceName=SEP0003E333FEBD`	`ConfControllerDn=1003;ConfControllerDeviceName=SEP0003E333FAD1;ConfRequestorDn-1003;ConfRequestorDeviceName=SEP0003E333FAD1`

Field Names	CDR7: Alice -> Conference Bridge (conference call)	CDR8: Bob -> Conference Bridge (conference call)	CDR9: Conference Bridge -> Conference Bridge	CDR10: Bob -> Conference Bridge (conference call)	CDR11: Dave -> Conference Bridge (conference call)	CDR12: Ed -> Conference Bridge (conference call)
globalCallID_callId	1	1	3	3	3	3
origLegCallIdentifier	11	12	25	11	12	24
destLegCallIdentifier	15	16	28	15	16	27
callingPartyNumber	1001	1001	b0029901222	1000	1001	1003
originalCalledPartyNumber	b0029901001	b0029901001	b0029901001	b0029901222	b0029901222	b0029901222
finalCalledPartyNumber	b0029901001	b0029901001	b0029901001	b0029901222	b0029901222	b0029901222
lastRedirectDn	1001	1001	1002	b0029901001	1003	1003
origTerminationOnBehalfOf	16	4	4	4	0	0
destTerminationOnBehalfOf	0	4	4	4	0	0
lastRedirectRedirectReason	98	98	4	98	98	98
lastRedirectRedirectOnBehalfOf	4	4	10	4	4	4
origConversationID	0	0	2222	0	0	0
destConversationID	1111	1111	1111	2222	2222	2222
Comment	ConfControllerDn=1001;ConfControllerDeviceName=SEP0003E333FEBD;ConfRequestorDn-1001;ConfRequestorDeviceName=SEP0003E333FEBD	ConfControllerDn=1001;ConfControllerDeviceName=SEP0003E333FEBD;ConfRequestorDn-1001;ConfRequestorDeviceName=SEP0003E333FEBD	ConfControllerDn=1003;ConfControllerDeviceName=SEP0003E333FAD1;ConfRequestorDn-1003;ConfRequestorDeviceName=SEP0003E333FAD1	ConfControllerDn=1003;ConfControllerDeviceName=SEP0003E333FAD1;ConfRequestorDn-1003;ConfRequestorDeviceName=SEP0003E333FAD1	ConfControllerDn=1003;ConfControllerDeviceName=SEP0003E333FAD1;ConfRequestorDn-1003;ConfRequestorDeviceName=SEP0003E333FAD1	ConfControllerDn=1003;ConfControllerDeviceName=SEP0003E333FAD1;ConfRequestorDn-1003;ConfRequestorDeviceName=SEP0003E333FAD1

Linked Conference Party (Controller) Removal

CDRs get generated in the order in which the parties leave a conference. When the remaining conference only has two parties, these two parties get joined directly together.

Alice calls Bob, and Bob conferences Carol (Conference 1). Dave calls Carol and conferences in Ed (Conference 2). Two separate conferences get created; Carol participates in both conferences. At this point, CDR1, CDR2, CDR3, and CDR4 get generated.

Carol presses the **Direct Transfer** (DirTrfr) softkey on the call to the first conference. Alice and Bob exist in Conference 1, while Dave and Ed are in Conference 2. Conference 1 and Conference 2 get transferred together. Bob hangs up which leaves only two parties that are connected to Conference 1.

Because only two parties exist in Conference1, Alice and the conference link get joined directly together. At this point, CDR7, CDR8, and CDR9 get generated. Because Alice has been in the conference longer, she becomes the calling party in the call between Alice and Conference 2. When the remaining parties hang up, the remaining CDRs get generated in the order in which the parties leave the conference.

Note The direction of a call between the final two parties of a conference depends on who has been in the conference the longest. The party that has been in the conference the longest becomes the calling party.

Removing a Controller From a Linked Conference Example

Field Names	CDR1: Alice -> Bob (original call)	CDR2: Bob -> Carol (consultation call)	CDR3: Dave -> Carol (original call)	CDR4: Dave -> Carol (consultation call)	CDR5: Carol -> Conference Bridge (conference call)	CDR6: Carol -> Conference Bridge (conference call)
globalCallID_callId	1	2	3	4	1	3
origLegCallIdentifier	11	13	21	23	14	22
destLegCallIdentifier	12	14	22	24	17	25
callingPartyNumber	1000	1001	1003	1003	1002	1002
originalCalledPartyNumber	1001	1002	1002	1004	b0029901001	b0029901222
finalCalledPartyNumber	1001	1002	1002	1004	b0029901001	b0029901222
lastRedirectDn	1001	1002	1002	1004	1001	1003
origTerminationOnBehalfOf	4	4	4	4	10	10
destTerminationOnBehalfOf	4	4	4	4	10	10
lastRedirectRedirectReason	0	0	0	0	98	98
lastRedirectRedirectOnBehalfOf	0	0	0	0	4	4
origConversationID	0	0	0	0	0	0
destConversationID	0	0	0	0	1111	2222
Comment					`ConfControllerDn=1001;ConfControllerDeviceName=SEP0003E333FEBD;ConfRequestorDn-1001;ConfRequestorDeviceName=SEP0003E333FEBD`	`ConfControllerDn=1003;ConfControllerDeviceName=SEP0003E333FAD1;ConfRequestorDn-1003;ConfRequestorDeviceName=SEP0003E333FAD1`

Field Names	CDR7: Conference Bridge -> Conference Bridge	CDR8: Alice -> Conference Bridge (conference call)	CDR9: Conference Bridge -> Conference Bridge	CDR10: Alice -> Conference Bridge (conference call)	CDR11: Dave -> Conference Bridge (conference call)	CDR12: Ed -> Conference Bridge (conference call)
globalCallID_callId	1	1	3	3	3	3
origLegCallIdentifier	12	11	25	11	21	24
destLegCallIdentifier	16	15	28	25	26	27
callingPartyNumber	1001	1000	b0029901001	1001	1003	1004
originalCalledPartyNumber	b0029901001	b0029901001	b0029901001	b0029901222	b0029901222	b0029901222
finalCalledPartyNumber	b0029901001	b0029901001	b0029901001	b0029901222	b0029901222	b0029901222
lastRedirectDn	1001	1001	1002	b0029901001	1003	1003
origTerminationOnBehalfOf	4	16	4	4	0	0
destTerminationOnBehalfOf	4	0	4	4	0	0
lastRedirectRedirectReason	98	98	4	98	98	98
lastRedirectRedirectOnBehalfOf	4	4	10	4	4	4
origConversationID	0	0	2222	0	0	0
destConversationID	1111	1111	1111	2222	2222	2222
Comment	ConfControllerDn=1001;ConfControllerDeviceName=SEP0003E333FEBD;ConfRequestorDn-1001;ConfRequestorDeviceName=SEP0003E333FEBD	ConfControllerDn=1001;ConfControllerDeviceName=SEP0003E333FEBD;ConfRequestorDn-1001;ConfRequestorDeviceName=SEP0003E333FEBD	ConfControllerDn=1003;ConfControllerDeviceName=SEP0003E333FAD1;ConfRequestorDn-1003;ConfRequestorDeviceName=SEP0003E333FAD1	ConfControllerDn=1003;ConfControllerDeviceName=SEP0003E333FAD1;ConfRequestorDn-1003;ConfRequestorDeviceName=SEP0003E333FAD1	ConfControllerDn=1003;ConfControllerDeviceName=SEP0003E333FAD1;ConfRequestorDn-1003;ConfRequestorDeviceName=SEP0003E333FAD1	ConfControllerDn=1003;ConfControllerDeviceName=SEP0003E333FAD1;ConfRequestorDn-1003;ConfRequestorDeviceName=SEP0003E333FAD1

Linked Conference Removal

Alice calls Bob, and Bob conferences Carol (Conference 1). Dave calls Carol, and conferences in Ed (Conference 2). Two separate conferences get created; Carol participates in both conferences. At this point, CDR1, CDR2, CDR3, and CDR4 get generated.

Carol presses the **Direct Transfer** (DirTrfr) softkey on the call to the first conference. Alice and Bob exist in Conference 1, while Dave and Ed are in Conference 2. Conference 1 and Conference 2 get transferred together.

Bob presses the ConfList softkey and has Alice, Bob, and the conference link "Conference" shown in the list. Bob selects "Conference" and presses the **Remove** softkey. At this point, CDR7, CDR8, and CDR9 get generated. The conference link gets removed, which leaves two parties in the conference.

The remaining two parties get joined together. In Conference 1, Alice and Bob get joined together, and in Conference 2, Dave and Ed get joined together. When the remaining parties hang up, the remaining CDRs get generated in the order in which the parties leave the conference.

Removing the Linked Conference Example

Field Names	CDR1: Alice -> Bob (original call)	CDR2: Bob -> Carol (consultation call)	CDR3: Dave -> Carol (original call)	CDR4: Dave -> Carol (consultation call)	CDR5: Carol -> Conference Bridge (conference call)	CDR6: Carol -> Conference Bridge (conference call)
globalCallID_callId	1	2	3	4	1	3
origLegCallIdentifier	11	13	21	23	14	22
destLegCallIdentifier	12	14	22	24	17	25
callingPartyNumber	1000	1001	1003	1003	1002	1002
originalCalledPartyNumber	1001	1002	1002	1004	b0029901001	b0029901222
finalCalledPartyNumber	1001	1002	1002	1004	b0029901001	b0029901222
lastRedirectDn	1001	1002	1002	1004	1001	1003
origTerminationOnBehalfOf	4	4	4	4	10	10
destTerminationOnBehalfOf	4	4	4	4	10	10
lastRedirectRedirectReason	0	0	0	0	98	98
lastRedirectRedirectOnBehalfOf	0	0	0	0	4	4
origConversationID	0	0	0	0	0	0
destConversationID	0	0	0	0	1111	2222
Comment					`ConfControllerDn=1001;ConfControllerDeviceName=SEP0003E333FEBD;ConfRequestorDn-1001;ConfRequestorDeviceName=SEP0003E333FEBD`	`ConfControllerDn=1003;ConfControllerDeviceName=SEP0003E333FAD1;ConfRequestorDn-1003;ConfRequestorDeviceName=SEP0003E333FAD1`

Field Names	CDR7: Conference Bridge -> Conference Bridge	CDR8: Alice -> Conference Bridge (conference call)	CDR9: Bob -> Conference Bridge	CDR10: Dave -> Conference Bridge (conference call)	CDR11: Ed -> Conference Bridge (conference call)	CDR12: Bob-> Alice
globalCallID_callId	3	1	1	3	3	3
origLegCallIdentifier	25	11	12	21	24	21
destLegCallIdentifier	28	15	16	26	27	24
callingPartyNumber	b0029901222	1000	1001	1003	1004	1003
originalCalledPartyNumber	b0029901001	b0029901001	b0029901001	b0029901222	b0029901222	b0029901222
finalCalledPartyNumber	b0029901001	b0029901001	b0029901001	b0029901222	b0029901222	1004
lastRedirectDn	1002	1001	1001	1003	1003	b0029901222
origTerminationOnBehalfOf	4	4	4	16	0	0
destTerminationOnBehalfOf	4	4	4	0	0	0
lastRedirectRedirectReason	4	98	98	98	98	98
lastRedirectRedirectOnBehalfOf	10	4	4	4	4	4
origConversationID	0	0	0	0	0	0
destConversationID	1111	1111	1111	2222	2222	0
Comment	`ConfControllerDn=1003;ConfControllerDeviceName=SEP0003E333FAD1;ConfRequestorDn-1003;ConfRequestorDeviceName=SEP0003E333FAD1`	`ConfControllerDn=1001;ConfControllerDeviceName=SEP0003E333FEBD;ConfRequestorDn-1001;ConfRequestorDeviceName=SEP0003E333FEBD`	`ConfControllerDn=1001;ConfControllerDeviceName=SEP0003E333FEBD;ConfRequestorDn-1001;ConfRequestorDeviceName=SEP0003E333FEBD`	`ConfControllerDn=1003;ConfControllerDeviceName=SEP0003E333FAD1;ConfRequestorDn-1003;ConfRequestorDeviceName=SEP0003E333FAD1`	`ConfControllerDn=1003;ConfControllerDeviceName=SEP0003E333FAD1;ConfRequestorDn-1003;ConfRequestorDeviceName=SEP0003E333FAD1`	`ConfControllerDn=1003;ConfControllerDeviceName=SEP0003E333FAD1;ConfRequestorDn-1003;ConfRequestorDeviceName=SEP0003E333FAD1`

Field Names	CDR13: Dave -> Ed
globalCallID_callId	3
origLegCallIdentifier	21
destLegCallIdentifier	24
callingPartyNumber	1003
originalCalledPartyNumber	b0029901222
finalCalledPartyNumber	1004

Field Names	CDR13: Dave -> Ed
lastRedirectDn	b0029901222
origTerminationOnBehalfOf	0
destTerminationOnBehalfOf	0
lastRedirectRedirectReason	98
lastRedirectRedirectOnBehalfOf	4
origConversationID	0
destConversationID	0
Comment	`ConfControllerDn=1003;ConfControllerDeviceName=SEP0003E333FAD1;ConfRequestorDn-1003;ConfRequestorDeviceName=SEP0003E333FAD1`

Agent Greeting Calls

The Agent Greeting call feature instructs Unified Communications Manager to play a prerecorded announcement to the customer automatically after successful media connection to the agent device occurs. Both the agent and the customer hear the Agent Greeting.

Example of an Agent Greeting Call

1. The customer (1001) calls the agent (1006).
2. The agent (1006) answers the call. The customer and the agent connect.
3. The Agent Greeting call feature instructs Unified Communications Manager to play a prerecorded announcement to the customer automatically after successful media connection to the agent device occurs. This causes an IVR (1000) to connect to the Built-In Bridge (BIB) of agent phone. Both the agent and the customer hear the Agent Greeting.
4. The customer-agent call ends. A CDR gets generated for the customer-to-agent call. A CDR gets generated for the IVR (1000) to BIB of agent phone.

The CDR for the IVR to agent BIB specifies the comment **`AgentGreeting=<agentCI>`**. The OnBehalfOf field is set to 33 and redirectReason code is set to 752 for Agent Greeting call.

Field Names	Call From Customer to Agent	Call From IVR to Agent BIB
globalCallID_callId	270001	270002
origLegCallIdentifier	22980857	22980861
destLegCallIdentifier	22980858	22980862

Field Names	Call From Customer to Agent	Call From IVR to Agent BIB
callingPartyNumber	1001	1000
originalCalledPartyNumber	1006	b00121104001
finalCalledPartyNumber	1006	b00121104001
origCallTerminationOnBehalfOf	12	0
destCallTerminationOnBehalfOf	0	33
origCalledPartyRedirectOnBehalfOf	0	33
lastRedirectRedirectOnBehalfOf	0	33
origCalledPartyRedirectReason	0	752
lastRedirectRedirectReason	0	752
destConversationId	0	22980858
joinOnBehalfOf		33
comment		AgentGreeting=22980858
duration	23	9

Related Topics

Cisco Call Detail Records Field Descriptions, on page 127
Cisco Call Detail Records Codes, on page 171
Example Cisco Call Management Records, on page 209

Barge

When a shared line uses the barge feature, the **origCalledPartyNumber**, **finalCalledPartyNumber,** and **lastRedirectDn** represent the conference bridge number 'b00. . .'. The redirect and join OnBehalfOf fields reflect a value of Barge = 15, and the redirect reason fields specify Barge = 114.

Barge Examples

1. 40003 calls 40001, and 40001 answers. Shared line 40001' on another phone presses the Barge softkey. All the parties get conferenced together; then, 40003 hangs up.

Note Both CDRs have the same globalCallID_callId, and the conversationID field links back to the CI (call Identifier) of the barged call.

Field Names	Original Call CDR	Barge Call CDR
globalCallID_callId	7	7

Field Names	Original Call CDR	Barge Call CDR
origLegCallIdentifier	16777230	16777232
destLegCallIdentifier	16777231	16777235
callingPartyNumber	40003	40003
origCalledPartyNumber	40001	b001501001
finalCalledPartyNumber	40001	b001501001
lastRedirectDn	40001	b001501001
origCause_Value	16	0
dest_CauseValue	0	0
origCalledPartyRedirectReason	0	114
lastRedirectRedirectReason	0	114
origCalledPartyRedirectOnBehalfOf		15
lastRedirectRedirectOnBehalfOf		15
joinOnBehalfOf		15
destConversationID	0	16777231

2. 40003 calls 40001, and 40001 answers. Shared line 40001' on another phone presses the Barge softkey. All the parties get conferenced together; then, 40001 hangs up.

Note Both CDRs have the same globalCallID_callId, and the conversationID field links back to the CI (call Identifier) of the barged call.

Field Names	Original Call CDR	Barge Call 1 CDR	Final Call 2 CDR
globalCallID_callId	9	9	9
origLegCallIdentifier	16777236	16777238	16777236
destLegCallIdentifier	16777237	16777241	16777238
callingPartyNumber	40003	40001	40003
origCalledPartyNumber	40001	b001501001	40001
finalCalledPartyNumber	40001	b001501001	40001
lastRedirectDn	40001	b001501001	40001
origCause_Value	0	393216	16

Field Names	Original Call CDR	Barge Call 1 CDR	Final Call 2 CDR
dest_CauseValue	16	393216	0
origCalledPartyRedirectReason	0	114	0
lastRedirectRedirectReason	0	114	0
origTerminationOnBehalfOf		15	12
destTerminationOnBehalfOf	12	15	12
lastRedirectRedirectOnBehalfOf		15	
joinOnBehalfOf		15	
destConversationID	0	16777237	0

3. 40003 calls 40001, and 40001 answers. Shared line 40001' on another phone presses the Barge softkey. All the parties get conferenced together; then, 40001' (another shared line and phone) presses the Barge softkey. 40003 hangs up first.

Note All CDRs have the same **globalCallID_callId**, and the conversationID field links back to the CI (call Identifier) of the barged call.

Field Names	Original Call CDR	Barge Call 1 CDR	Final Call 2 CDR
globalCallID_callId	14	14	14
origLegCallIdentifier	16777249	16777251	16777255
destLegCallIdentifier	16777250	16777254	16777258
callingPartyNumber	40003	40001	40001
origCalledPartyNumber	40001	b001501001	b001501001
finalCalledPartyNumber	40001	b001501001	b001501001
lastRedirectDn	40001	b001501001	b001501001
origCause_Value	16	0	0
dest_CauseValue	0	0	0
origCalledPartyRedirectReason	0	114	114
lastRedirectRedirectReason	0	114	114
origTerminationOnBehalfOf	12	15	15
destTerminationOnBehalfOf			

Field Names	Original Call CDR	Barge Call 1 CDR	Final Call 2 CDR
origRedirectRedirectOnBehalfOf		15	15
lastRedirectRedirectOnBehalfOf		15	15
joinOnBehalfOf		15	15
destConversationID	0	16777250	16777251

Related Topics

Cisco Call Detail Records Field Descriptions, on page 127
Cisco Call Detail Records Codes, on page 171
Example Cisco Call Management Records, on page 209

Call Monitoring

The system generates CDRs for the Call Monitoring feature by using existing CDR fields.

The monitoring calls have one-way media. The media fields stay empty for one side of the call for one-way media CDRs.

The **destConversationID** field of the Call Monitoring CDR matches the agent call leg identifier in the CDR of the call that is monitored and links together the Call Monitoring CDR and the CDR of the monitored call.

Call Monitoring Examples

1. The customer (9728134987) calls the agent (30000), and the agent answers. The supervisor (40003) monitors the call. The destConversationID from the monitoring call matches the destLegCallIdentifier of the monitored call.

Field Names	Monitored Call CDR	Monitoring Call CDR
globalCallID_callId	7	10
origLegCallIdentifier	16777230	16777232
destLegCallIdentifier	16777231	16777235
callingPartyNumber	9728134987	40003
originalCalledPartyNumber	30000	b001501001
finalCalledPartyNumber	30000	b001501001
lastRedirectDn	30000	b001501001
origCause_Value	16	0
dest_CauseValue	0	0
origCalledPartyRedirectReason	0	370

Field Names	Monitored Call CDR	Monitoring Call CDR
lastRedirectRedirectOnBehalfOf	0	370
origCalledPartyRedirectOnBehalfOf		28
lastRedirectRedirectOnBehalfOf		28
destConversationID	0	16777231

2. The agent (30000) calls the customer (9728134987), and the customer answers. The supervisor (40003) monitors the call. The destConversationID from the monitoring call matches the origLegCallIdentifier of the monitored call.

Field Names	Monitored Call CDR	Monitoring Call CDR
globalCallID_callId	71	101
origLegCallIdentifier	16777299	16777932
destLegCallIdentifier	16777300	16777235
callingPartyNumber	30000	40003
originalCalledPartyNumber	9728134987	b001501002
finalCalledPartyNumber	9728134987	b001501002
lastRedirectDn	9728134987	b001501002
origCause_Value	16	0
dest_CauseValue	0	0
origCalledPartyRedirectReason	0	370
lastRedirectRedirectOnBehalfOf	0	370
origCalledPartyRedirectOnBehalfOf		28
lastRedirectRedirectOnBehalfOf		28
destConversationID	0	16777299

Related Topics

Cisco Call Detail Records Field Descriptions, on page 127
Cisco Call Detail Records Codes, on page 171
Example Cisco Call Management Records, on page 209

Call Park

Call Park generates two CDRs, one for the original call that gets parked and another for the call that gets picked up or reverted. These CDRs will have the same globalCallID_callId.

Related Topics

Call Park Pickup, on page 37
Call Park Reversion, on page 38
Cisco Call Detail Records Field Descriptions, on page 127
Cisco Call Detail Records Codes, on page 171
Example Cisco Call Management Records, on page 209

Call Park Pickup

When the call is parked, the call gets split. The original call generates a CDR. The **origTerminationOnBehalfOf** and **destTerminationOnBehalfOf** fields get set to Call Park = 3 for this CDR.

When the parked call gets retrieved, the user goes off hook and enters the park code. This call joins with the parked call. Because the user who is picking up the call gets joined with the parked call, the system treats the user as the originator of the call, and the parked user gets treated as the destination. This means that the **callingPartyNumber** field of the call contains the directory number of the user who is picking up the call, and the **originalCalledNumber** and **finalCalledNumber** fields contain the directory number of the parked user. The **lastRedirectDn** field contains the park code that is used to pick up the call. The **lastRedirectRedirectReason** field specifies Call Park Pickup = 8. The **lastRedirectRedirectOnBehalfOf** field should specify Call Park = 3.

Call Park Pickup CDR Example

50003 calls 50002; 50002 presses the Park softkey. 50001 picks up the parked call by dialing the park code (44444).

Field Names	Original Call That Is Parked CDR	Parked Call That Is Picked Up CDR
globalCallID_callId	1	1
origLegCallIdentifier	20863957	20863961
destLegCallIdentifier	20863958	20863957
callingPartyNumber	50003	50001
originalCalledPartyNumber	50002	50003
finalCalledPartyNumber	50002	50003
lastRedirectDn	50002	44444
origCause_Value	393216	0
dest_CauseValue	393216	16
origCalledPartyRedirectReason	0	0

Field Names	Original Call That Is Parked CDR	Parked Call That Is Picked Up CDR
lastRedirectRedirectReason	0	8
origCalledPartyRedirectOnBehalfOf	0	0
lastRedirectRedirectOnBehalfOf	0	3
origTerminationOnBehalfOf	3	0
destTerminationOnBehalfOf	3	12
joinOnBehalfOf	0	3
duration	4	60

Call Park Reversion

When a call is parked and not picked up, the call park reversion timer expires and redirects the call to the called party. In this case, the system generates two CDRs. The first CDR appears the same as the preceding Call Park Pickup scenario, but the second CDR differs slightly. When the Call Pickup Reversion timer expires, the call gets redirected to the called party.

When the call is parked, the call gets split. This action generates a CDR for the original call. The **origTerminationOnBehalfOf** and **destTerminationOnBehalfOf** fields get set to Call Park = 3 for this CDR, the same as the Call Park Pickup scenario.

When the Call Park Reversion timer expires, the call gets redirected to the called party. The **origCalledPartyRedirectOnBehalfOf** and **lastRedirectRedirectOnBehalfOf** fields specify Call Park = 3. The **origCalledPartyRedirectReason** field specifies Call Park = 7, and the **lastRedirectRedirectReason** field specifies Call Park Reversion = 11.

Call Park Reversion CDR Example

- **Call Park Reversion Example** – 50003 calls 50002; 50002 presses the Park softkey. Nobody picks up the parked call; the parked call reverts to 50002, and 50002 answers.

Field Names	Original Call That Is Parked CDR	Reverted Call CDR
globalCallID_callId	2	2
origLegCallIdentifier	20863963	20863963
destLegCallIdentifier	20863964	20863967
callingPartyNumber	50003	50003
originalCalledPartyNumber	50002	50002
finalCalledPartyNumber	50002	50002
lastRedirectDn	50002	50002
origCause_Value	393216	0

Field Names	Original Call That Is Parked CDR	Reverted Call CDR
dest_CauseValue	393216	16
origCalledPartyRedirectReason	0	7
lastRedirectRedirectReason	0	11
origCalledPartyRedirectOnBehalfOf	0	3
lastRedirectRedirectOnBehalfOf	0	3
origTerminationOnBehalfOf	3	3
destTerminationOnBehalfOf	3	12
joinOnBehalfOf	0	3
duration	7	60

Call Pickup

There are two types of call pickup in Cisco Unified Communications Manager: Pickup and Auto Pickup. The CDR records appear slightly different for these two types of call pickup.

Related Topics

Pickup, on page 39
Auto Pickup, on page 40
Cisco Call Detail Records Field Descriptions, on page 127
Cisco Call Detail Records Codes, on page 171
Example Cisco Call Management Records, on page 209

Pickup

Pickup CDR Example

A call comes in from the PSTN to extensions 2000, 2001, and 2002. These extensions reside in the same pickup group. Extension 2002 picks up the call that is ringing on 2001. Extension 2002 answers the call, and the call connects between the PSTN caller and extension 2002.

Field Names	Pickup Call CDR
globalCallID_callId	22
callingPartyNumber	9728131234
originalCalledPartyNumber	2001
finalCalledPartyNumber	2002
lastRedirectDn	2001

Field Names	Pickup Call CDR
origCause_Value	0
dest_CauseValue	16
origTerminationOnBehalfOf	16
destTerminationOnBehalfOf	16
lastRedirectOnBehalfOf	16
lastRedirectReason	5
joinOnBehalfOf	16

Auto Pickup

Auto Pickup acts like call pickup with auto answer. The user does not need to press the last answer softkey. The call automatically connects. Two CDRs get generated for Auto Pickup. These CDRs have the same Call ID.

- The first CDR gets generated for the original call. This CDR has the **origTerminationOnBehalfOf** and **destTerminationOnBehalfOf** fields equal to 16 (Pickup). This value indicates that the call got terminated on behalf of the Pickup feature.
- The second CDR represents the final call after it was picked up. This CDR has the **lastRedirectOnBehalfOf** and the **joinOnBehalfOf** fields set to 16 (Pickup). This value indicates that the call was joined on behalf of the Pickup feature. The **lastRedirectReason** contains the redirect reason of 5 (Pickup).

Auto Pickup CDRs look the same for all types of auto pickup: Auto Pickup, Auto Group Pickup and Auto Other Pickup.

Note When the Service Parameter **Auto Call Pickup Enabled** is set to True for an IP Phone and a Unified Communications Manager receives an incoming call that the IP phone picks up, the prefix digit defined in the Translation Pattern is added to the **callingPartyNumber** in CDR. However, the prefix digit is not added when the Service Parameter **Auto Call Pickup Enabled** is set to False.

Auto Pickup CDR Example

- **Auto Pickup Example** - Call goes from the PSTN to extension 2001; 2001 and 2002 exist in the same pickup group. 2002 picks up the call that rings on 2001; the call automatically connects between the PSTN caller and 2002. They talk for 2 minutes.

Note The prefix digits defined in the Translation Pattern only applies to basic call.

Field Names	Original Call CDR	Pickup CDR
globalCallID_callId	11	11
origLegCallIdentifier	12345	12345
destLegCallIdentifier	12346	12347
callingPartyNumber	9728134987	9728134987
originalCalledPartyNumber	2001	2001
finalCalledPartyNumber	2001	2002
lastRedirectDn	2001	2001
origCause_Value	393216	16
dest_CauseValue	393216	0
origTerminationOnBehalfOf	16	12
destTerminationOnBehalfOf	16	16
lastRedirectRedirectReason	0	5
lastRedirectRedirectOnBehalfOf	0	16
joinOnBehalfOf	0	16
duration	0	120

Call Recording

The system generates CDRs for the Call Recording feature by using existing CDR fields.

The recording calls have one-way media. The media fields stay empty for one side of the call for one-way media CDRs.

The **origConversationID** field of the two Call Recording CDRs matches the agent call leg identifier in the Recording Call CDR and links together the Call Recording CDR and the CDR of the recorded call.

Note If the CDR Log Calls with Zero Duration Flag service parameter is set to true, two additional server call records are created.

Call Recording CDR Examples

1. The customer (9728134987) calls the agent (30000), and the agent answers. The Recorder's DN is 90000. The recording feature creates two recording calls to the recording device, which results in two additional CDRs: one for the agent voice, and another for the customer voice. The origConversationID from the

recording CDRs matches the destLegCallIdentifier of the recorded CDR. In this scenario, the customer hangs up.

Field Names	Recorded Call CDR	Recording Call CDR1	Recording Call CDR2
globalCallID_callId	7	10	11
origLegCallIdentifier	16777110	16777120	16777122
destLegCallIdentifier	16777111	16777121	16777123
callingPartyNumber	9728134987	BIB	BIB
originalCalledPartyNumber	30000	90000	90000
finalCalledPartyNumber	30000	90000	90000
lastRedirectDn	30000	90000	90000
origCause_Value	16	0	0
dest_CauseValue	0	0	0
origCalledPartyRedirectReason	0	354	354
lastRedirectRedirectOnBehalfOf	0	354	354
origCalledPartyRedirectOnBehalfOf		27	27
lastRedirectRedirectOnBehalfOf		27	27
origConversationID	0	16777111	16777111

2. The agent (30000) calls the customer (9728134987), and the customer answers. The Recorder's DN is 90000. The recording feature creates two recording calls to the recording device, which results in two additional CDRs: one for the agent voice, and another for the customer voice. The **origConversationID** field from the recording CDRs will match the **origLegCallIdentifier** field of the recorded CDR. In this scenario, the agent hangs up.

Field Names	Recorded Call CDR	Recording Call CDR 1	Recording Call CDR 2
globalCallID_callId	71	100	110
origLegCallIdentifier	16777113	16777220	16777222
destLegCallIdentifier	16777114	16777221	16777223
callingPartyNumber	30000	BIB	BIB
originalCalledPartyNumber	9728134987	90000	90000
finalCalledPartyNumber	9728134987	90000	90000
lastRedirectDn	9728134987	90000	90000
origCause_Value	16	16	16

Field Names	Recorded Call CDR	Recording Call CDR 1	Recording Call CDR 2
dest_CauseValue	0	0	0
origCalledPartyRedirectReason	0	354	354
lastRedirectRedirectOnBehalfOf	0	354	354
origCalledPartyRedirectOnBehalfOf		27	27
lastRedirectRedirectOnBehalfOf		27	27
origConversationID	0	16777113	16777113

Related Topics

Cisco Call Detail Records Field Descriptions, on page 127
Cisco Call Detail Records Codes, on page 171
Example Cisco Call Management Records, on page 209

Call Secured Status

This field identifies security status of the call. It contains the highest level of security that is reached during a call. For example, if the call is originally unsecured, and later the call changes to secured, the CDR contains 1 for "Secured" even though different portions of the call have different status values. The **callSecuredStatus** field identifies the security status of the call.

Call Secured Status CDR Examples

1. **Encrypted Call** - The system encrypts the call between 20000 and 20001. The parties talk for 5 minutes.

Field Names	CDR
globalCallID_callId	102
origLegCallIdentifier	16777140
destLegCallIdentifier	16777141
callingPartyNumber	20000
origCalledPartyNumber	20001
finalCalledPartyNumber	20001
lastRedirectDn	20001
origCause_Value	0
dest_CauseValue	16
callSecuredStatus	2
duration	300

2. **Authenticated Call** - The call between 20000 and 20001 gets authenticated (not encrypted). The parties talk for 10 minutes.

Field Names	CDR
globalCallID_callId	103
origLegCallIdentifier	16777142
destLegCallIdentifier	16777143
callingPartyNumber	20000
origCalledPartyNumber	20001
finalCalledPartyNumber	20001
lastRedirectDn	20001
origCause_Value	0
dest_CauseValue	16
callSecuredStatus	1
duration	600

Related Topics

Cisco Call Detail Records Field Descriptions, on page 127
Cisco Call Detail Records Codes, on page 171
Example Cisco Call Management Records, on page 209

Calling Party Normalization

This feature provides the support of the international escape code "+" to Unified Communications Manager. This addition enhances the dialing capabilities of dual-mode phones and improves callbacks for companies in different geographical locations.

The **callingPartyNumber**, **originalCalledPartyNumber**, **finalCalledPartyNumber**, **lastRedirectDN fields**, and the new fields, **outpulsedCallingPartyNumber** and **outpulsedCalledPartyNumber,** may now contain a "+" in the CDR. The device reports the Calling Party Number that it outpulsed back to Call Control only if calling party normalization/localization takes place. If calling party normalization/localization occurs, the action gets recorded in the CDR in the new field **outpulsedCallingPartyNumber**.

Calling Party Normalization CDR Examples

1. A call gets placed from a Dallas PSTN to an enterprise phone. The 7-digit calling number comprises 500 1212; the Dallas area code displays 972. The calling party transformation contains +1972. The **callingPartyNumber** field in the CDR contains +1 972 500 1212 (global format). The new field **outpulsedCallingPartyNumber** contains the localized number 500 1212.

Field Names	Values
globalCallID_callId	1
origLegCallIdentifier	100
destLegCallIdentifier	101
callingPartyNumber	+19725001212
outpulsedCallingPartyNumber	5001212
duration	60

1. A call gets placed from an enterprise phone to a Dallas PSTN. The extension of the enterprise phone comprises 12345; the fully qualified number comprises 9725002345. Calling party transformation checks the external phone number mask feature. The **callingPartyNumber** field in the CDR contains +1 972 500 2345 (global format). The new field **outpulsedCallingPartyNumber** contains the localized number 9725002345.

Field Names	Values
globalCallID_callId	2
origLegCallIdentifier	102
destLegCallIdentifier	103
callingPartyNumber	+19725002345
outpulsedCallingPartyNumber	9725002345
duration	60

Related Topics

Cisco Call Detail Records Field Descriptions, on page 127
Cisco Call Detail Records Codes, on page 171
Example Cisco Call Management Records, on page 209

Calls with Busy or Bad Destinations

The system logs all these calls as normal calls, and all relevant fields contain data. The Calling or Called Party Cause fields contain a cause code that indicates why the call does not connect, and the Called Party IP and Date/Time Connect fields remain blank. The system logs all unsuccessful calls, even if zero duration calls are not being logged (CdrLogCallsWithZeroDurationFlag set at **True** or **False**, a duration of zero, and a DateTimeConnect value of zero).

Examples of Unsuccessful Calls CDRs

1. Call goes to PSTN number, but party already is engaged (cause 17 = user busy)

Field Names	CDR
globalCallID_callId	3
origLegCallIdentifier	300
destLegCallIdentifier	301
callingPartyNumber	2001
originalCalledPartyNumber	9728134987
origCause_Value	0
dest_CauseValue	17
duration	0

2. Call goes to PSTN number, but number does not exist (cause 1 = number unavailable)

Field Names	CDR
globalCallID_callId	4
origLegCallIdentifier	302
destLegCallIdentifier	303
callingPartyNumber	2001
originalCalledPartyNumber	9728134987
origCause_Value	1
dest_CauseValue	0
duration	0

3. Call to PSTN fails because PSTN trunks are out of order (cause 38 = Network Out Of Order).

Field Names	CDR
globalCallID_callId	5
origLegCallIdentifier	304
destLegCallIdentifier	305
callingPartyNumber	2001
originalCalledPartyNumber	9728134987
origCause_Value	0
dest_CauseValue	38

Field Names	CDR
duration	0

Related Topics

Cisco Call Detail Records Field Descriptions, on page 127
Cisco Call Detail Records Codes, on page 171
Example Cisco Call Management Records, on page 209

cBarge

The cBarge feature acts very similar to the conference feature. When a shared line uses the cBarge feature, the **origCalledPartyNumber, finalCalledPartyNumber** and **lastRedirectDn** represent the conference bridge number 'b00. . . '. The redirect and join **OnBehalfOf** fields have a value of Conference = 4, and the **redirect reason** fields specify Conference = 98.

cBarge CDR Example

40003 calls 40001, and 40001 answers; 40001' (shared line) on another phone presses the cBarge button.

Field Names	Orig Call CDR	cBarge Call CDR 1	cBarge Call CDR 2	cBarge Call CDR 3	Final Call CDR
globalCallID_callId	49	49	49	49	49
origLegCallIdentifier	1677346	1677348	1677347	1677346	1677347
destLegCallIdentifier	1677347	1677353	1677351	1677352	1677346
callingPartyNumber	40003	40001	40001	40003	40001
originalCalledPartyNumber	40001	b0029901001	b0029901001	b0029901001	40003
finalCalledPartyNumber	40001	b0029901001	b0029901001	b0029901001	40003
lastRedirectDn	40001	b0029901001	40001	40001	b0029901001
origCause_Value	393216	16	393216	393216	16
dest_CauseValue	393216	0	393216	393216	0
origCalledPartyRedirectReason	0	98	98	98	0
lastRedirectRedirectReason	0	98	98	98	98
destTerminationOnBehalfOf	4		4	4	4
origCalledRedirectOnBehalfOf		4	4	4	
lastRedirectRedirectOnBehalfOf		4	4	4	4
joinOnBehalfOf		4	4	4	4

Field Names	Orig Call CDR	cBarge Call CDR 1	cBarge Call CDR 2	cBarge Call CDR 3	Final Call CDR
Conversation ID	0	16777220	16777220	16777220	1
duration	60	360		360	360

Comment	
Orig Call CDR	
cBarge Call CDR 1	ConfControllerDn=40003;ConfControlerDeviceName=SEP0003E333FEBD
cBarge Call CDR 2	ConfControllerDn=40003;ConfControlerDeviceName=SEP0003E333FEBD
cBarge Call CDR 3	ConfControllerDn=40003;ConfControlerDeviceName=SEP0003E333FEBD
Final Call CDR	ConfControllerDn=40003;ConfControlerDeviceName=SEP0003E333FEBD

Related Topics

Cisco Call Detail Records Field Descriptions, on page 127
Cisco Call Detail Records Codes, on page 171
Example Cisco Call Management Records, on page 209

Client Matter Code (CMC)

When the CMC feature gets invoked, the system writes the client matter code into the CDR. The **clientMatterCode** field contains the client matter code that the caller enters.

CMC CDR Example

10000 calls 2142364624; the user gets prompted for a client matter code and enters 11111. The caller answers the call and talks for 10 minutes.

Field Names	Values
globalCallID_callId	101
origLegCallIdentifier	16777130
destLegCallIdentifier	16777131
callingPartyNumber	10000
origCalledPartyNumber	2142364624
finalCalledPartyNumber	2142364624
lastRedirectDn	2142364624
origCause_Value	0
dest_CauseValue	16

Field Names	Values
clientMatterCode	11111
duration	600

Related Topics

Cisco Call Detail Records Field Descriptions, on page 127
Cisco Call Detail Records Codes, on page 171
Example Cisco Call Management Records, on page 209

Conference Calls

Multiple records get logged for calls that are part of a conference. The number of CDR records that get generated depends on the number of parties in the conference. One CDR exists for each party in the conference; one CDR for the original placed call, one CDR for each setup call that gets used to join other parties to the conference, and one CDR for the last two parties that get connected in the conference. For a three-party, ad hoc conference, six CDRs exist: one CDR for the original call, three CDRs for the parties that get connected to the conference, one CDR for each setup call, and one CDR for the final two parties in the conference. You can associate the setup calls with the correct call leg in the conference by examining the calling leg ID and called leg ID.

The conference bridge device represents special significance to the Unified Communications Manager, and calls to the conference bridge appear as calls to the conference bridge device. A special number in the form "b0019901001" shows the conference bridge port. Records show all calls into the conference bridge, regardless of the actual direction; however, by examining the setup call CDRs, you can determine the original direction of each call.

You can find the conference controller information in the comment field of the CDR. The format of this information follows:

Comment field = "ConfControllerDn=1000;ConfControllerDeviceName=SEP0003"

- The conference controller DN + conference controller device name uniquely identify the conference controller. The system needs the device name in the case of shared lines.
- If the call is involved in multiple conference calls, the comment field contains multiple conference controller information. This situation can occur when the conference goes down to two parties, and one of these parties starts another conference. If this is the case, the **last** conference controller information in the comment field identifies the conference controller.

The call legs that are connected to the conference include the following information fields:

- The **finalCalledPartyNumber** field contains the conference bridge number "b0019901001."
- The **origCalledPtyRedirectOnBehalfOf** field gets set to Conference = 4.
    - The **lastRedirectRedirectOnBehalfOf** field gets set to Conference = 4.
    - The **joinOnBehalfOf** field gets set to (Conference = 4).
    - The **comment** field identifies the conference controller.

- The **destConversationID** field remains the same for all members in the conference. You can use this field to identify members of a conference call.

The original placed call and all setup calls that were used to join parties to the conference have the following characteristics:

- The **origCallTerminationOnBehalfOf** field gets set to Conference = 4.
- The **destCallTerminationOnBehalfOf** field gets set to Conference = 4.

Conference Call CDR Example

- Call goes from 2001 to 2309.
- 2309 answers and talks for 60 seconds.
- 2001 presses the conference softkey and dials 3071111.
- 307111 answers and talks for 20 seconds; then, 2001 presses the conference softkey to complete the conference.
- The three members of the conference talk for 360 seconds.

3071111 hangs up and leaves 2001 and 2309 in the conference. Because only two participants are left in the conference, the conference features joins these two directly together, and they talk for another 55 seconds.

Note Each conference call leg gets shown as placing a call into the conference bridge. The system shows the call as a call into the bridge, regardless of the actual direction of the call.

Field Names	Orig Call CDR	Setup Call CDR	Conference CDR 1	Conference CDR 2	Conference CDR 3	Final CDR
globalCallID_callId	1	2	1	1	1	1
origLegCallIdentifier	101	105	101	102	106	101
destLegCallIdentifier	102	106	115	116	117	102
callingPartyNumber	2001	2001	2001	2309	3071111	2001
originalCalledPartyNumber	2309	3071111	b0029901001	b0029901001	b0029901001	2309
finalCalledPartyNumber	2301	3071111	b0029901001	b0029901001	b0029901001	2309
lastRedirectDn	2001	3071111	b0029901001	b0029901001	b0029901001	b0029901001
origCause_Value	393216	0	16	393216	393216	16
dest_CauseValue	393216	0	393216	393216	393216	0
origCalledPartyRedirectReason	0	0	0	0	0	0
lastRedirectRedirectReason	0	0	0	0	0	98

Field Names	Orig Call CDR	Setup Call CDR	Conference CDR 1	Conference CDR 2	Conference CDR 3	Final CDR
origTerminationOnBehalfOf	4	4	12	12	4	12
destTerminationOnBehalfOf	4	4	0	0	4	4
origCalledRedirectOnBehalfOf	0	0	4	4	4	0
lastRedirectRedirectOnBehalfOf	0	0	4	4	4	4
joinOnBehalfOf	0	0	4	4	4	4
Conversation ID	0	0	1	1	1	0
duration	60	20	360	360	360	55

Comment	
Orig Call CDR	
Setup Call CDR	ConfControllerDn=2001;ConfControlerDeviceName=SEP0003E333FEBD
Conference CDR 1	ConfControllerDn=2001;ConfControlerDeviceName=SEP0003E333FEBD
Conference CDR 2	ConfControllerDn=2001;ConfControlerDeviceName=SEP0003E333FEBD
Conference CDR 3	ConfControllerDn=2001;ConfControlerDeviceName=SEP0003E333FEBD
Final CDR	

Related Topics

Cisco Call Detail Records Field Descriptions, on page 127
Cisco Call Detail Records Codes, on page 171
Example Cisco Call Management Records, on page 209

Operational Factors

Three major operational factors exist for conference call CDRs:

1. When a conference decreases to two parties, the two parties connect directly and release the conference resource. This change generates an additional CDR for the call between the last two parties in the conference call.

   For example, if four people connect in a conference call (Amy, Dustin, Spencer, Ethan), when Ethan hangs up, three people remain in the conference call that is connected to the conference bridge (Amy, Dustin, Spencer). When Spencer hangs up, only two people remain in the conference call (Amy and Dustin). The system joins Amy and Dustin directly, and, the conference resource gets released. Directly joining Amy and Dustin creates an additional CDR between the last two parties in the conference.

2. The system adds the conference controller information to the comment field in the CDR. This information identifies the conference controller. No need now exists to examine the consultation call to determine who is the conference controller. The following example shows this information:

Comment field = “ConfControllerDn=1000;ConfControllerDeviceName=SEP0003E333FEBD”

- The conference controller DN + conference controller device name uniquely identify the conference controller. A need for the device name exists in the case of shared lines.
- If the call is involved in multiple conference calls, the comment field contains multiple conference controller information. This situation may occur when the conference goes down to two parties, and one of these parties starts another conference. If this is the case, the last conference controller information in the comment field identifies the conference controller.

3. The party that added the participant, known as the requestor party, appears in the CDR comment field. The tags for the requestor information include ConfRequestorDn and ConfRequestorDeviceName. The party that requested to remove a participant, known as the drop requestor, appears in the CDR comment field. The tags for the drop requestor information include DropConfRequestorDn and DropConRequestorDeviceName.

Calls that are part of a conference have multiple records that are logged for them. The number of CDRs that get generated depends on the number of parties in the conference. One CDR exists for each party in the conference, one CDR for the original placed call, and one CDR for each setup call that is used to join other parties to the conference. Therefore, for a three-party ad hoc conference, six CDRs exist:

- One CDR for the original call.
- Three CDRs for the parties that are connected to the conference.
- One CDR for each setup call.
- One CDR for the final two parties in the conference.

You can associate the setup calls with the correct call leg in the conference by examining the calling leg ID and the called leg ID.

The conference bridge device holds special significance to the Unified Communications Manager. Calls to the conference bridge appear as calls to the conference bridge device. A special number in the form “b0019901001” shows the conference bridge port. All calls get shown “into” the conference bridge, regardless of the actual direction. You can determine the original direction of each call by examining the setup call CDRs.

The call legs that are connected to the conference have the following values for these fields:

- **finalCalledPartyNumber**—Represents a conference bridge “b0019901001”.
- **origCalledPartyRedirectOnBehalfOf**—Set to Conference (4).
- **lastRedirectRedirectOnBehalfOf**—Set to Conference (4).
- **joinOnBehalfOf**—Set to Conference (4).
- **comment**—Identifies the conference controller.

The original placed call and all setup calls that get used to join parties to the conference have the following values for the fields:

- **origCallTerminationOnBehalfOf**—Set to Conference (4).
- **destCallTerminationOnBehalfOf**—Set to Conference (4).

Conference Drop Any Party

The Conference Drop Any Party feature terminates calls that look the same as other calls except for a new cause code. The cause code identifies the calls that this feature terminates.

Conference Drop Any Party CDR Example

The following table contains an example CDR for a call that connects to a conference and gets dropped by this feature.

Calling Party	Calling Partition	Original Called Party	Orig Cause	Original Called Partition	Called Leg	Dest Cause	Final Called Party	Final Called Partition	Last Redirect Party
2001	ACNTS	2309	0	MKTG	102	16	2309	MKTG	2001
2001	ACNTS	2309	16	MKTG	115	0	b0029901001		b0029901001
2309	ACNTS	b0029901001	0		116	128	b0029901001		b0029901001
3071111	PSTN	b0029901001	16		117	0	b0029901001		b0029901001
2001	ACNTS	2309	16	PSTN	106	0	3071111	PSTN	30711111

Orig Conversation ID	OrigCall Termination OnBehalfOf	DestCall Termination OnBehalfOf	OriginalCalled Pty Redirect OnBehalfOf	LastRedirect Redirect OnBehalfOf	Join OnBehalfOf	Duration
0	4	4	0	0	0	60
1	12	0	4	4	4	360
1	13	0	4	4	4	200
1	4	4	4	4	4	360
0	4	4	0	0	0	20

Related Topics

Cisco Call Detail Records Field Descriptions, on page 127
Cisco Call Detail Records Codes, on page 171
Example Cisco Call Management Records, on page 209

Original Calling Party on Transfer

This feature changes the calling party number for a consultation call of a Cisco Unity or Cisco Unity Connection-initiated call transfer. The CDR of the consultation call shows that the original caller calls the transfer destination, not that the Cisco Unity or Cisco Unity Connection port calls the transfer destination.

You must configure this feature in the service parameters in Cisco Unified Communications Manager. See additional information in the "Configuring CDR Service Parameters" section of the *CDR Analysis and Reporting Administration Guide*.

Original Calling Party on Transfer CDR Example

4001 calls 4002. 4002 transfers the call to 4003. The system generates three CDRs:

- The call between the original parties (4001 to 4002).
- The consultation call between the transferring party (4002) to the final transfer destination (4003).
- The call from the transferred party (4001) to the transfer destination (4003).

Call	CallingPartyNumber	originalCalledPartyNumber
1	4001	4002
2	4002	4003
3	4001	4003

Note No **originalCallingParty** field exists in the CDR.

DTMF Method

These fields identify the Dual Tone Multi-Frequency (DTMF) method that gets used for the call.

DTMF CDR Examples

1. **No Preference Example -** The DTMF method that gets used during this call represents No Preference/Best Effort. This call connects for 1 minute.

Field Names	CDR
globalCallID_callId	200
origLegCallIdentifier	16777500
destLegCallIdentifier	16777501
callingPartyNumber	20000
origCalledPartyNumber	20001
finalCalledPartyNumber	20001
lastRedirectDn	20001
origCause_Value	0

Field Names	CDR
dest_CauseValue	16
origDTMFMethod	0
destDTMFMethod	0
duration	60

1. **Preferred OOB Example** - The DTMF method that is used during this call represents OOB Preferred. This call remains connected for 1 minute.

Field Names	CDR
globalCallID_callId	201
origLegCallIdentifier	16777502
destLegCallIdentifier	16777503
callingPartyNumber	20000
origCalledPartyNumber	20001
finalCalledPartyNumber	20001
lastRedirectDn	20001
origCause_Value	0
dest_CauseValue	16
origDTMFMethod	1
destDTMFMethod	1
duration	60

Related Topics

Cisco Call Detail Records Field Descriptions, on page 127
Cisco Call Detail Records Codes, on page 171
Example Cisco Call Management Records, on page 209

End-to-End Call Trace

The End-to-End Call Trace feature facilitates tracing calls that traverse multiple Cisco voice products, such as Unified CM, Cisco IOS Gateways, and other products.

End-to-End Call Trace Example

1. H323 - Calling party 1003 calls 1004 via H.323 trunk.

FieldNames	Values
cdrRecordType	1
globalCallID_callManagerId	1
globalCallID_callId	32009
origLegCallIdentifier	19654113
dateTimeOrigination	1221263718
origNodeId	1
origSpan	0
origIpAddr	1897990154
callingPartyNumber	1004
origCause_value	16
origPrecedenceLevel	4
origMediaTransportAddress_IP	1897990154
origMediaTransportAddress_Port	19824
origMediaCap_payloadCapability	4
origMediaCap_maxFramesPerPacket	20
destLegIdentifier	19654114
destNodeId	1
destSpan	19654114
destIpAddr	424630538
originalCalledPartyNumber	1003
finalCalledPartyNumber	1003
destCause_value	0
destPrecedenceLevel	4
destMediaTransportAddress_IP	-1759442934
destMediaTransportAddress_Port	27508
destMediaCap_payloadCapability	4
destMediaCap_maxFramesPerPacket	20
dateTimeConnect	1221263720

FieldNames	Values
dateTimeDisconnect	1221263721
lastRedirectDn	1003
Pkid	c8868f84-0f4e-452c-a814-bf97a7fe69fc
Duration	1
origDeviceName	SEP003094C2B08C
destDeviceName	self-loop
origCallTerminationOnBehalfOf	12
destCallTerminationOnBehalfOf	0
origDTMFMethod	3
destDTMFMethod	4
origMediaCap_Bandwidth	64
destMediaCap_Bandwidth	64
origIpv4v6Addr	10.8.33.113
destIpv4v6Addr	10.8.33.151
IncomingProtocolID	0
IncomingProtocolCallRef	
OutgoingProtocolID	2
OutgoingProtocolCallRef	0053C43F6701B18C030004010A082171

2. **Q931** - 1004 calls 1003 via Q931.

FieldNames	Values
cdrRecordType	1
globalCallID_callManagerId	1
globalCallID_callId	32008
origLegCallIdentifier	19654111
dateTimeOrigination	1221263350
origNodeId	1
origSpan	2

FieldNames	Values
origIpAddr	122640650
callingPartyNumber	1004
origCause_value	0
origPrecedenceLevel	4
origMediaTransportAddress_IP	122640650
origMediaTransportAddress_Port	17218
origMediaCap_payloadCapability	4
origMediaCap_maxFramesPerPacket	20
destLegIdentifier	19654112
destNodeId	1
destSpan	0
destIpAddr	-1759442934
originalCalledPartyNumber	1003
finalCalledPartyNumber	1003
destCause_value	16
destPrecedenceLevel	4
destMediaTransportAddress_IP	-1759442934
destMediaTransportAddress_Port	23350
destMediaCap_payloadCapability	4
destMediaCap_maxFramesPerPacket	20
dateTimeConnect	1221263351
dateTimeDisconnect	1221263352
lastRedirectDn	1003
Pkid	b576bd8d-9703-4f66-ae45-64ae5c04738e
Duration	1
origDeviceName	BRI/S1/SU0/P1@nw052b-3640.cisco.com
destDeviceName	SEP003094C2D263
origCallTerminationOnBehalfOf	0

FieldNames	Values
destCallTerminationOnBehalfOf	12
origDTMFMethod	1
destDTMFMethod	3
origMediaCap_Bandwidth	64
destMediaCap_Bandwidth	64
origIpv4v6Addr	10.89.79.7
destIpv4v6Addr	10.8.33.151
IncomingProtocolID	4
IncomingProtocolCallRef	01-1004-1003
OutgoingProtocolID	0
OutgoingProtocolCallRef	

Related Topics

Cisco Call Detail Records Field Descriptions, on page 127
Cisco Call Detail Records Codes, on page 171
Example Cisco Call Management Records, on page 209

Forced Authorization Code (FAC)

When the FAC feature gets invoked, the system writes the authorization description and level into the CDR. For security reasons, the actual authorization code does not get written to the CDR.

- The **authCodeDescription** field contains the description of the authorization code.
- The **authorizationLevel** field contains the level of authorization that is associated with the authorization code.

FAC CDR Example 1

45000 calls 9728134987; the system prompts the user for an authorization code and enters 12345. FAC code 12345 gets configured as level 1 and name Legal1. The caller answers the call and talks for 2 minutes.

Field Names	Values
globalCallID_callId	100
origLegCallIdentifier	16777123
destLegCallIdentifier	16777124

Field Names	Values
callingPartyNumber	45000
origCalledPartyNumber	9728134987
finalCalledPartyNumber	9728134987
lastRedirectDn	9728134987
origCause_Value	0
dest_CauseValue	16
authCodeDescription	Legal1
authorizationLevel	1
duration	120

Related Topics

Cisco Call Detail Records Field Descriptions, on page 127
Cisco Call Detail Records Codes, on page 171
Example Cisco Call Management Records, on page 209

Forwarded or Redirected Calls

Forwarded calls generate a single CDR and show the Calling Party, Original Called Number, Last Redirecting Number, Final Called Number, and the associated partitions. If the call gets forwarded more than twice, the intermediate forwarding parties do not populate in the CDR.

Call forwarding can occur on several conditions (always, busy, and no answer). The condition under which the call gets forwarded does not populate in the CDR.

The CDRs for forwarded calls match those for normal calls, except for the originalCalledPartyNumber field and the originalCalledPartyNumberPartition field. These fields contain the directory number and partition for the destination that was originally dialed by the originator of the call. If the call gets forwarded, the finalCalledPartyNumber and finalCalledPartyNumberPartition fields differ and contain the directory number and partition of the final destination of the call.

Also, when a call gets forwarded, the lastRedirectDn and lastRedirectDnPartition fields contain the directory number and partition of the last phone that forwarded or redirected the call.

Call Forwarding uses the redirect call primitive to forward the call. Features that use the redirect call primitive have similar CDRs. Some of the important CDR fields for forwarded calls follow:

- The **originalCalledPartyNumber** contains the number of the original called party.
- The **finalCalledPartyNumber** represents the number that answered the call.
- The **lastRedirectDn** field specifies the number that performed the last redirect.
- The **origCalledPartyRedirectReason** represents the reason that the call was redirected the first time. For call forwarding, this field can contain **Call Forward Busy=1, Call Forward No Answer=2, Call Forward All=15**.

- The **lastRedirectRedirectReason** specifies the reason that the call was redirected the last time. For call forwarding, this field can contain **Call Forward Busy=1, Call Forward No Answer=2, Call Forward All=15**.
- The **origCalledPartyRedirectOnBehalfOf** field identifies which feature redirects the call for the first redirect. For call forwarding, this field specifies 5 (Call Forward).
- The **lastRedirectRedirectOnBehalfOf** field identifies which feature redirects the call for the last redirect. For call forwarding, this field specifies 5 (Call Forward).

Forwarded Calls CDR Examples

1. **CFA** - Call comes in from the PSTN to extension 2001; the call gets forwarded (CFA) to 2309, where the call is answered, and talk occurs for 2 minutes.

Field Names	CDR
globalCallID_callId	12345
origLegCallIdentifier	100
destLegCallIdentifier	102
callingPartyNumber	9728134987
originalCalledPartyNumber	2001
finalCalledPartyNumber	2309
lastRedirectDn	2001
origCause_Value	0
dest_CauseValue	16
origCalledPartyRedirectReason	15
lastRedirectRedirectReason	15
origCalledPartyRedirectOnBehalfOf	5
lastRedirectRedirectOnBehalfOf	5
duration	120

2. **Multiple Hop CFA & CFNA** - Call comes in from the PSTN to extension 1000; the call gets forwarded (CFA) to 2000; then, the call gets forwarded (CFNA) to the voice-messaging system (6000) where the caller leaves a message.

Field Names	CDR
globalCallID_callId	12346
origLegCallIdentifier	102

Field Names	CDR
destLegCallIdentifier	105
callingPartyNumber	9728134987
originalCalledPartyNumber	1000
finalCalledPartyNumber	6000
lastRedirectDn	2000
origCause_Value	0
dest_CauseValue	16
origCalledPartyRedirectReason	15
lastRedirectRedirectReason	2
origCalledPartyRedirectOnBehalfOf	5
lastRedirectRedirectOnBehalfOf	5
duration	15

3. **Multiple Hop CFNA & CFB** - Call comes in from the PSTN to extension 4444; the call gets forwarded (CFNA) to 5555; then, it gets forwarded (CFB) to 6666 where the call is answered, and they talk for 30 seconds.

Field Names	CDR
globalCallID_callId	12347
origLegCallIdentifier	106
destLegCallIdentifier	108
callingPartyNumber	9728134987
originalCalledPartyNumber	4444
finalCalledPartyNumber	6666
lastRedirectDn	5555
origCause_Value	16
dest_CauseValue	0
origCalledPartyRedirectReason	2
lastRedirectRedirectReason	1
origCalledPartyRedirectOnBehalfOf	5

Field Names	CDR
lastRedirectRedirectOnBehalfOf	5
duration	30

Related Topics

Cisco Call Detail Records Field Descriptions, on page 127
Cisco Call Detail Records Codes, on page 171
Example Cisco Call Management Records, on page 209

Hunt List Support

Hunt List Examples

1. **Answered Calls** - In this example, calls go to a hunt list and a member of the hunt list answers the call.
   - Cisco Unified IP Phones 3001, 3002, 3003 and 3004 are part of the hunt list. The display names for the phones are 3001-Name, 3002-Name, 3003-Name and 3004-Name, respectively.
   - Hunt Pilot 2000 is associated with a hunt list. Hunt pilot 2000 is configured with display name as 2000-Name.
   - Phone 1000 calls hunt pilot 2000; call is offered at 3001 and answered.

When the **Show Line Group Member DN in finalCalledPartyNumber CDR Field** service parameter is set to True, the following values appear in the CDR.

Field Names	CDR
callingPartyNumber	1000
callingPartyNumberPartition	
originalCalledPartyNumber	2000
originalCalledPartyNumberPartition	
finalCalledPartyNumber	3001
finalCalledPartyNumberPartition	
origDeviceName	Phone 1000
destDeviceName	Phone 3001
huntPilotDN	2000
huntPilotPartition	

When the **Show Line Group Member DN in finalCalledPartyNumber CDR Field** service parameter is set to False, the following values in the table display in the CDR.

Field Names	CDR
callingPartyNumber	1000
callingPartyNumberPartition	
originalCalledPartyNumber	2000
originalCalledPartyNumberPartition	
finalCalledPartyNumber	2000
finalCalledPartyNumberPartition	
origDeviceName	Phone 1000
destDeviceName	Phone 3001
huntPilotDN	2000
huntPilotPartition	

1. **Abandoned or Failed Calls** - In this example, calls go to a hunt list and a member of the hunt list abandons or fails the call.
   - Cisco Unified IP Phones 3001, 3002, 3003 and 3004 are part of the hunt list.
   - Hunt Pilot 2000 is associated with a hunt list.
   - Phone 1000 calls hunt pilot 2000; call is offered at 3001 and abandoned. When the **Show Line Group Member DN in finalCalledPartyNumber CDR field** service parameter is set to True, the following values from the table display in the CDR:

Field Names	CDR
callingPartyNumber	1000
callingPartyNumberPartition	
originalCalledPartyNumber	2000
originalCalledPartyNumberPartition	
finalCalledPartyNumber	**2000**
finalCalledPartyNumberPartition	
origDeviceName	Phone 1000
destDeviceName	Phone 3001
huntPilotDN	
huntPilotPartition	
calledPartyPatternUsage	7

Note If the call is not answered by any of the hunt group members, the finalCalledPartyNumber field shows the hunt pilot DN. The number shows a line group member DN only when one of the line group member answers the call.

Because the call does not get answered, the huntPilotDN is not available in the CDR. The **PatternUsage** (7 = PATTERN_HUNT_PILOT) field gets set to 7 to indicate that the call was made to a hunt pilot. When the service parameter is enabled, the **finalCalledPartyNumber** field denotes the member hunt DN and the **originalCalledPartyNumber** field denotes the huntPilot DN.

When the **Show Line Group Member DN in the finalCalledPartyNumber CDR field** service parameter is set to False, the following values in the table display in the CDR:

Field Names	CDR
callingPartyNumber	1000
callingPartyNumberPartition	
originalCalledPartyNumber	2000
originalCalledPartyNumberPartition	
finalCalledPartyNumber	**2000**
finalCalledPartyNumberPartition	
origDeviceName	Phone 1000
destDeviceName	Phone 3001
huntPilotDN	
huntPilotPartition	
calledPartyPatternUsage	7

Because the call is not answered, the huntPilotDN is not available in the CDR. The **PatternUsage** (7 = PATTERN_HUNT_PILOT) field gets set to 7 to indicate that the call was made to a hunt pilot. When the service parameter is not enabled, the **finalCalledPartyNumber** field denotes the member hunt DN.

H.239

Unified Communications Manager supports H.239. This feature defines the procedures for use of up to two video channels in H.320-based systems and for labeling individual channels with a role of “presentation” or “live.” This procedure indicates the requirements for processing the channel and the role of the channel content in the call. Role labels apply to both H.320 and H.245 signaling-based systems.

Several new CDR fields support a second video channel for both the origination and destination devices. This CDR provides an example of these new fields.

H.239 CDR Example

When A and B declare H.239 capability in Terminal Capability Set (TCS) and one, or both, of the endpoints initiates the receiving channel to have an extended video channel in an H.239 mechanism for presentation or video feed, the new CDR fields display in the CDR in addition to the existing fields of a video call.

Calling party 51234 calls the called party 57890. Let 103 represent H.264, 187962284 represents 172.19.52.11, 288625580 represents 172.19.52.17, and 352 represents 352K.

Field Names	CDR
globalCallID_callId	121
origLegCallIdentifier	101
destLegCallIdentifier	102
callingPartyNumber	51234
originalCalledPartyNumber	57890
finalCalledPartyNumber	57890
lastRedirectDn	57890
origCause_Value	0
destCause_Value	16
origVideoCap_Codec	103
origVideoCap_Bandwidth	352
origVideoCap_Resolution	0
origVideoTransportAddress_IP	187962284
origVideoTransportAddress_Port	2406
destVideoCap_Codec	103
destVideoCap_Bandwidth	352
destVideoCap_Resolution	0
destVideoTransportAddress_IP	288625580
destVideoTransportAddress_Port	2328
origVideoCap_Codec_Channel2	103
origVideoCap_Bandwidth_Channel2	352
origVideoCap_Resolution_Channel2	0
origVideoTransportAddress_IP_Channel2	187962284

Field Names	CDR
origVideoTransportAddress_Port_Channel2	2410
origVideoChannel_Role_Channel2	0
destVideoCap_Codec_Channel2	103
destVideoCap_Bandwidth_Channel2	352
destVideoCap_Resolution_Channel2	0
destVideoTransportAddress_IP_Channel2	288625580
destVideoTransportAddress_Port_Channel2	2330
destVideoChannel_Role_Channel2	0

Related Topics

CDR Field Descriptions, on page 127
Cisco Call Detail Records Field Descriptions, on page 127
Cisco Call Detail Records Codes, on page 171
Example Cisco Call Management Records, on page 209

iLBC Calls

Internet Low Bit Rate Codec (iLBC) enables graceful speech quality degradation in a lossy network where frames get lost. For iLBC calls, the codec specifies Media_Payload_ILBC = 86.

The system adds an audio bandwidth field to the CDR for iLBC calls.

Field Names	Definitions
origMediaCap_bandwidth	This integer field contains the audio bandwidth.
destMediaCap_bandwidth	This integer field contains the audio bandwidth.

The system populates the bandwidth fields based on the following table:

Codec	Bandwidth
G711Alaw64k	64
G711Alaw56k	56
G711mu-law64k	64
G711mu-law56k	56
G722 64k	64
G722 56k	56

Codec	Bandwidth
G722 48k	48
G7231	7
G728	16
G729	8
G729AnnexA	8
Is11172AudioCap	0
Is13818AudioCap	0
G729AnnexB	8
G729AnnexAwAnnexB	8
GSM Full Rate	13
GSM Half Rate	7
GSM Enhanced Full Rate	13
Wideband 256K	256
Data 64k	64
Data 56k	56
G7221 32K	32
G7221 24K	24
AAC-LD (mpeg4-generic)	256
AAC-LD (MP4A-LATM) 128K	128
AAC-LD (MP4A-LATM) 64K	64
AAC-LD (MP4A-LATM) 56K	56
AAC-LD (MP4A-LATM) 48K	48
AAC-LD (MP4A-LATM) 32K	32
AAC-LD (MP4A-LATM) 24K	24
GSM	13
iLBC	15 or 13
iSAC	32
XV150 MR 729A	8

Codec	Bandwidth
NSE VBD 729A	8

iLBC Call CDR Example

This example applies to a call with iLBC codec.

Field Names	iLBC CDR
globalCallID_callId	121
origLegCallIdentifier	101
destLegCallIdentifier	102
callingPartyNumber	51234
originalCalledPartyNumber	57890
finalCalledPartyNumber	57890
lastRedirectDn	57890
origCause_Value	0
dest_CauseValue	16
origMediaCap_payloadCapability	86
origMediaCap_Bandwidth	15
destMediaCap_payloadCapability	86
destMediaCap_Bandwidth	15

Related Topics

Cisco Call Detail Records Field Descriptions, on page 127
Cisco Call Detail Records Codes, on page 171
Example Cisco Call Management Records, on page 209

Intercompany Media Engine

Successful IME Calls

A call is made to PSTN. The gateway has it in the learned IME route and the call is extended to IME trunk. Call is successfully routed out through IME trunk.

Field Names	CDR
globalCallID_callId	3

Field Names	CDR
origLegCallIdentifier	300
destLegCallIdentifier	301
callingPartyNumber	2001
originalCalledPartyNumber	9728134987
lastRedirectRedirectOnBehalfOf	30
lastRedirectRedirectReason	0
duration	10

Failed IME Calls Due to IME Trunk Rejection

A call is made to PSTN. The gateway has it in the learned IME route and the call is extended to IME trunk. The IME trunk rejects the call, and the call treatment does not cause the call to be redirected to the PSTN, so the call gets rejected. Depending on the reason of IME trunk reject, different lastRedirectRedirectReason can be reported.

Field Names	CDR
globalCallID_callId	3
origLegCallIdentifier	300
destLegCallIdentifier	301
callingPartyNumber	2001
originalCalledPartyNumber	9728134987
origTerminationOnBehalfOf	30
lastRedirectRedirectOnBehalfOf	30
lastRedirectRedirectReason	496 OR 512 OR 528 OR 544 OR 560 OR 576 OR 592 OR 608 OR 624 OR 640 OR 656 OR 672 OR 688 OR 704
origCause_Value	31
duration	0

IME Calls Redirected to PSTN Due to IME Trunk Rejection

A call is made to PSTN. The gateway has it in the learned IME route and the call is extended to IME trunk. The IME Trunk rejects the call, and the call treatment DOES cause the call to be redirected to the PSTN, so the call gets rejected. Depending on reason of IME trunk reject, different lastRedirectRedirectReason can be reported.

Field Names	CDR
globalCallID_callId	3
origLegCallIdentifier	300
destLegCallIdentifier	301
callingPartyNumber	2001
originalCalledPartyNumber	9728134987
lastRedirectRedirectOnBehalfOf	30
lastRedirectRedirectReason	496 OR 512 OR 528 OR 544 OR 560 OR 576 OR 592 OR 608 OR 624 OR 640 OR 656 OR 672 OR 688 OR 704
duration	10

IME Call Successfully Routed Out Through IME Trunk, Call Fallback to PSTN Due to Poor QoS

A call is made to PSTN. The gateway has it in the learned IME route and the call is extended to IME trunk. Call is routed out through IME trunk. Bad QoS later discovered and call falls back to PSTN.

Two CDRs are generated in this case; one for the IME call and one for fallback to PSTN call.

Table 3: For IME call

Field Names	CDR
globalCallID_callId	3
origLegCallIdentifier	300
destLegCallIdentifier	301
callingPartyNumber	2001
originalCalledPartyNumber	9728134987
OrigTerminationOnBehalfOf	30
lastRedirectRedirectOnBehalfOf	30
origCause_value	132
duration	5

Table 4: For Fallback to PSTN Call

Field Names	CDR
globalCallID_callId	3

Field Names	CDR
origLegCallIdentifier	300
destLegCallIdentifier	301
callingPartyNumber	2001
originalCalledPartyNumber	9728134987
lastRedirectRedirectOnBehalfOf	31
joinOnBehalfOf	31
lastRedirectRedirectReason	722
duration	5

Clear the PSTN Failback Call Case 1

A call is made to PSTN. The gateway has it in the learned IME route and the call is extended to IME trunk. Call is routed out through IME trunk. Bad QoS is discovered later and fall back initiated to PSTN. Call is rejected by PSTN gateway. Call is intercepted by Fallback Manager which clears the call. IME call remains untouched.

Two CDRs are generated in this case; one for the IME call and one for fallback to PSTN call.

Table 5: For IME Call

Field Names	CDR
globalCallID_callId	3
origLegCallIdentifier	300
destLegCallIdentifier	301
callingPartyNumber	2001
originalCalledPartyNumber	9728134987
lastRedirectRedirectOnBehalfOf	30
lastRedirectRedirectReason	0
duration	5

Table 6: For Fallback to PSTN Call

Field Names	CDR
globalCallID_callId	3
origLegCallIdentifier	300

Field Names	CDR
destLegCallIdentifier	301
callingPartyNumber	2001
originalCalledPartyNumber	9728134987
OrigTerminationOnBehalfOf	31
origCause_value	Existing PSTN GW cause codes
duration	0

Clear the PSTN Failback Call Case 2

A call is made to PSTN. The gateway has it in the learned IME route and the call is extended to IME trunk. Call is routed out through IME trunk. Bad QoS is discovered later and fall back initiated to PSTN. Cannot find link to IME call. Call is intercepted by Fallback Manager which clears the call. IME call remains untouched.

Two CDRs are generated in this case; one for the IME call and one for fallback to PSTN call.

Table 7: For IME Call

Field Names	CDR
globalCallID_callId	3
origLegCallIdentifier	300
destLegCallIdentifier	301
callingPartyNumber	2001
originalCalledPartyNumber	9728134987
lastRedirectRedirectOnBehalfOf	30
lastRedirectRedirectReason	0
duration	5

Table 8: For Fallback to PSTN Call

Field Names	CDR
globalCallID_callId	3
origLegCallIdentifier	300
destLegCallIdentifier	301
callingPartyNumber	2001
originalCalledPartyNumber	9728134987

Field Names	CDR
OrigTerminationOnBehalfOf	31
origCause_value	133 OR 134 OR existing cause codes
duration	0

Immediate Divert (to Voice-Messaging System)

Immediate Divert (IDivert) gets invoked in three different call states:

- You can invoke the IDivert feature while the incoming call is ringing. The CDR for the ringing case acts very similar to call forwarding, but the **origCalledPartyRedirectOnBehalfOf** and the **lastRedirectRedirectOnBehalfOf** fields specify Immediate Divert = 14.
- You can invoke the IDivert feature while the call is connected or on hold. These scenarios generate two CDRs. Both CDRs have the same **globalCallID_CallId** field. The first CDR applies to the original connection, and the second CDR applies to the call redirected to the voice-messaging system. The first call has the **origTerminationOnBehalfOf** and **destTerminationOnBehalfOf** fields set to Immediate Divert = 14.
- The call that gets redirected to the voice-messaging system has the **origCalledPartyRedirectOnBehalfOf** and **lastRedirectRedirectOnBehalfOf** fields set to Immediate Divert = 14.

IDivert CDR Examples

1. **IDivert during Alerting –** 40003 calls 40001, and while 40001 is ringing, 40001 presses the IDivert button, and call diverts to the voice-messaging system 40000.

Note If the call gets redirected by IDivert in the Alerting state, only one CDR gets generated.

Field Names	Original call CDR
globalCallID_callId	37
origLegCallIdentifier	16777327
destLegCallIdentifier	16777329
callingPartyNumber	40003
origCalledPartyNumber	40001
finalCalledPartyNumber	40000
lastRedirectDn	40001
origCause_Value	16
dest_CauseValue	0

Field Names	Original call CDR
origCalledPartyRedirectReason	50
lastRedirectRedirectReason	50
origCalledPartyRedirectOnBehalfOf	14
lastRedirectRedirectOnBehalfOf	14
joinOnBehalfOf	14

2. **IDivert during Connect –** 40003 calls 40001, and 40001 answers the call. 40001 decides to divert the caller to the voice-messaging system and presses the IDivert softkey. 40003 gets diverted to the voice-messaging system 40000.

   Because the call gets connected before the redirect, two CDRs get generated: one for the original connected call, and another for the call that is diverted to the voice-messaging system.

Field Names	Original Connected Call CDR	Diverted Call CDR
globalCallID_callId	38	38
origLegCallIdentifier	16777330	16777330
destLegCallIdentifier	16777331	16777332
callingPartyNumber	40003	40003
origCalledPartyNumber	40001	40001
finalCalledPartyNumber	40001	40000
lastRedirectDn	40001	40001
origCause_Value	0	16
dest_CauseValue	0	0
origCalledPartyRedirectReason	0	50
lastRedirectRedirectReason	0	50
origCalledPartyRedirectOnBehalfOf		14
lastRedirectRedirectOnBehalfOf		14
origTerminationOnBehalfOf	14	14
destTerminationOnBehalfOf	14	12
joinOnBehalfOf		14

Related Topics

Cisco Call Detail Records Field Descriptions, on page 127

Cisco Call Detail Records Codes, on page 171
Example Cisco Call Management Records, on page 209

IMS Application Server

1. IMS A with calls IMS B through Unified Communications Manager.

   The incoming invite to Unified Communications Manager contains:

   Icid: 5802170000010000000000A85552590A (say, PCV1) and orig_ioi: rcdn-85.swyan.open-ims.test (say, IOI_1).

   The INVITE from the Unified Communications Manager to IMS B has the same icid as 5802170000010000000000A85552590A (PCV1), orig_ioi as rcdn-85.swyan.open-ims.test (IOI_1).

2. When B answers, 200 OK to Unified Communications Manager has icid as 5802170000010000000000A85552590A (PCV1), orig_ioi as rcdn-85.swyan.open-ims.test (IOI_1), and term_ioi, rcdn-86.swyan.open-ims.test (IOI_2). There could be extra fields with this 200 OK.
3. The 200 OK from Unified Communications Manager to IMS A has icid 5802170000010000000000A85552590A (PCV1), orig_ioi as rcdn-85.swyan.open-ims.test (IOI_1), and term_ioi as rcdn-86.swyan.open-ims.test (IOI_2). The extra fields in the 200 OK will be passed to IMS A.

CDR	Side A			Side B		
parties	icid	orig_ioi	term_ioi	icid	orig_ioi	term_ioi
A-B	PCV1	IOI_1	IOI_2	PCV1	IOI_1	IOI_2

Fields Names	CDR
globalCallID_callId	3
origLegCallIdentifier	300
destLegCallIdentifier	301
origDeviceName	CUCM_ISC_TRUNK1
destDeviceName	CUCM_ISC_TRUNK2
IncomingICID	5802170000010000000000A85552590A
IncomingOrigIOI	rcdn-85.swyan.open-ims.test
IncomingTermIOI	rcdn-86.swyan.open-ims.test
OutgoingICID	5802170000010000000000A85552590A
OutgoingOrigIOI	rcdn-85.swyan.open-ims.test
OutgoingTermIOI	rcdn-86.swyan.open-ims.test

Intercom Calls

The Intercom feature provides one-way audio; therefore, the CDR reflects one-way audio. For talk-back intercom, two-way audio exists, and the CDR reflects two-way audio.

The Intercom feature requires a partition (intercom partition), and existing CDR partition fields get used to identify intercom calls.

The following two examples show CDRs for intercom.

Intercom CDR Examples

1. **Whisper Intercom** - Phone 20000 invokes the intercom. The configured intercom partition name specifies "Intercom."

Field Names	Original Call CDR
globalCallID_callId	1111000
origLegCallIdentifier	21822467
destLegCallIdentifier	21822468
callingPartyNumber	20000
originalCalledPartyNumber	20001
finalCalledPartyNumber	20001
origCause_Value	16
dest_CauseValue	0
origMediaTransportAddress_IP	0
origMediaTransportAddress_Port	0
destMediaTransportAddress_IP	-47446006
destMediaTransportAddress_Port	28480
origCalledPartyNumberPartition	Intercom
callingPartyNumberPartition	Intercom
finalCalledPartyNumberPartition	Intercom
duration	5

2. **Talk-Back Intercom** - Phone 20000 presses the intercom button. 20001 invokes Talk-Back and talks to 20000. The configured intercom partition name specifies "Intercom."

Field Names	Original Call CDR
globalCallID_callId	1111000

Field Names	Original Call CDR
origLegCallIdentifier	21822469
destLegCallIdentifier	21822470
callingPartyNumber	20000
originalCalledPartyNumber	20001
finalCalledPartyNumber	20001
origCause_Value	16
dest_CauseValue	0
origMediaTransportAddress_IP	-131332086
origMediaTransportAddress_Port	29458
destMediaTransportAddress_IP	-47446006
destMediaTransportAddress_Port	29164
origCalledPartyNumberPartition	Intercom
callingPartyNumberPartition	Intercom
finalCalledPartyNumberPartition	Intercom
duration	5

Related Topics

Cisco Call Detail Records Field Descriptions, on page 127
Cisco Call Detail Records Codes, on page 171
Example Cisco Call Management Records, on page 209

IPv6 Calls

Unified Communications Manager supports IPv6 in this release. There are two new fields in the CDR for this feature:

- **origIpv4v6Addr—**This field identifies the IP address of the device that originates the call signaling. The field can be in either IPv4 or IPv6 format depending on the IP address type that gets used for the call.
- **destIpv4v6Addr**—This field identifies the IP address of the device that terminates the call signaling. The field can be in either IPv4 or IPv6 format depending on the IP address type that gets used for the call.

The following CDR examples display IPv6 with successful and unsuccessful calls.

Successful Calls

1. A talks to B; A hangs up. A is configured as v4_only and B is configured as v4_only. The new fields **origIpv4v6Addr** and **destIpv4v6Addr** get populated with the format of their respective v4 addresses.

Field Names	Values
globalCallID_callId	1
origLegCallIdentifier	100
destLegCallIdentifier	101
callingPartyNumber	2001
originalCalledPartyNumber	2309
finalCalledPartyNumber	2309
lastRedirectDn	2309
origIpAddr	352737802
destIpAddr	1878566390
origIpv4v6Addr	10.90.6.21
destIpv4v6Addr	10.90.7.144
duration	60

2. A talks to B; A hangs up. A is configured as v6_only and B is configured as v6_only. The new fields **origIpv4v6Addr anddestIpv4v6Addr** get populated with the format of their respective v6 addresses.

Field Names	Values
globalCallID_callId	1
origLegCallIdentifier	100
destLegCallIdentifier	101
callingPartyNumber	2001
originalCalledPartyNumber	2309
finalCalledPartyNumber	2309
lastRedirectDn	2309
origIpAddr	0
destIpAddr	0
origIpv4v6Addr	2001:fecd:ba23:cd1f:dcb1:1010:9234:40881
destIpv4v6Addr	2001:420:1e00:e5:217:8ff:fe5c:2fa9

Field Names	Values
duration	60

3. A talks to B; A hangs up. A is configured as v4_only and B is configured as v6_only. The new fields **origIpv4v6Addr** and **destIpv4v6Addr** get populated with the format of their respective v4/v6 addresses.

Field Names	Values
globalCallID_callId	1
origLegCallIdentifier	100
destLegCallIdentifier	101
callingPartyNumber	2001
originalCalledPartyNumber	2309
finalCalledPartyNumber	2309
lastRedirectDn	2309
origIpAddr	352737802
destIpAddr	-1878566390
origIpv4v6Addr	10.90.6.21
destIpv4v6Addr	10.90.7.144
duration	60

4. A talks to B; A hangs up. A is configured as v4_v6 and B is configured as v4_only. In this case, media negotiates v4. The new fields **origIpv4v6Addr and destIpv4v6Addr** get populated with the format of their respective v4 addresses.

Field Names	Values
globalCallID_callId	1
origLegCallIdentifier	100
destLegCallIdentifier	101
callingPartyNumber	2001
originalCalledPartyNumber	2309
finalCalledPartyNumber	2309
lastRedirectDn	2309
origIpAddr	352737802
destIpAddr	-1878566390

Field Names	Values
origIpv4v6Addr	10.90.6.21
destIpv4v6Addr	10.90.7.144
duration	60

5. A talks to B; A hangs up. A is configured as v4_v6 and B is configured as v6_only. In this case, media negotiates v6. The new fields **origIpv4v6Addr** and **destIpv4v6Addr** get populated with the format of their respective v6 addresses.

Field Names	Values
globalCallID_callId	1
origLegCallIdentifier	100
destLegCallIdentifier	101
callingPartyNumber	2001
originalCalledPartyNumber	2309
finalCalledPartyNumber	2309
lastRedirectDn	2309
origIpAddr	352737802
destIpAddr	0
origIpv4v6Addr	2001:fecd:ba23:cd1f:dcb1:1010:9234:4088
destIpv4v6Addr	2001:420:1e00:e5:217:8ff:fe5c:2fa9
duration	60

Unsuccessful Calls

1. A calls B; A abandons the call. A is configured as v4_only and B is configured as v6_only. The new field **origIpv4v6Addr** gets populated with the format of its v4 address. The new field **destIpv4v6Addr** does not get populated.

Field Names	Values
globalCallID_callId	1
origLegCallIdentifier	100
destLegCallIdentifier	101
callingPartyNumber	2001
originalCalledPartyNumber	2309

Field Names	Values
finalCalledPartyNumber	2309
lastRedirectDn	2309
origIpAddr	352737802
destIpAddr	-569419254
origIpv4v6Addr	10.90.15.222
destIpv4v6Addr	
duration	0

2. A calls B; the call fails. A is configured as v6_only and B is configured as v4_v6. The new field **origIpv4v6Addr** gets populated with the format of its v6 address. The new field **destIpv4v6Addr** does not get populated in this case.

Field Names	Values
globalCallID_callId	1
origLegCallIdentifier	100
destLegCallIdentifier	101
callingPartyNumber	2001
originalCalledPartyNumber	2309
finalCalledPartyNumber	2309
lastRedirectDn	2309
origIpAddr	0
destIpAddr	0
origIpv4v6Addr	2001:fecd:ba23:cd1f:dcb1:1010:9234:4088
destIpv4v6Addr	
duration	0

Related Topics

Cisco Call Detail Records Field Descriptions, on page 127
Cisco Call Detail Records Codes, on page 171
Example Cisco Call Management Records, on page 209

Legacy Call Pickup

Legacy Call Pickup calls act similar to forwarded calls. Legacy Call Pickup uses the redirect call control primitive like call forwarding. Some of the important CDR fields for Legacy Call Pickup calls follow:

- The **originalCallPartyNumber** field contains the number of the original called party.
- The **finalCalledPartyNumber** field specifies the number of the party that picks up the call.
- The **lastRedirectDn** field specifies the number that rings when the call gets picked up.
- The **origCalledPartyRedirectReason** field specifies the reason that the call gets redirected the first time. For call pickup calls, this field can contain **Call Pickup = 5**.
- The **lastRedirectRedirectReason** field specifies the reason that the call gets redirected the last time. For call pickup, this field can contain **Call Pickup = 5**.
- The **origCalledPartyRedirectOnBehalfOf** field identifies which feature redirects the call for the first redirect. For call pickup, this field specifies **Pickup = 16**.
- The **lastRedirectRedirectOnBehalfOf** field identifies which feature redirects the call for the last redirect. For call pickup, this field specifies **Pickup = 16**.

Legacy Call Pickup CDR Example

Call from the PSTN to extension 2001; 2001 and 2002 exist in the same pickup group. 2002 picks up the call that rings on 2001. 2002 answers the call, and the call connects between the PSTN caller and 2002. They talk for 2 minutes.

Field Names	CDR
globalCallID_callId	22
origLegCallIdentifier	1
destLegCallIdentifier	2
callingPartyNumber	9728134987
originalCalledPartyNumber	2001
finalCalledPartyNumber	2002
lastRedirectDn	2001
origCause_Value	0
dest_CauseValue	16
origCalledPartyRedirectReason	0
lastRedirectRedirectReason	5
origCalledPartyRedirectOnBehalfOf	16

Field Names	CDR
lastRedirectRedirectOnBehalfOf	16
duration	120

Related Topics

Cisco Call Detail Records Field Descriptions, on page 127
Cisco Call Detail Records Codes, on page 171
Example Cisco Call Management Records, on page 209

Local Route Groups and Called Party Transformation

In this release, Unified Communications Manager supports the new feature, local route groups and called party transformation. The device reports the Called Party Number that it outpulsed back to Call Control only if called party transformation occurs. This action gets recorded in the CDR in the new field **outpulsedCalledPartyNumber**.

Local Route Groups and Called Party Normalization CDR Example

A call gets placed from an enterprise phone in Dallas to the PSTN; the dialed number specifies 9.5551212.

The translation causes the called party number to take the digits as dialed by the originator, discard PreDot and add the Prefix +1 214.

The **finalCalledPartyNumber** in the CDR comprises the globally unique E.164 string +12145551212.

If a San Jose gateway gets selected, it transforms the global string +1 214 555 1212 into 12145551212, and if a Dallas gateway gets selected, the global string gets transformed into 2145551212.

The device returns this global string to Call Control as the **outpulsedCalledPartyNumber; it gets** recorded in the CDR.

The following CDR gets created if the San Jose gateway gets selected.

Field Names	Values
globalCallID_callId	1
origLegCallIdentifier	100
destLegCallIdentifier	101
callingPartyNumber	2001
originalCalledPartyNumber	+12145551212
finalCalledPartyNumber	2309
lastRedirectDn	2309
origCause_Value	16
dest_CauseValue	0

Field Names	Values
duration	60
outpulsedCalledPartyNumber	12145551212

The following CDR gets created if the Dallas gateway gets selected.

Field Names	Values
globalCallID_callId	1
origLegCallIdentifier	100
destLegCallIdentifier	101
callingPartyNumber	2001
originalCalledPartyNumber	+12145551212
finalCalledPartyNumber	+12145551212
lastRedirectDn	+12145551212
origCause_Value	16
dest_CauseValue	0
duration	60
outpulsedCalledPartyNumber	2145551212

Related Topics

Cisco Call Detail Records Field Descriptions, on page 127
Cisco Call Detail Records Codes, on page 171
Example Cisco Call Management Records, on page 209

Logical Partitioning Calls

The Telecom Regulatory Authority of India (TRAI) requires that voice traffic over an enterprise data network and a PSTN network remain separate. The logical partitioning feature ensures that a single system can be used to support both types of calls as long as calls that pass through a PSTN gateway can never directly connect to a VoIP phone or VoIP PSTN gateway in another geographic location (geolocation).

CDR Example for Call Termination Cause Code CCM_SIP_424_BAD_LOCATION_INFO

A SIP trunk call goes from cluster1 to cluster2. The call contains a geolocation header but does not include an XML location. Cluster2 releases the call with a SIP Status code of 424 (bad location information [decimal value = 419430421]).

Cause code CCM_SIP_424_BAD_LOCATION_INFO gets logged for calls that are cleared because of bad location information by the SIP trunk on the Unified Communications Manager. The remote endpoint on the

SIP trunk can send the 424 SIP Status code for cases when the geolocation information is bad for some of the following reasons:

- The geolocation header indicates the inclusion of PIDF-LO, but the message body does not carry this information.
- The geolocation header has a CID header that refers to a URL, but no corresponding Content-IP header with the same URL exists.
- The geolocation header has a URL other than the CID header (that is a SIP, or SIPS URL).

Refer to additional CDR examples for more information on other call termination cause codes.

Field Names	Values
globalCallID_callId	1
origLegCallIdentifier	100
destLegCallIdentifier	101
callingPartyNumber	2001
originalCalledPartyNumber	9900
finalCalledPartyNumber	9900
lastRedirectDn	9900
origCause_Value	0
dest_CauseValue	419430421
duration	0

CDR Example for Call Termination Cause Code 503

Call 82291002 from cluster1 gets call-forwarded to the PSTN 41549901. A call occurs from cluster2 from DN 89224001 to cluster1 DN 82291002. The call gets denied because of logical partitioning with a call termination cause code of CCM_SIP_503_SERVICE_UNAVAIL_SER_OPTION_NOAVAIL [decimal value of -1493172161]) for the dest_CauseValue.

Cause code CCM_SIP_503_SERVICE_UNAVAIL_SER_OPTION_NOAVAIL gets logged for calls that get cleared because of restricted logical partitioning policy checks during the call establishment phase (basic call, call forwarding, call pickup, call park, meet-me conferences, and so forth). Refer to additional CDR examples for more information on other call termination cause codes.

Field Names	Values
globalCallID_callId	1
origLegCallIdentifier	100
destLegCallIdentifier	101
callingPartyNumber	89224001

Field Names	Values
originalCalledPartyNumber	82291002
finalCalledPartyNumber	41549901
lastRedirectDn	82291002
origCause_Value	0
dest_CauseValue	-1493172161
duration	0

Related Topics

CDR Examples, on page 15
Cisco Call Detail Records Field Descriptions, on page 127
Cisco Call Detail Records Codes, on page 171
Example Cisco Call Management Records, on page 209

Malicious Calls

When a call gets identified as a malicious call (button press), the local Unified Communications Manager network flags the call. The Comment field flags the malicious call.

Malicious Calls CDR Example

The following table contains an example CDR of a customer call that gets marked as malicious.

Calling Party	Calling Partition	Original Called Party	Original Called Partition	Orig Cause	Dest Cause	Comment
9728552001	CUST	5555	ACNTS	0	16	'callFlag=MALICIOUS'

Related Topics

Cisco Call Detail Records Field Descriptions, on page 127
Cisco Call Detail Records Codes, on page 171
Example Cisco Call Management Records, on page 209

Meet-Me Conferences

A meet-me conference occurs when several parties individually dial into a conference bridge at a predetermined time.

The Cisco Secure Conference feature uses the existing **callSecuredStatus** field to display the highest security status that a call reaches. For meet-me conferences, the system clears calls that try to join the conference but do not meet the security level of the meet-me conference with a terminate cause = 58 (Bearer capability not presently available).

Meet-Me Conference CDR Example

The following table contains an example CDR for the following scenario. 5001 specifies the dial-in number. The conference bridge device signifies special significance to the Unified Communications Manager, and calls to the conference bridge appear as forwarded calls; that is, User A phones the predetermined number (5001); the call gets forwarded to a conference bridge port. The conference bridge port appears with a special number of the form "b0019901001."

- User A (2001) calls into a meet-me conference bridge with the phone number 5001.
- User B (2002) calls into a meet-me conference bridge with the phone number 5001.
- User C (2003) calls into a meet-me conference bridge with the phone number 5001.

	Calling Party	Calling Partition	Original Called Party	Original Called Partition	Final Called Party	Final Called Partition	Last Redirect Party	Last Redirect Partition	Duration
A	2001	Accounts	5001		b0019901001		b0019901001		70
B	2002	Accounts	5001		b0019901001		b0019901001		65
C	2003	Accounts	5001		b0019901001		b0019901001		80

Related Topics

Cisco Call Detail Records Field Descriptions, on page 127
Cisco Call Detail Records Codes, on page 171
Example Cisco Call Management Records, on page 209

Mobility

The following call detail record (CDR) fields apply specifically to Mobility calls. If the call does not invoke a mobility feature, these fields remain empty:

- mobileCallingPartyNumber
- finalMobileCalledPartyNumber
- origMobileDeviceName
- destMobileDeviceName
- origMobileCallDuration
- destMobileCallDuration
- mobileCallType

The system generates a standard CDR for every call that uses the Mobility features. When a call gets split, redirected, or joined by the Mobility feature, the corresponding **OnBehalfOf** code represents a new value that is designated to Mobility. If any of the following **OnBehalfOf** fields has the Mobility code of 24, the CDR has the Mobility call type:

- origCallTerminationOnBehalfOf

- destCallTerminationOnBehalfOf
- origCalledPartyRedirectOnBehalfOf
- lastRedirectRedirectOnBehalfOf
- joinOnBehalfOf

MobileCallType Values

The following table displays the field values for the mobileCallType CDR field. Cisco Analysis and Reporting (CAR) uses the mobileCallType field to determine the CAR call type. If a single call invokes more than one mobility feature, the value of the mobileCallType field represents the integer values added together. For example, if a call uses the Mobile Connect feature and then invokes Hand-Out, the mobile call type will be 136 (8 + 128).

Mobility Feature	mobileCallType Value
Nonmobility call	0
Dial via Office Reverse Callback	1
Dial via Office Forward	2
Reroute remote destination call to enterprise network	4
Mobile Connect	8
Interactive Voice Response	10
Enterprise Feature Access	20
Hand-In	40
Hand-Out	80
Redial	100
Least Cost Routing with dial-via-office reverse callback	200
Least Cost Routing with dial-via-office forward	82
Send call to mobile	800
Session handoff	1000

Last Redirect Reason

In legacy deployments before 10.0, CAR uses the lastRedirectReason field to identify the mobility call type. The following table shows the Mobility values for lastRedirectReason.

Mobility Feature	lastRedirectReason Value
Hand-In	303

Mobility Feature	lastRedirectReason Value
Hand-Out	319
Mobile Connect	335
Redial	351
Interactive Voice Response	399
Dial via Office Reverse Callback	401
Enterprise Feature Access	402
Session Handoff	403
Send call to mobile	415
Reroute remote destination call to enterprise network	783

Mobility CDR Examples

The following examples demonstrate how mobility features display in CDR records:

1. **Mobile phone initiates Dial via Office Reverse Callback** - A mobile phone with a device name of BOTSAU, a mobile number of 2145551234, and an enterprise number of 1000 invokes the dial-via-office Reverse callback feature to place a call to extension 2000. The MAC address of the called device is SEP001FCAE90004. The IP address of the SIP gateway is 10.194.108.70. The total call duration is 55 seconds.

Field	Dial via Office Reverse Callback CDR
origCallTerminationOnBehalfOf	0
destCallTerminationOnBehalfOf	12
origCalledRedirectOnBehalfOf	24
lastRedirectOnBehalfOf	24
joinOnBehalfOf	24
origCalledPartyRedirectReason	401
lastRedirectReason	401
origDeviceName	10.194.108.70
destDeviceName	SEP001FCAE9004
finalCalledPartyNumber	2000
huntPilotDN	
mobileCallingPartyNumber	2145551234

Field	Dial via Office Reverse Callback CDR
finalMobileCalledPartyNumber	
origMobileDeviceName	BOTSAU
destMobileDeviceName	
origMobileCallDuration	55
destMobileCallDuration	
mobileCallType	1

2. **Mobile phone initiates Dial via Office Forward** - Mobile phone 2145551234 initiates the Dial via Office Forward feature to place a call. The mobile phone has a device name of BOTSAU and is mapped to enterprise number 1000. The called number is extension 823006 with a device MAC address of SEP001FCAE90004. The call crosses a SIP gateway at 10.194.108.70 and lasts for a total duration of 120 seconds.

Field	Dial via Office Forward CDR
origCallTerminationOnBehalfOf	0
destCallTerminationOnBehalfOf	12
origCalledRedirectOnBehalfOf	0
lastRedirectOnBehalfOf	0
joinOnBehalfOf	0
origCalledPartyRedirectReason	0
lastRedirectReason	0
origDeviceName	10.194.108.70
destDeviceName	SEP001FCAE90004
finalCalledPartyNumber	823006
huntPilotDN	
mobileCallingPartyNumber	2145551234
finalMobileCalledPartyNumber	
origMobileDeviceName	BOTSAU
destMobileDeviceName	
origMobileCallDuration	120
destMobileCallDuration	0

Field	Dial via Office Forward CDR
mobileCallType	2

3. **Call to remote destination is rerouted to enterprise number** - Cisco Unified IP Phone SEP001FCAE90004, at extension 2000, dials mobile number 2145551234. The destination mobile phone is mapped to enterprise number 1000 and has the Reroute Remote Destination Calls to Enterprise Number service parameter enabled in Unified Communications Manager. Unified Communications Manager reroutes the mobile call to enterprise number 1000. The call crosses SIP gateway GW_SIP and lasts for a total duration of 60 seconds.

Field	Reroute Remote Detination CDR
origCallTerminationOnBehalfOf	0
destCallTerminationOnBehalfOf	12
origCalledRedirectOnBehalfOf	24
lastRedirectOnBehalfOf	24
joinOnBehalfOf	0
origCalledPartyRedirectReason	783
lastRedirectReason	783
origDeviceName	SEP001FCAE90004
destDeviceName	GW_SIP
finalCalledPartyNumber	1000
huntPilotDN	
mobileCallingPartyNumber	
finalMobileCalledPartyNumber	2145551234
origMobileDeviceName	
destMobileDeviceName	2145551234:rdp
origMobileCallDuration	0
destMobileCallDuration	60
mobileCallType	4

4. **Mobile phone invokes deskphone call pickup** - Cisco Unified IP Phone SEP001FCAE90004 calls extension 1000, which is shared between a desk phone and a mobile device. The mobile phone answers the call and then hangs up, triggering the desktop pickup feature. The desktop call pickup timer runs for about 10 seconds before expiring. After the timer expires, the call is resumed on a Wi-Fi device for another 10 seconds.

Field	Desktop Call Pickup CDR
origCallTerminationOnBehalfOf	0
destCallTerminationOnBehalfOf	12
origCalledRedirectOnBehalfOf	0
lastRedirectOnBehalfOf	0
joinOnBehalfOf	0
origCalledPartyRedirectReason	0
lastRedirectReason	0
origDeviceName	SEP001FCAE90004
destDeviceName	GW_SIP
finalCalledPartyNumber	1000
huntPilotDN	
mobileCallingPartyNumber	
finalMobileCalledPartyNumber	
origMobileDeviceName	
destMobileDeviceName	
origMobileCallDuration	0
destMobileCallDuration	10
mobileCallType	8

5. **Mobile Connect Call - Single Number Reach Voicemail policy set to Timer Control** - Cisco Unified IP Phone SEP001FCAE90004, at extension 2000, calls enterprise number 1000. Mobile Connect is invoked and both the desk phone and mobile phone ring. The mobile phone uses a mobile identity with a device name of BOTSARAH. The Single Number Reach Voicemail policy is set to Timer Control. The call traverses a SIP gateway and lasts for 10 minutes.

Field	CDR
origCallTerminationOnBehalfOf	0
destCallTerminationOnBehalfOf	12
origCalledRedirectOnBehalfOf	0
lastRedirectOnBehalfOf	0
joinOnBehalfOf	0

Field	CDR
origCalledPartyRedirectReason	0
lastRedirectReason	0
origDeviceName	SEP001FCAE90004
destDeviceName	GW_SIP
finalCalledPartyNumber	1000
huntPilotDN	
mobileCallingPartyNumber	
finalMobileCalledPartyNumber	2145551234
origMobileDeviceName	
destMobileDeviceName	BOTSARAH
origMobileCallDuration	0
destMobileCallDuration	10
mobileCallType	8

6. **Mobile Connect Call - Single Number Reach Voicemail Policy set to UserControl Mode** -Cisco Unified IP Phone SEP001FCAE91231 at enterprise number 238011 calls across a SIP gateway, GW_SIP. The called party is SEP001FCEA91289, at enterprise number 238006 and mobile number 14089022179. Three CDRs are produced.

Field	Announcement to user	0 Duration Call	IP Phone to Mobile Phone
origCallTerminationOnBehalfOf	24	24	12
destCallTerminationOnBehalfOf	24	24	24
origCalledRedirectOnBehalfOf	24	24	24
lastRedirectOnBehalfOf	24	24	24
joinOnBehalfOf	0	0	24
origCalledPartyRedirectReason	335	335	335
lastRedirectReason	335	335	335
origDeviceName	SEP001FCAE91231	ParkingLotDevice	SEP001FCAE91231
destDeviceName	GW_SIP	GW_SIP	GW_SIP
finalCalledPartyNumber	238006	238006	238006

Field	Announcement to user	0 Duration Call	IP Phone to Mobile Phone
huntPilotDN			
mobileCallingPartyNumber			
finalMobileCalledPartyNumber	14089022179	14089022179	14089022179
origMobileDeviceName			
destMobileDeviceName	14089022179:rdp	14089022179:rdp	14089022179:rdp
origMobileCallDuration	0	0	
destMobileCallDuration	3		6
mobileCallType	8	8	8

7. **Mobile phone makes an Enterprise Feature Access (EFA) call with two-stage dialing** - Remote destination deepak-RDP, at 4089022179 with shared-line desk phone SEP001EBE90DE95 and enterprise number 238006, calls internal desk phone SEP001FCAE91231, at enterprise number 238011, using Enterprise Feature Access two-stage dialing. Total call duration is 30 seconds. Two CDRs are produced: one for the mobile phone dialing the EFA access codes into Unified Communications Manager and the second for the mobile phone to desk phone conversation.

Field	Mobile Phone to Unified Communications Manager	Mobile Phone to Desk Phone
origCallTerminationOnBehalfOf	24	12
destCallTerminationOnBehalfOf	24	24
origCalledRedirectOnBehalfOf	24	24
lastRedirectOnBehalfOf	24	24
joinOnBehalfOf	24	24
origCalledPartyRedirectReason	402	402
lastRedirectReason	402	402
origDeviceName	GW_SIP	GW_SIP
destDeviceName	ParkingLotDevice	SEP001FCAE91231
finalCalledPartyNumber	00111101001	238011
huntPilotDN		
mobileCallingPartyNumber	14089022179	14089022179
finalMobileCalledPartyNumber		
origMobileDeviceName	14089022179:rdp	14089022179:rdp

Field	Mobile Phone to Unified Communications Manager	Mobile Phone to Desk Phone
destMobileDeviceName		
origMobileCallDuration	5	25
destMobileCallDuration	0	
mobileCallType	32	32

8. **Mobile phone makes a Mobile Voice Access call** - Remote destination 4089022179, with shared line desk phone SEP001EBE90DE95 at enterprise number 238006, uses Mobile Voice Access to call an internal desk phone SEP00000000000002 at enterprise number 238011. The remote destination has a remote destination profile of deepak-rdp. The call traverses SIP gateway GW_SIP and lasts for 60 seconds.

Field	Mobile Phone to Desk Phone
origCallTerminationOnBehalfOf	12
destCallTerminationOnBehalfOf	0
origCalledRedirectOnBehalfOf	24
lastRedirectOnBehalfOf	24
joinOnBehalfOf	24
origCalledPartyRedirectReason	399
lastRedirectReason	399
origDeviceName	GW_SIP
destDeviceName	SEP00000000000002
finalCalledPartyNumber	238011
huntPilotDN	
mobileCallingPartyNumber	14089022179
finalMobileCalledPartyNumber	
origMobileDeviceName	14089022179:rdp
destMobileDeviceName	
origMobileCallDuration	60
destMobileCallDuration	
mobileCallType	16

9. **Mobility Hand-In** - Cisco Unified IP Phone SEP001FCAE91231 at enterprise number 238011, calls enterprise number 238006, which is unregistered on the VoIP side, but registered to the smartphone TCTSAU. The mobile identity of the smartphone is 14089022179. At the beginning of the call TCTSAU is located in a cellular network, but the device moves into Wi-Fi range and the Hand-In feature is invoked to move the call into the enterprise. The total call duration is 85 seconds, with the called device in Wi-Fi range for the last 30 seconds.

Field	IP Phone to Cellular Phone	IP Phone to IP Phone
origCallTerminationOnBehalfOf	24	12
destCallTerminationOnBehalfOf	24	24
origCalledRedirectOnBehalfOf	0	0
lastRedirectOnBehalfOf	0	24
joinOnBehalfOf	0	24
origCalledPartyRedirectReason	0	303
lastRedirectReason	0	303
origDeviceName	SEP001FCAE91231	SEP001FCAE91231
destDeviceName	GW_SIP	TCTSAU
finalCalledPartyNumber	238006	238006
huntPilotDN		
mobileCallingPartyNumber		
finalMobileCalledPartyNumber	14089022179	
origMobileDeviceName		
destMobileDeviceName	TCTSAU	
origMobileCallDuration	0	0
destMobileCallDuration	55	
mobileCallType	8	72

10. **Mobility Hand-Out** - Cisco Unified IP Phone SEP001FCAE94005 at enterprise number 238011, calls a dual-mode smartphone with a mobile identity of 14089022179, at enterprise number 238006. The smartphone is in local Wi-Fi range when the call is answered and the two parties speak for 27 seconds. The smartphone moves out of the enterprise network and the call is switched to the cell network, after which the parties continue to speak for another 25 seconds.

Field	IP Phone to IP Phone	IP Phone to Cell Network
origCallTerminationOnBehalfOf	24	0
destCallTerminationOnBehalfOf	24	12

Field	IP Phone to IP Phone	IP Phone to Cell Network
origCalledRedirectOnBehalfOf	0	0
lastRedirectOnBehalfOf	0	24
joinOnBehalfOf	0	24
origCalledPartyRedirectReason	0	0
lastRedirectReason	0	319
origDeviceName	SEP001FCAE94005	SEP001FCAE94005
destDeviceName	TCTSAU	GW_SIP
finalCalledPartyNumber	238006	238006
huntPilotDN		
mobileCallingPartyNumber		
finalMobileCalledPartyNumber		14089022179
origMobileDeviceName		
destMobileDeviceName		TCTSAU
origMobileCallDuration	0	0
destMobileCallDuration	0	23
mobileCallType	0	128

11. **Mobile phone invokes Least Cost Routing Hand-Out using Dial via Office Reverse Callback** - A dual-mode phone, BOTSAU, with a mobile identity of 14089022179, is within the enterprise wifi network and registered to enterprise number 238006. The phone invokes Dial via Office Reverse Callback (DVOR) using least cost routing to call enterprise number 238011. The two parties speak for 25 seconds, but the mobile phone moves out of Wi-Fi range, triggering the handout feature to the cellular network. On the cell network, the two parties speak for another 35 seconds.

Field	DVOR callback	IP phone to IP phone	Mobile phone to IP phone
origCallTerminationOnBehalfOf	24	24	0
destCallTerminationOnBehalfOf	24	24	12
origCalledRedirectOnBehalfOf	0	0	0
lastRedirectOnBehalfOf	0	0	24
joinOnBehalfOf	0	0	24
origCalledPartyRedirectReason	0	0	0

Field	DVOR callback	IP phone to IP phone	Mobile phone to IP phone
lastRedirectReason	0	0	319
origDeviceName	ParkingLotDevice	BOTSAU	GW_SIP
destDeviceName	GW_SIP	SEP001FCAE91231	SEP001FCAE91231
finalCalledPartyNumber	238006	238011	238011
huntPilotDN			
mobileCallingPartyNumber			14089022179
finalMobileCalledPartyNumber	14089022179		
origMobileDeviceName			BOTSAU
destMobileDeviceName	BOTSAU		
origMobileCallDuration	0	0	35
destMobileCallDuration	0	0	
mobileCallType	0	0	512

12. **Mobile Phone invokes Least Cost Routing Hand-Out using Dial via Office Forward** - A dual-mode phone, BOTSAU, with a mobile number of 14089022179, is mapped to enterprise number 238006 and is within Wi-Fi range of the enterprise. The phone invokes Dial via Office Forward with least cost routing to place a call to enterprise number 238011, which is registered to a Cisco Unified IP Phone SEP001FCAE91006. The two parties talk for 30 seconds before the mobile phone moves out of Wi-Fi range and the call is handed out to the cell network, following which the call continues for another 25 seconds.

Field	IP Phone to IP Phone	Mobile Phone to IP Phone
origCallTerminationOnBehalfOf	24	12
destCallTerminationOnBehalfOf	24	0
origCalledRedirectOnBehalfOf	0	0
lastRedirectOnBehalfOf	0	24
joinOnBehalfOf	0	24
origCalledPartyRedirectReason	0	0
lastRedirectReason	0	319
origDeviceName	BOTSAU	GW_SIP
destDeviceName	SEP001FCAE91006	SEP001FCAE91006
finalCalledPartyNumber	238011	238011

Field	IP Phone to IP Phone	Mobile Phone to IP Phone
huntPilotDN		
mobileCallingPartyNumber		14089022179
finalMobileCalledPartyNumber		
origMobileDeviceName		BOTSAU
destMobileDeviceName		
origMobileCallDuration	0	0
destMobileCallDuration	0	25
mobileCallType	0	130

13. **Send Call to Mobile** - A Cisco Unified IP Phone SEP001FCAE90001 at 238011, makes a call to enterprise number 238006. The called party answers the call on Cisco Unified IP Phone SEP001FCAE90022. The conversation continues for 45 seconds before the called party presses the Mobility softkey to send the call to the mobile phone, BOTSAU, at 12145551234. The call continues on the mobile phone for another 35 seconds. The total call duration is 55 seconds.

Field	Announcement	IP Phone to IP Phone	IP Phone to Mobile Phone
origCallTerminationOnBehalfOf	24	24	24
destCallTerminationOnBehalfOf	24	24	12
origCalledRedirectOnBehalfOf	0	0	0
lastRedirectOnBehalfOf	0	0	24
joinOnBehalfOf	0	0	24
origCalledPartyRedirectReason	0	0	0
lastRedirectReason	0	0	415
origDeviceName	SEP001FCAE90001	SEP001FCAE90001	SEP001FCAE90001
destDeviceName	GW_SIP	SEP001FCAE90022	GW_SIP
finalCalledPartyNumber	238006	238006	238006
huntPilotDN			
mobileCallingPartyNumber			
finalMobileCalledPartyNumber	12145551234		12145551234
origMobileDeviceName			
destMobileDeviceName	BOTSAU		BOTSAU

Field	Announcement	IP Phone to IP Phone	IP Phone to Mobile Phone
origMobileCallDuration	0	0	0
destMobileCallDuration	0	0	35
mobileCallType	0	0	2048

14. **Session Handoff** - Cisco Unified IP Phone SEP001FCAE90001, at extension 1000, calls extension 2500. The phone rings at both a desk phone and a mobile phone. The called party answers on a mobile phone, BOTSAU at mobile number 2145551234 and a conversation begins. After 35 seconds, the called party triggers the Session Handoff feature and transfers the call to a desk phone. The call continues on desk phone SEP001FCAE90022 for another 60 seconds.

Field	Parking Lot to Desk Phone	IP Phone to Mobile Phone	IP Phone to IP Phone
origCallTerminationOnBehalfOf	24	24	24
destCallTerminationOnBehalfOf	24	24	12
origCalledRedirectOnBehalfOf	0	0	0
lastRedirectOnBehalfOf	0	0	24
joinOnBehalfOf	0	0	24
origCalledPartyRedirectReason	0	0	0
lastRedirectReason	0	0	403
origDeviceName	SEP001FCAE90001	SEP001FCAE90001	SEP001FCAE90001
destDeviceName	SEP001FCAE90022	BOTSARAH	SEP001FCAE90022
finalCalledPartyNumber	2500	2500	2500
huntPilotDN			
mobileCallingPartyNumber			
finalMobileCalledPartyNumber		2145551234	
origMobileDeviceName			
destMobileDeviceName		BOTSARAH	
origMobileCallDuration	0	0	0
destMobileCallDuration	0	15	10
mobileCallType	0	0	5096

Related Topics

Cisco Call Detail Records Field Descriptions, on page 127
Cisco Call Detail Records Codes, on page 171
Example Cisco Call Management Records, on page 209

Native Call Queuing

The Native Call Queuing feature provides an enhanced capability to handle incoming calls to a hunt pilot number. Unified Communications Manager provides call queuing natively to users so that callers can be held in a queue until hunt members are available to answer them. Callers in a queue receive an initial greeting announcement followed by music on hold or tone on hold. If the caller remains in queue for a period of time, a secondary announcement is played at a configured interval until the call can be answered—or until the maximum wait timer expires.

Native Call Queuing Example

Unified Communications Manager cluster has four IP Phones: DN 1000, 1001, 1002, and 1003.

A hunt pilot (HP) 2000 is created with line group DN 1000 associated with it. So, this hunt pilot 2000 can only handle one call. Now, Check the “Queuing” enabled flag on the hunt pilot 2000 configuration page. Set the “Max Call Waiting Timer” to be 30 seconds, and select “Route the call to this destination” to be DN 1003. Ideally, if a caller has been put into that queue for 30 seconds, then it will be routed to DN 1003.

1. DN 1001 calls HP 2000, and 1000 answers the call.
2. DN 1002 calls HP 2000. Since the agent is busy, call is queued.
3. After 30 seconds, call is routed to DN 1003.
4. DN 1003 answers the call.

Field Names	CDR
globalCallID_callId	87029
origLegCallIdentifier	30117105
callingPartyNumber	1002
originalCalledPartyNumber	2000
wasCallQueued	1
totalWaitTimeInQueue	30

Normal Calls (Cisco Unified IP Phone to Cisco Unified IP Phone)

Normal calls log three records per call; one CDR and two CMRs, one for each endpoint. In the CDR, the “originalCalledPartyNumber” field contains the same Directory Number as the “finalCalledPartyNumber” field.

Successful Normal Calls CDR Examples

A successful call between two Cisco Unified IP Phones generates a single CDR at the end of the call.

1. The caller terminates a 60-second call. Because the calling party hangs up, the **orig_CauseValue** specifies 16 (Normal Clearing).

Field Names	CDR
globalCallID_callId	1
origLegCallIdentifier	100
destLegCallIdentifier	101
callingPartyNumber	2001
originalCalledPartyNumber	2309
finalCalledPartyNumber	2309
lastRedirectDn	2309
origCause_Value	16
dest_CauseValue	0
duration	60

2. The called party clears a 60-second call. Because the called party hangs up, the **dest_CauseValue** specifies 16 (Normal Clearing).

Field Names	CDR
globalCallID_callId	1
origLegCallIdentifier	100
destLegCallIdentifier	101
callingPartyNumber	2001
originalCalledPartyNumber	2309
finalCalledPartyNumber	2309
lastRedirectDn	2309
origCause_Value	0
dest_CauseValue	16
duration	60

Related Topics

Cisco Call Detail Records Field Descriptions, on page 127

Cisco Call Detail Records Codes, on page 171
Example Cisco Call Management Records, on page 209

Original Calling Party on Transfer

This feature changes the calling party number for a consultation call of a Cisco Unity or Cisco Unity Connection-initiated call transfer. The CDR of the consultation call shows that the original caller calls the transfer destination, not that the Cisco Unity or Cisco Unity Connection port calls the transfer destination.

You must configure this feature in the service parameters in Unified Communications Manager. See additional information at "Configuring CDR Service Parameters" section of the *CDR Analysis and Reporting Administration Guide*.

Original Calling Party on Transfer CDR Example

4001 calls 4002. 4002 transfers the call to 4003. The system generates three CDRs:

- The call between the original parties (4001 to 4002).
- The consultation call between the transferring party (4002) to the final transfer destination (4003).
- The call from the transferred party (4001) to the transfer destination (4003).

Call	CallingPartyNumber	originalCalledPartyNumber
1	4001	4002
2	4002	4003
3	4001	4003

Note No originalCallingParty field exists in the CDR.

Related Topics

Cisco Call Detail Records Field Descriptions, on page 127
Cisco Call Detail Records Codes, on page 171
Example Cisco Call Management Records, on page 209

Personal Assistant Calls

This section contains information about Personal Assistant Calls.

Related Topics

Personal Assistant Direct Call, on page 105
Personal Assistant Interceptor Going to Media Port and Transferring Call, on page 105
Personal Assistant Interceptor Going Directly to Destination, on page 106
Personal Assistant Interceptor Going to Multiple Destinations, on page 107

Personal Assistant Conferencing, on page 110

Personal Assistant Direct Call

A personal assistant direct call acts similar to the Blind Transfer from the Calling Party call type.

Personal Assistant Direct Call CDR Example

The following table contains an example CDR for this scenario:

- User A (2101) calls Personal Assistant route point (2000) and says “call User B.”
- The call transfers to User B (2105). In this case, User B did not configure any rules.

Note In the following example, 2000 represents the main personal assistant route point to reach personal assistant, 21XX represents the personal assistant interceptor route point, and 2001 - 2004 represents the media port.

In all cases, 2101 specifies the calling number.

Calling Party Num	OrigLegCall Identifier	Calling Party Number Partition	DestLeg Identifier	Final Called Party Num	Final Called Party Number Partition	Original Called Party Num	Original Called Party Number Partition	Last Redir DN	Last Redirect DN Partition	Duration (secs)
2101	16777217	PAManaged	16777219	2004	Phones	2000	1023970182	2000	Phones	34
2004	16777221	Phones	16777222	2105	PAManaged	2105	1023970182	2105	PAManaged	0
2101	16777217	PAManaged	16777222	2105	PAManaged	2105	1023970191	2105	PAManaged	5

Related Topics

Transferred Calls, on page 120

Personal Assistant Interceptor Going to Media Port and Transferring Call

This scenario acts similar to Blind Transfer from the Calling Party and Forwarded Calls actions.

Personal Assistant Interceptor Going to Media Port and Transferring the Call CDR Example

The following table contains an example CDR for this scenario:

- User A (2101) dials 2105.
- The personal assistant interceptor (21XX) picks up the call and redirects it to a media port (2002).
- Personal assistant processes the call according to the rules (if any) and transfers the call to the destination (2105), which has not configured any rules.

Calling Party Num	OrigLegCall Identifier	Calling Party Number Partition	DestLeg Identifier	Final Called Party Num	Final Called Party Number Partition	Original Called Party Num	Original Called Party Number Partition	Last Redir DN	Last Redirect DN Partition	Duration (secs)
2002	16777234	Phones	16777285	2105	PAManaged	2105	1023970478	2105	PAManaged	2
2101	16777230	PAManaged	16777232	2002	PA	2105	1023970478	21xx	" "	9
2105	16777235	PAManaged	16777230	2101	" "	" "	1023970483	" "	" "	5

Related Topics

Forwarded or Redirected Calls, on page 60
Transferred Calls, on page 120

Personal Assistant Interceptor Going Directly to Destination

This scenario can have two different cases: with rules and with no rules.

Example Personal Assistant Interceptor Going Directly to Destination with No Rules CDR

The following table contains an example CDR for this scenario:

- User A (2101) dials 2105.
- The personal assistant interceptor (21XX) picks up the call, processes it according to the rules (if any), and redirects the call to the destination (2105).

The following table contains an example CDR for this scenario:

Calling Party Number	OrigLegCall Identifier	Calling Party Number Partition	DestLeg Identifier	Final Called Party Number	FinalCalled Party Number Partition	Original Called Party Number	Original Called Party Number Partition	Last Redirect DN	Last Redirect DN Partition	Duration(secs)
2101	16777240	PAManaged	16777242	2105	PA	2105	1023970710	21XX	" "	8

Example Personal Assistant Going Directly to Destination with Rule to Forward Calls to Different Destination CDR

The following table contains an example CDR for this scenario:

- User A (2101) dials 2105.
- The Personal Assistant interceptor (21XX) picks up the call and processes it according to the rules.
- The Personal Assistant interceptor then redirects the call to the final destination (2110). In this case, 2105 configured a rule to forward the call to extension 2110.

Calling Party Number	OrigLegCall Identifier	Calling Party Number Partition	DestLeg Identifier	Final Called Party Number	FinalCalled Party Number Partition	Original Called Party Number	Original Called Party Number Partition	Last Redirect DN	Last Redirect DN Partition	Duration(secs)
2101	16777240	PAManaged	16777242	2110	PA	2105	1023970710	21XX	" "	8

Personal Assistant Interceptor Going to Multiple Destinations

This scenario can have several different cases. In each case, User B (2105) configures a rule to reach him at extension 2110 or 2120. This rule can activate when a caller calls Personal Assistant route point (2000) and says “call User B” (direct case) or when the caller dials User B (2105) directly (interceptor case).

Personal Assistant Interceptor Going to Multiple Destinations CDR Examples

The following sections contain examples of each case. The following tables contain example CDRs for each of these scenarios:

- Personal assistant direct multiple destinations: 2110 and 2120 (call accepted at first destination)
- Personal assistant direct multiple destinations: 2110 and 2120 (call accepted at second destination)
- Personal assistant direct multiple destinations: 2110 and 2120 (call accepted at third destination)
- Personal assistant intercept multiple destinations: 2110 and 2120 (call accepted at first destination)
- Personal assistant intercept multiple destinations: 2110 and 2120 (call accepted at second destination)
- Personal assistant intercept multiple destinations: 2110 and 2120 (call accepted at third destination)

Personal Assistant Direct Multiple Destinations: 2110 and 2120 (Call Accepted at First Destination)

- User A calls personal assistant and says, “call User B.”
- User B answers the call at 2110 extension.

Calling Party Num	OrigLegCall Identifier	Calling Party Number Partition	DestLeg Identifier	Final Called Party Num	Final Called Party Number Partition	Original Called Party Num	Original Called Party Number Partition	Last Redir DN	Last Redirect DN Partition	Duration (secs)
2004	16777262	Phones	16777263	2110	PAManaged	2110	1023971303	2110	PAManaged	6
2101	16777258	PAManaged	16777260	2004	Phones	2000	1023971303	2000	Phones	22
2110	16777263	PAManaged	16777258	2101	" "	" "	1023971312	" "	" "	9

Personal Assistant Direct Multiple Destinations: 2110 and 2120 (Call Accepted at Second Destination)

- User A calls personal assistant and says, “call User B.”

- User B answers the call at 2120 extension.

Calling Party Num	OrigLegCall Identifier	Calling Party Number Partition	DestLeg Identifier	Final Called Party Num	Final Called Party Number Partition	Original Called Party Num	Original Called Party Number Partition	Last Redir DN	Last Redirect DN Partition	Duration (secs)
2001	16777269	Phones	16777270	2110	PAManaged	2110	1023971456	2110	PAManaged	0
2001	16777272	Phones	16777273	2120	PAManaged	2120	1023971467	2120	PAManaged	4
2101	16777265	PAManaged	16777267	2001	Phones	2000	1023971467	2000	Phones	37
2120	16777273	PAManaged	16777265	2101	" "	" "	1023971474	" "	" "	7
2110	16777275	PAManaged	0	" "	" "	" "	1023971476	" "	" "	0

Personal Assistant Direct Multiple Destinations: 2110 and 2120 (Call Accepted at Third Destination)

- User A calls personal assistant and says, “call User B.”
- User B does not answer at either extension 2110 or 2120.
- Personal Assistant transfers the call to the original destination (2105), and User B then answers at that extension.

Note 2105 (the original destination) represents the third destination in this case.

Calling Party Num	OrigLegCall Identifier	Calling Party Number Partition	DestLeg Identifier	Final Called Party Num	Final Called Party Number Partition	Original Called Party Num	Original Called Party Number Partition	Last Redir DN	Last Redirect DN Partition	Duration (secs)
2002	16777281	Phones	16777282	2110	PAManaged	2110	1023971602	2110	PAManaged	0
2002	16777284	Phones	16777285	2120	PAManaged	2120	1023971615	2120	PAManaged	0
2101	16777277	PAManaged	16777279	2002	Phones	2000	1023971619	2000	Phones	38
2002	16777287	Phones	16777288	2105	PAManaged	2105	1023971619	2105	PAManaged	0
2101	16777277	PAManaged	16777288	2105	PAManaged	2105	1023971627	2105	PAManaged	7
2105	16777289	PAManaged	0	" "	" "	" "	1023971629	" "	" "	0

Personal Assistant Intercept Multiple Destinations: 2110 and 2120 (Call Accepted at First Destination)

- User A calls personal assistant and says, “call User B.”

- User B answers the call at extension 2110.

Calling Party Num	OrigLegCall Identifier	Calling Party Number Partition	DestLeg Identifier	Final Called Party Num	Final Called Party Number Partition	Original Called Party Num	Original Called Party Number Partition	Last Redir DN	Last Redirect DN Partition	Duration (secs)
2003	16777295	Phones	16777296	2110	PAManaged	2110	1023971740	2110	PAManaged	4
2101	16777291	PAManaged	16777293	2003	PA	2105	1023971740	21XX	" "	10
2110	16777296	PAManaged	16777291	2101	" "	" "	1023971749	" "	" "	9

Personal Assistant Intercept Multiple Destinations: 2110 and 2120 (Call Accepted at Second Destination)

- User A calls personal assistant and says, “call User B.”
- User B answers the call at extension 2120.

Calling Party Num	OrigLegCall Identifier	Calling Party Number Partition	DestLeg Identifier	Final Called Party Num	Final Called Party Number Partition	Original Called Party Num	Original Called Party Number Partition	Last Redir DN	Last Redirect DN Partition	Duration (secs)
2004	16777302	Phones	16777303	2110	PAManaged	2110	1023971815	2110	PAManaged	0
2004	16777305	Phones	16777306	2120	PAManaged	2120	1023971824	2120	PAManaged	3
2101	16777298	PAManaged	16777300	2004	PA	2105	1023971824	21XX	" "	22
2120	16777306	PAManaged	16777298	2101	" "	" "	1023971832	" "	" "	8

Personal Assistant Intercept Multiple Destinations: 2110 and 2120 (Call Accepted at Third Destination)

- User A calls personal assistant and says, “call User B.”
- User B does not answer at either extension 2110 or 2120.
- Personal assistant transfers the call to the original destination (2105), which User B then answers.

Note 2110 (the original destination) represents the third destination in this case.

Calling Party Num	OrigLegCall Identifier	Calling Party Number Partition	DestLeg Identifier	Final Called Party Num	Final Called Party Number Partition	Original Called Party Num	Original Called Party Number Partition	Last Redir DN	Last Redirect DN Partition	Duration (secs)
2001	16777312	Phones	16777313	2110	PAManaged	2110	1023971923	2110	PAManaged	0
2001	16777315	Phones	16777316	2120	PAManaged	2120	1023971936	2120	PAManaged	0
2101	16777308	PAManaged	16777310	2001	PA	2105	1023971940	21XX	" "	30
2001	16777318	Phones	16777319	2105	PAManaged	2105	1023971940	2105	PAManaged	0
2101	16777308	PAManaged	16777319	2105	PAManaged	2105	1023971953	2105	PAManaged	12

Personal Assistant Conferencing

Personal assistant conferencing acts similar to the ad hoc conferences call type.

Personal Assistant Conferencing CDR Example

The following table contains an example CDR for this scenario:

- User A calls personal assistant route point (2000) and says, “conference User B (2105) and User C (2110).”
- Personal assistant conferences User B and C into User A conference.

Calling Party Num	OrigLegCall Identifier	Calling Party Number Partition	DestLeg Identifier	Final Called Party Num	Final Called Party Number Partition
2003	16777345	Phones	16777346	2105	PAManaged
2101	16777340	PAManaged	16777342	2003	Phones
2003	16777350	Phones	16777351	2002	PAManaged
2003	16777342	Phones	16777347	2110	" "
2110	16777351	PAManaged	16777352	b00110201001	" "
2105	16777346	PAManaged	16777349	b00110201001	" "
2101	16777340	PAManaged	16777348	b00110201001	" "

This table continues with this additional information.

Original CalledParty Number	Original Called Party Number Partition	Last Redirect DN	Last Redirect DN Partition	Duration (seconds)
2105	1023972575	2105	PAManaged	6
2000	1023972576	2003	Phones	62
2110	1023972595	2110	PAManaged	39
b00110201001	1023972601	b00110201001	" "	25
b00110201001	1023972609	b00110201001	" "	14
b00110201001	1023972610	b00110201001	" "	34
b00110201001	1023972610	b00110201001	" "	34

Related Topics

Conference Calls, on page 49

Precedence Calls (MLPP)

Precedence calls take place the same as other calls except the precedence level fields get set in the CDR. Also, when a higher level precedence call preempts a call, the cause codes indicate the reason for the preemption.

Precedence Call CDR Examples

1. A call to another IP phone occurs by dialing a precedence pattern (precedence level 2).

Field Names	Precedence Call CDR
globalCallID_callId	100
origLegCallIdentifier	12345
destLegCallIdentifier	12346
callingPartyNumber	2001
origCalledPartyNumber	826001
origCause_Value	0
dest_CauseValue	16
origPrecedenceLevel	2
destPrecedenceLevel	2

2. A precedence call gets received from another network (precedence level 1).

Field Names	Precedence Call CDR
globalCallID_callId	102
origLegCallIdentifier	11111
destLegCallIdentifier	11112
callingPartyNumber	9728552001
origCalledPartyNumber	6001
origCause_Value	16
dest_CauseValue	0
origPrecedenceLevel	1
destPrecedenceLevel	1

3. A call gets preempted by a higher precedence level call.

Field Names	Original call CDR	Higher Level Call CDR
globalCallID_callId	10000	10001
origLegCallIdentifier	12345678	12345680
destLegCallIdentifier	12345679	12345681
callingPartyNumber	2001	9728551234
origCalledPartyNumber	826001	826001
origCause_Value	0	0
dest_CauseValue	9	16
origPrecedenceLevel	2	1
destPrecedenceLevel	2	1

Redirection (3xx) Calls

This example shows CDRs for a the redirection feature (3xx).

When a call is redirected by the Redirection Feature (3xx), the **origCalledPartyRedirectOnBehalfOf** and **lastRedirectRedirectOnBehalfOf** fields specify Unified CM Redirection = 19. The **origCalledPartyRedirectReason** and the **lastRedirectRedirectReason fields** specify Redirection = 162.

Redirection (3xx) CDR Example

Activate CFA on phone 10010 that is running SIP (registered to Unified Communications Manager) with a CFA destination of 10000. 35010 calls 10010, which is CFA to 10000. The call gets redirected from 10010 to 10000. 10000 answers the call and talks for 1 minute.

Field Names	Original Call CDR
globalCallID_callId	11
origLegCallIdentifier	21832023
destLegCallIdentifier	21832026
callingPartyNumber	35010
originalCalledPartyNumber	10010
finalCalledPartyNumber	10000
lastRedirectDn	10010
origCause_Value	0
dest_CauseValue	16
origCalledPartyRedirectReason	162
lastRedirectRedirectReason	162
origCalledPartyRedirectOnBehalfOf	19
lastRedirectRedirectOnBehalfOf	19
origTerminationOnBehalfOf	0
destTerminationOnBehalfOf	12
joinOnBehalfOf	19
duration	60

Redirecting Number Transformation

When the redirecting number transformation feature is enabled, original called and last redirecting number are transformed before it sends out in outgoing setup message.

Redirecting Number Transformation Example

1. CCM1 - Phone A [180000] , Phone B [180001] , Phone C [180002]
2. SIP trunk is configured on CCM1 pointing to SIP Gateway
3. Phone B has external mask set as +9111XXXX
4. Phone C has external mask set as +9122XXXX

On SIP trunk, redirecting party CSS is configured which has the partitions P1 and there is a Calling Party transformation pattern associated with P1. This pattern has external phone number mask enabled.

Scenario

A - calls Phone B ---- CFA - Phone C CFA --- SIP Trunk --- SIP Gateway

B - Original Called Party, C - Last Redirecting Party

There are 2 diversion headers corresponding to original and last redirecting party that is sent out in outgoing SIP INVITE message and these diversion headers have transformed redirecting number, that is, +91110001 and +91220002.

These values are also stored in CDR records. Transformed original called number will be stored in outpulsedOriginalCalledPartyNumber and transformed last redirecting number will be stored in outpulsedLastRedirectingNumber.

Field Names	CDR
globalCallID_callId	115010
origLegCallIdentifier	30751507
callingPartyNumber	180000
outpulsedCallingPartyNumber	880003
outpulsedOriginalCalledPartyNumber	+91110001
outpulsedLastRedirectingNumber	+91220002

Refer Calls

See the replaces calls topic for an example of Refer with Replaces.

Related Topics

Replaces Calls, on page 114

Replaces Calls

The examples show CDRs for various types of Replaces calls.

Replaces CDR Examples

1. **Invite with Replaces** – Phone 35010 that is running SIP calls phone 35020 that is running SIP. The transfer button gets pressed on 35010, and a call gets placed to SCCP phone 3000, 3000 answers the call; then, phone 35010 completes the transfer. The final transferred call occurs between 35020 and 3000.

Note When the transfer is complete, the system sends an Invite with Replaces to Unified Communications Manager.

Field Names	Original Call CDR	Reverted Call CDR
globalCallID_callId	5045247	5045248
origLegCallIdentifier	21822467	21822469
destLegCallIdentifier	21822468	21822468
callingPartyNumber	35010	35020
originalCalledPartyNumber	3000	3000
finalCalledPartyNumber	3000	3000
lastRedirectDn	3000	35010
origCause_Value	393216	0
dest_CauseValue	393216	16
origCalledPartyRedirectReason	0	0
lastRedirectRedirectReason	0	146
origCalledPartyRedirectOnBehalfOf	0	0
lastRedirectRedirectOnBehalfOf	0	18
origTerminationOnBehalfOf	18	0
destTerminationOnBehalfOf	18	12
joinOnBehalfOf	0	18
duration	5	60

2. **Refer with Replaces** – Phone 35010 that is running SIP calls SCCP 3000, the transfer button gets pressed on 35010, and a call is placed to SCCP 3001; 3001 answers the call; then, phone 35010 completes the transfer. The final transferred call occurs between 3000 and 3001.

Note When the transfer completes, a Refer with Replaces gets sent to Unified Communications Manager.

Field Names	Original Call CDR	Consultation Call CDR	Final Transferred Call CDR
globalCallID_callId	5045245	5045246	5045245
origLegCallIdentifier	21822461	21822463	21822462
destLegCallIdentifier	21822462	21822464	21822464
callingPartyNumber	35010	35010	3000

Field Names	Original Call CDR	Consultation Call CDR	Final Transferred Call CDR
originalCalledPartyNumber	3000	3001	3001
finalCalledPartyNumber	3000	3001	3001
lastRedirectDn	3000	3001	35010
origCause_Value	393216	393216	16
dest_CauseValue	393216	393216	0
origCalledPartyRedirectReason	0	0	130
lastRedirectRedirectReason	0	0	146
origCalledPartyRedirectOnBehalfOf	0	0	17
lastRedirectRedirectOnBehalfOf	0	0	18
origTerminationOnBehalfOf	17	18	12
destTerminationOnBehalfOf	17	18	17
joinOnBehalfOf	0	0	18
duration	25	4	25

RSVP

These fields identify the status of RSVP reservation for the call. Be aware that the Unified Communications Manager RSVP CDR status field value gets concatenated, and the system retains the last 32 status values for the call.

For example, if a call is established with “Optional” policy, and the initial RSVP reservation is successful, and then it subsequently loses its bandwidth reservation and then regains its bandwidth reservation after retry, for several times during middle of the call, the call ends with a successful RSVP reservation. The CDR shows the following string as the Unified Communication RSVP reservation status for that particular stream: “2:5:2:5:2:5:2” (success:lost_bw:success:lost_bw:success:lost_bw:success).

RSVP Call CDR Examples

1. The example represents a call that gets established with “Optional” policy, and the initial RSVP reservation succeeds. The parties talk for 5 minutes.

Field Names	CDR
globalCallID_callId	300
origLegCallIdentifier	16777300
destLegCallIdentifier	16777301

Field Names	CDR
callingPartyNumber	20000
origCalledPartyNumber	20001
finalCalledPartyNumber	20001
lastRedirectDn	20001
origCause_Value	0
dest_CauseValue	16
origDTMFMethod	2
destDTMFMethod	2
duration	300

2. The example represents a call that is established with “Optional” policy, and the initial RSVP reservation succeeds, then it loses its bandwidth reservation but regains it after a retry. The parties talk for 1 minute.

Field Names	CDR
globalCallID_callId	301
origLegCallIdentifier	16777302
destLegCallIdentifier	16777303
callingPartyNumber	20000
origCalledPartyNumber	20001
finalCalledPartyNumber	20001
lastRedirectDn	20001
origCause_Value	0
dest_CauseValue	16
origDTMFMethod	2:5:2
destDTMFMethod	2:5:2
duration	60

Secure Conference Meet-Me

The following example shows a CDR for a meet-me secure conference. 35010 calls into a secure meet-me conference, but 35010 is a non-secure phone. Because 35010 does not meet the minimum security level of

the meet-me conference, the call gets cleared with the cause code of 58 (meet-me conference minimum security level not met).

Secure Conference Meet-Me CDR Example

Field Names	Call to the Meet-Me Conference CDR
globalCallID_callId	5045247
origLegCallIdentifier	123456879
destLegCallIdentifier	123456999
callingPartyNumber	35010
originalCalledPartyNumber	50000
finalCalledPartyNumber	50000
lastRedirectDn	50000
origCause_Value	58
dest_CauseValue	0
origCalledPartyRedirectReason	0
lastRedirectRedirectReason	0
origCalledPartyRedirectOnBehalfOf	0
lastRedirectRedirectOnBehalfOf	0
origTerminationOnBehalfOf	6
destTerminationOnBehalfOf	6

Short Calls

A short call, with a **CdrLogCallsWithZeroDurationFlag** set at True and a duration of less than 1 second, appears as a zero duration call in the CDR. The **DateTimeConnect** field, which shows the actual connect time of the call, differentiates these calls from failed calls. For failed calls (which never connected), this value equals zero.

Short Calls CDR Example

The following table contains an example of a successful On Net call with a duration of less than 1 second that the called party cleared.

Calling Party	Calling Partition	Original Called Party	Original CalledPartition	Orig Cause	Dest Cause	DateTime Connect	Duration
2001	Accounts	2309	Marketing	0	16	973795815	0

SIP Call with URL in CallingPartyNumber Field

Calling and called parties can have SIP calls where the extension number is a URL. The extension number can use all printable ASCII characters. Do not leave any spaces in the URL. For example, extension "1000 1001" does not get accepted as a valid URL.

Note Printable ASCII characters represent characters with ASCII code (in decimal) from 33 to 126.

SIP Call with URL in CallingPartyNumber Field CDR Example

The SIP trunk of the Unified Communications Manager receives an incoming call. The call contains a SIP URL for the callingPartyNumber.

Field Names	Values
globalCallID_callId	1
origLegCallIdentifier	100
destLegCallIdentifier	101
callingPartyNumber	bob@abc.com
originalCalledPartyNumber	2309
finalCalledPartyNumber	2309
lastRedirectDn	2309
origCause_Value	16
dest_CauseValue	0
duration	60

Successful on Net Calls

A successful call between two Cisco Unified IP Phones generates a single CDR at the end of the call.

Successful On Net Call CDR Examples

The following table contains two examples:

- A—A 60-second call that the caller terminates
- B—A 60-second call that the called party clears

	Calling Party	Calling Partition	Original Called Party	Original Called Partition	Orig Cause	Dest Cause	Duration
A	2001	Accounts	2309	Marketing	16	0	60
B	2001	Accounts	2309	Marketing	0	16	60

Transferred Calls

Calls that are transferred generate multiple CDRs. One CDR exists for the original call, one for the consultation call, and another for the final transferred call.

For the original call, the **origCause_value** and **destCause_value** gets set to split = 393216, which indicates the call was split. The **origCallTerminationOnBehalfOf** and **destCallTerminationOnBehalfOf** fields get set to Transfer = 10 to indicate that this call was involved in a transfer.

For the consultation call, the **origCause_value** and **destCause_value** fields get set to split = 393216, which indicates that the call was split. The **origCallTerminationOnBehalfOf** and **destCallTerminationOnBehalfOf** fields get set to Transfer = 10 to indicate that this call was involved in a transfer.

For the final transferred call, the **joinOnBehalfOf** field gets set to Transfer = 10 to indicate that this call resulted from a transfer.

Transferred Calls CDR Examples

The following examples, which are not an exhaustive set, illustrate the records that would be generated under the stated circumstances. These examples help clarify what records are generated on transferred calls.

Blind Transfer From the Calling Party CDR Example

Call goes from extension 2001 to a PSTN number; they talk for 120 seconds. 2001 initiates a blind transfer to 2002. **CDR 1** (original call) shows a call from extension 2001 to a PSTN number, talking for 120 seconds. **CDR 2** (consultation call) shows a call from 2001 to extension 2002. **CDR 3** represents the final transferred call where 2001 completes the transfer, drops out of the call, and leaves a call between the PSTN and 2002.

Field Names	Original Call CDR	Consultation Call CDR	Final Transferred CDR
globalCallID_callId	1	2	1
origLegCallIdentifier	101	103	102
destLegCallIdentifier	102	104	104
callingPartyNumber	2001	2001	3071111
originalCalledPartyNumber	3071111	2002	2002
finalCalledPartyNumber	3071111	2002	2002
lastRedirectDn	3071111	2002	2001
origCause_Value	393216	393216	16

Field Names	Original Call CDR	Consultation Call CDR	Final Transferred CDR
dest_CauseValue	393216	393216	0
origTerminationOnBehalfOf	10	10	0
destTerminationOnBehalfOf	10	10	0
joinOnBehalfOf	0	0	10
duration	120	0	360

Consultation Transfer From the Calling Party CDR Example

Call goes from extension 2001 to a PSTN number; they talk for 60 seconds. 2001 initiates a consultation transfer to 2002 and talks for 10 seconds before the transfer completes. The final transferred call talks for 360 seconds. **CDR 1** (original call) shows a call from extension 2001 to a PSTN number, talking for 60 seconds. **CDR 2** (consultation call) shows a call from 2001 to extension 2002, talking for 10 seconds. **CDR 3** represents the final transferred call where 2001 completes the transfer, drops out of the call, and leaves a call between the PSTN and 2002.

Field Names	Original Call CDR	Consultation Call CDR	Final Transferred Call CDR
globalCallID_callId	1	2	1
origLegCallIdentifier	111	113	112
destLegCallIdentifier	112	114	114
callingPartyNumber	2001	2001	3071111
originalCalledPartyNumber	3071111	2002	2002
finalCalledPartyNumber	3071111	2002	2002
lastRedirectDn	50001	50001	2001
origCause_Value	393216	393216	16
dest_CauseValue	393216	393216	0
origTerminationOnBehalfOf	10	10	0
destTerminationOnBehalfOf	10	10	0
joinOnBehalfOf	0	0	10
duration	60	10	360

Blind Transfer From the Called Party CDR Example

Call goes from 50000 to 50001; they talk for 120 seconds. 50001 initiates a blind transfer to 50002. **CDR 1** (original call) shows a call from extension 50001 to 50002, talking for 120 seconds. **CDR 2** (consultation

call) shows a call from 50001 to extension 50002. **CDR 3** represents the final transferred call where 50001 completes the transfer, drops out of the call, and leaves a call between 50000 and 50002.

Field Names	Original Call CDR	Consultation Call CDR	Final Transferred Call CDR
globalCallID_callId	1	2	1
origLegCallIdentifier	200	202	200
destLegCallIdentifier	201	203	203
callingPartyNumber	50000	50001	50000
originalCalledPartyNumber	50001	50002	50002
finalCalledPartyNumber	50001	50002	50002
lastRedirectDn	50001	50001	50001
origCause_Value	393216	393216	16
dest_CauseValue	393216	393216	0
origTerminationOnBehalfOf	10	10	0
destTerminationOnBehalfOf	10	10	0
joinOnBehalfOf	0	0	10
duration	120	0	360

Consultation Transfer From the Called Party CDR Example

Call goes from 50000 to 50001; they talk for 120 seconds. 50000 initiates a blind transfer to 50002. **CDR 1** (original call) shows a call from extension 50000 to a 50001, talking for 120 seconds. **CDR 2** (consultation call) shows a call from 50000 to extension 50002. **CDR 3** represents the final transferred call where 50000 completes the transfer, drops out of the call, and leaves a call between 50001 and 50002.

Field Names	Original Call CDR	Consultation Call CDR	Final Transferred Call CDR
globalCallID_callId	1	2	1
origLegCallIdentifier	200	202	201
destLegCallIdentifier	201	203	203
callingPartyNumber	50000	50001	50000
originalCalledPartyNumber	50001	50002	50002
finalCalledPartyNumber	50001	50002	50002
lastRedirectDn	50001	50001	50001

Field Names	Original Call CDR	Consultation Call CDR	Final Transferred Call CDR
origCause_Value	393216	393216	16
dest_CauseValue	393216	393216	0
origTerminationOnBehalfOf	10	10	0
destTerminationOnBehalfOf	10	10	0
joinOnBehalfOf	0	0	10
duration	120	0	360

Video Calls

The following example shows a CDR for a video call.

Video Calls CDR Example

Calling party 51234 calls the called party 57890. In the following example, let 100 = H.261, 187962284 = 172.19.52.11, 288625580 = 172.19.52.17, 320 = 320K, and 2 = QCIF.

Field Names	Video Call CDR
globalCallID_callId	121
origLegCallIdentifier	101
destLegCallIdentifier	102
callingPartyNumber	51234
origCalledPartyNumber	57890
finalCalledPartyNumber	57890
lastRedirectDn	57890
origCause_Value	0
dest_CauseValue	16
origVideoCap_Codec	100
origVideoCap_Bandwidth	320
origVideoCap_Resolution	2
origVideoTransportAddress_IP	187962284
origVideoTransportAddress_Port	49208

Field Names	Video Call CDR
destVideoCap_Codec	100
destVideoCap_Bandwidth	320
destVideoCap_Resolution	2
destVideoTransportAddress_IP	288625580
destVideoTransportAddress_Port	49254

Video Conference Calls

Calls that are part of a video conference have multiple records logged. The number of CDR records that are generated depends on the number of parties in the video conference. One CDR record exists for each party in the video conference, one for the original placed call, one for each setup call that was used to join other parties to the video conference, and one for the last two parties that are connected in the video conference.

Therefore, for a three party ad hoc video conference, six CDR records exist:

- 1 record for the original call
- 3 records for the parties that connected to the conference
- 1 record for each setup call
- 1 record for the final two parties in the conference

You can associate the setup calls with the correct call leg in the conference by examining the calling leg ID and called leg ID.

The conference bridge device has special significance to the Unified Communications Manager, and calls to the conference bridge appear as calls to the conference bridge device. A special number in the form "b0019901001" shows the conference bridge port.

All calls in or out of the conference bridge get shown going into the conference bridge, regardless of the actual direction. By examining the setup call CDR records, you can determine the original direction of each call.

You can find the conference controller information in the comment field of the CDR. The format of this information follows:

Comment field = "ConfControllerDn=1000;ConfControllerDeviceName=SEP0003"

- The conference controller DN + conference controller device name uniquely identifies the conference controller. You need the device name in the case of shared lines.
- If the call is involved in multiple conference calls, the comment field will contain multiple conference controller information. This could happen in case the conference goes down to two parties and one of these parties starts another conference. If this is the case, the last conference controller information in the comment field will identify the conference controller.

  The call legs that are connected to the conference will have the following fields information:

  - The **finalCalledPartyNumber** field contains the conference bridge number "b0019901001".
  - The **origCalledPtyRedirectOnBehalfOf** field gets set to (Conference = 4).

- The **lastRedirectRedirectOnBehalfOf** field gets set to (Conference = 4).
- The **joinOnBehalfOf field** gets set to (Conference = 4).
- The comment field identifies the conference controller.
- The **destConversationId** field remains the same for all members in the conference. You can use this field to identify members of a conference call.

The original placed call and all setup calls that were used to join parties to the conference will have the following fields:

- The **origCallTerminationOnBehalfOf** field gets set to (Conference = 4).
- The **destCallTerminationOnBehalfOf field** gets set to (Conference = 4).

Video Conference Call CDR Example

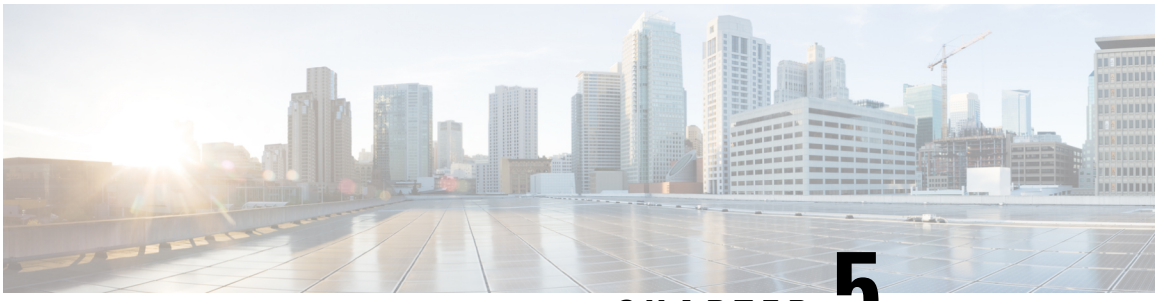
1. Call from 2001 to 2309; 2309 answers, and they talk for 60 seconds.
2. 2001 presses the conference softkey and dials 3071111.
3. 307111 answers and talks for 20 seconds; 2001 presses the conference softkey to complete the conference.
4. The three members of the conference talk for 360 seconds.
5. 3071111 hangs up; 2001 and 2309 stay in the conference. Because only two participants remain in the conference, the conference feature joins the two directly together, and they talk for another 55 seconds.

Note Each video conference call leg gets shown placing a call into the conference bridge. The call gets shown as a call into the bridge, regardless of the actual direction of the call.

FieldNames	Orig Call CDR	Setup Call CDR	Conference CDR 1	Conference CDR 2	Conference CDR 3	Final CDR
globalCallID_callId	1	2	1	1		1
origLegCallIdentifier	101	105	101	102	106	101
destLegCallIdentifier	102	106	115	116	117	102
callingPartyNumber	2001	2001	2001	2309	3071111	2001
originalCalledPartyNumber	2309	3071111	b0029901001	b0029901001	b0029901001	2309
finalCalledPartyNumber	2309	3071111	b0029901001	b0029901001	b0029901001	2309
lastRedirectDn	2001	3071111	b0029901001	b0029901001	b0029901001	b0029901001
origCause_Value	393216	0	16	393216	393216	16
dest_CauseValue	393216	0	393216	393216	393216	0

FieldNames	Orig Call CDR	Setup Call CDR	Conference CDR 1	Conference CDR 2	Conference CDR 3	Final CDR
origVideoCap_Codec	103	103	103	103	103	103
origVideoCap_Bandwidth	320	320	320	320	320	320
origVideoCap_Resolution	0	0	0	0	0	0
origVideoTransportAddress_IP	552953152	552953152	552953152	-822647488	-945658560	552953152
origVideoTransportAddress_Port	5445	5445	5445	5445	5445	5445
destVideoCap_Codec	103	103	103	103	103	103
destVideoCap_Bandwidth	320	320	320	320	320	320
destVideoCap_Resolution	0	0	0	0	0	0
destVideoTransportAddress_IP	-822647488	-945658560	-666216182	-666216182	-666216182	-822647488
destVideoTransportAddress_Port	5445	10002	10000	10004	10001	5445
origCalledPartyRedirectReason	0	0	0	0	0	0
lastRedirectRedirectReason	0	0	0	0	0	98
origTerminationOnBehalfOf	4	4	12	12	4	12
destTerminationOnBehalfOf	4	4	0	0	4	4
origCalledRedirectOnBehalfOf	0	0	4	4	4	0
lastRedirectRedirectOnBehalfOf	0	0	4	4	4	4
joinOnBehalfOf	0	0	4	4	4	4
Conversation ID	0	1		1	1	0
duration	60	360		360	360	55

Comment	
Orig Call CDR	
Setup Call CDR	**ConfControllerDn=2001;ConfControlerDeviceName=SEP0003E333FEBD**
Conference CDR 1	ConfControllerDn=2001;ConfControlerDeviceName=SEP0003E333FEBD
Conference CDR 2	ConfControllerDn=2001;ConfControlerDeviceName=SEP0003E333FEBD
Conference CDR 3	ConfControllerDn=2001;ConfControlerDeviceName=SEP0003E333FEBD
Final CDR	



CHAPTER 5

Cisco Call Detail Records Field Descriptions

This chapter defines all fields in the current CDRs in the order in which they appear in the CDR.

CDR Field Descriptions

The following table describes all fields in the current CDRs in the order in which they appear.

Table 9: CDR Field Descriptions

Field Name	Range of Values	Description
cdrRecordType	0, 1, 2	Defines the type of record. The following valid values apply: • 0—Start call detail record (not used) • 1—End call detail record (CDR) • 2—CMR record Default - For CDRs, this field always remains 1.
globalCallID_callManagerId	Positive Integer	Designates a unique Unified Communications Manager identity. The Global Call ID comprises two fields: globalCallID_callId globalCallID_callManagerId. All records that are associated with a standard call have the same Global Call ID in them. Default - Ensure that this field is populated.

Field Name	Range of Values	Description
globalCallID_callId	Positive Integer	

Field Name	Range of Values	Description
		Designates a unique call identity value that is assigned to each call. The system allocates this identifier independently on each call server. Values get chosen sequentially when a call begins. A value gets assigned for each call, successful or unsuccessful. When Unified Communications Manager restarts, it checks the file for the current globalCallID_callId number and assigns the next 1000th number to the next GlobalCallID_callId. The Global Call ID consists of two fields: globalCallID_callId globalCallID_callManagerId. All records that are associated with a standard call have the same Global Call ID in them. **Note** For Unified Communications Manager Release 5.x and later releases, the value in the GlobalCallId CDR field survives over Unified Communications Manager restarts. In Release 4.x and earlier releases, although the GlobalCallId field is time-based, the field gets reused under conditions of heavy traffic. Because of this behavior, problems can occur with customer billing applications and the ability of CAR to correlate CMRs with CDRs and to correlate conference call CDRs. For Release 5.x and later releases, GlobalCallId redesign ensures that the field retains a unique value, at least for a certain number of days.

Field Name	Range of Values	Description
		Now, the last used globalCallId_callId value gets written to disk periodically (for every x number of calls). The value gets retrieved after a Unified Communications Manager restart, and the new globalCallId_callId value begins with this number plus x. Default - Ensure that this field is populated.
origLegCallIdentifier	Positive Integer	Identifies the originating leg of a call. Be aware that this value is unique within a cluster. If the leg of a call persists across several subcalls and CDRs (as during a call transfer), this value remains constant. Default - Ensure that this field is populated.
dateTimeOrigination	Integer	Identifies the date and time when the user goes off the hook or the date and time that the H.323 SETUP message is received for an incoming call. The time gets stored as UTC. Default - Ensure that this field is populated.
origNodeId	Positive Integer	Identifies the server, or node within a cluster, to which the originator of the call is registered at the time that the call is made. Default - Ensure that this field is populated.

Field Name	Range of Values	Description
origSpan	0, Positive Integer	For calls that originate at a gateway, this field indicates the B-channel number of the T1, PRI, or BRI trunk where the call originates, or a zero value for FXS or FXO trunks. For H.323 gateways, the span number remains unknown, and this field contains the call leg ID of the originator. For calls that did not originate at a gateway, the value specifies zero. Default - This field gets populated based on these rules.
origIpAddr	Integer	Identifies the v4 IP address of the device that originates the call signaling. For Cisco Unified IP Phones, this field specifies the v4 address of the phone. For PSTN calls, this field specifies the v4 address of the H.323 gateway. For intercluster calls, this field specifies the v4 address of the remote Unified Communications Manager. Default - 0. If the v4 address does not exist for the originating device, this field equals 0. This field gets populated based on these rules.

Field Name	Range of Values	Description
callingPartyNumber	Text String	Specifies a numeric string of up to 25 characters that indicates the calling party number if the calling party is identified with a directory number. If the calling party uses a blended address in the identity headers, this field contains the directory number portion of the blended address. For calls that originate at a Cisco Unified IP Phone, this field shows the extension number of the line that is used. For incoming H.323 calls, this field specifies the value that is received in the Calling Party Number field in the Setup message. This field reflects any translations that are applied to the Calling Party Number before it arrives at the Unified Communications Manager (such as translations at the gateway). For the server calls, where Unified Communications Manager originates a half call without a calling party, this field may remain empty. CallingPartyNumber could contain a SIP URI. Default - This field gets populated based on these rules.
callingPartyUnicodeLoginUserID	Unicode – UTF_8	Specifies the calling party login user ID. The format of this field specifies UTF_8. Default - Empty string " ". If the user ID does not exist, this field stays empty.

Field Name	Range of Values	Description
origCause_location	0 to 15	Specifies the Location field that is indicated in the ISDN release message for clearing causes that are received over ISDN signaling links. See topics that are related to call termination cause codes for a list of the valid values per Q.850. For clearing causes that are created internally by the Unified Communications Manager, this value specifies zero. Default - 0
origCause_value	0 to 129	Reflects the reason for clearance for the calls that are cleared by the originating party. Unified Communications Managercurrently uses the Q.850 codes and some Unified Communications Managerdefined codes. See topics that are related to call termination cause codes for a listing. For calls that are cleared by the terminating party, this field specifies zero. In addition to the standard values that are described in Q.850, when a call is split by a feature (transfer or conference), the CDR terminates, and this field gets set to 393216. This represents a proprietary value for this field. Default - 0

Field Name	Range of Values	Description
origPrecedenceLevel	0 to 4	Represents the precedence level of the original leg. For MLPP, each call leg includes a precedence level. • Precedence 0 = FLASH OVERRIDE/ EXECUTIVE OVERRIDE • Precedence 1 = FLASH • Precedence 2 = IMMEDIATE • Precedence 3 = PRIORITY • Precedence 4 = ROUTINE Default - 4
origMediaTransportAddress_IP	0, Integer	Identifies the v4 IP address of the device that originates the media for the call. For Cisco Unified IP Phones, this field specifies the v4 address of the phone. For PSTN calls, this field specifies the v4 address of the H.323 gateway. For intercluster calls, this field specifies the v4 address of the remote phone. Default - 0. If media is not established or the address is not v4, this field equals 0.
origMediaTransportAddress_Port	0, Positive Integer	Identifies the IP port number that is associated with the OrigMediaTransportAddress_IP field. Default - 0. If media is not established, this field stays 0.

Field Name	Range of Values	Description
origMediaCap_payloadCapability	0, Positive Integer	Identifies the codec type that the originator uses to transmit media. Unified Communications Manager currently uses the following payload capability values: 0, 1-16, 18-20, 25, 32, 33, 81-86. See topics related to codec types for a listing of the valid values. Default - 0. If media is not established, this field stays 0.
origMediaCap_maxFramesPerPacket	0, Positive Integer	Identifies the number of milliseconds of data per packet that the originating party sends. This field normally gets set to 10, 20, or 30 for G.729 or G.711 codecs, but the field can store any nonzero value. Default - 0. If media is not established, this field stays 0.
origMediaCap_g723BitRate	0	This field is not used in the current release of Unified Communications Manager. Default - This field will remain 0.
origVideoCap_Codec	0, 100 = H.261, 101 = H.263, 103 = H.264	Identifies the codec type that the originator uses to transmit video (H.261, H.263, or H.264.) Default - 0. If media is not established, this field stays 0.
origVideoCap_Bandwidth	0, Positive Integer	Identifies the bandwidth that is measured in units of kbps. Default - 0. If media is not established, this field stays 0.

Field Name	Range of Values	Description
origVideoCap_Resolution	0, 1 = SQCIF, 2 = QCIF, 3 = CIF, 4 = CIF4, 5 = CIF16 6 = H263 custom resolution 7 = W360P 8 = VGA 9 = W448P 10 = HD720P 11 = HD1080P 12 = CIF2	Indicates the transmitting resolution. In the case of H.264 codec or SIP device, this field refers to the max transmitting resolution the device can transmit for this call. Default - 0. If media is not established, this field stays 0.
origVideoTransportAddress_IP	0, Integer	Identifies the v4 IP address of the device that originates the call. Default - 0. If media is not established or the address is not v4, this field stays 0.
origVideoTransportAddress_Port	0, Positive Integer	Identifies the video RTP port that is associated with the origVideoTransportAddress_IP field. Default - 0. If media is not established, this field stays 0.

Field Name	Range of Values	Description
origRSVPAudioStat	0 to 5	Provides the status of the RSVP audio reservation from originator to terminator. 0 – No reservation. 1 – RSVP Reservation Failure condition at call setup or feature invocation. 2 – RSVP Reservation Success condition at the call setup or feature invocation. 3 – RSVP Reservation No Response (RSVP Agent) condition at the call setup or feature invocation. 4 – RSVP Mid Call Failure Preempted condition (preempted after the call setup). 5 – RSVP Mid Call Failure Lost Bandwidth condition (includes all mid-call failures except MLPP preemption). Default – 0

Field Name	Range of Values	Description
origRSVPVideoStat	0 to 5	Provides the status of the RSVP video reservation from originator to terminator. 0 – No reservation. 1 – RSVP Reservation Failure condition at call setup or feature invocation. 2 – RSVP Reservation Success condition at call setup or feature invocation. 3 – RSVP Reservation No Response (RSVP Agent) condition at call setup or feature invocation. 4 – RSVP MID Call Failure Preempted condition (preempted after call setup). 5 – RSVP MID Call Failure Lost Bandwidth condition (includes all mid-call failures except MLPP preemption). Default – 0
destLegCallIdentifier	0, Positive Integer	Identifies the terminating leg of a call. This value remains unique within a cluster. If the leg of a call persists across several sub-calls and, consequently, several CDRs (as during a call transfer), this value remains constant. Default - 0. If the destination cannot be reached, this field stays 0.
destNodeId	0, Positive Integer	Identifies the location, or node within a cluster, to which the terminating party of the call is registered at the time that the call is made. Default - 0. If the destination cannot be reached, this field stays 0.

Field Name	Range of Values	Description
destSpan	0, Positive integer	For calls that are received at a gateway, this field indicates the B channel number of the T1, PRI, or BRI trunk where the call is received, or a zero value for FXS or FXO trunks. For H.323 gateways, the span number remains unknown, and this field contains the call leg ID of the destination. For calls not terminating at a gateway, the value specifies zero. Default - 0. If the destination cannot be reached, this field stays 0.
destIpAddr	0, Integer	Identifies the v4 IP address of the device that terminates the call signaling. For Cisco Unified IP Phones, this field specifies the v4 address of the phone. For PSTN calls, this field specifies the v4 address of the H.323 gateway. For intercluster calls, this field specifies the v4 address of the remote Unified Communications Manager. Default - 0. If the destination cannot be reached, this field stays 0. If the v4 address does not exist for this device, the field equals 0.

Field Name	Range of Values	Description
originalCalledPartyNumber	Text String	Specifies the number to which the original call was presented, prior to any call forwarding. If translation rules are configured, this number reflects the called number after the translations have been applied. If a blended address is used for the called party, this field specifies the directory number portion of the blended address. This field represents a numeric string of up to 48 characters that can be either digits or a SIP URL. Default - Empty string " ". If destination cannot be reached, or if the called party number is a directory URI, this field stays empty.
finalCalledPartyNumber	Text String	Specifies the phone number to which the call finally gets presented, until it is answered or rings out. If no forwarding occurs, this number shows the same number as the originalCalledPartyNumber. If the call finally gets presented to a directory URI, the field remains empty. If a blended address is used, this field specifies the directory number portion of the blended address. For calls to a conference bridge, this field contains the actual identifier of the conference bridge, which is an alphanumeric string (for example, b0019901001). This field represents an alphanumeric string that can be either digits or a SIP URL. Default - Empty string " ". If destination cannot be reached, this field stays empty.

Field Name	Range of Values	Description
finalCalledPartyUnicodeLoginUserID	Unicode – UTF_8	Specifies the login user ID. The format of this field specifies UTF_8. Default - Empty string " ". If the user ID does not exist, this field stays empty.
destCause_location	0 to 15	For clearing causes that are received over ISDN signaling links, the ISDN release message indicates this location field. See topics that are related to call termination cause codes for a listing of the valid values per Q.850. For clearing causes that Unified Communications Manager creates internally, this value equals zero. Default - 0. If the destination cannot be reached, this field stays 0.
destCause_value	0 to 129	Reflects the reason for the calss that the destination party cleared. See topics that are related to call termination cause codes for a listing of the valid values per Q.850. For calls that the originating party clears, this field stays zero. In addition to the standard values that are described in Q.850, when a call gets split by a feature (transfer or conference), the CDR terminates, and this field gets set to 393216. This represents a proprietary value for this field. Default - 0. If the destination cannot be reached, this field stays 0.

Field Name	Range of Values	Description
destPrecedenceLevel	0 to 4	Represents the destination legs precedence level. For MLPP, each call leg has a precedence level. • Precedence 0 = FLASH OVERRIDE • Precedence 1 = FLASH • Precedence 2 = IMMEDIATE • Precedence 3 = PRIORITY • Precedence 4 = ROUTINE Default - 4
destMediaTransportAddress_IP	0, Integer	Identifies the v4 IP address of the device that terminates the media for the call. For Cisco Unified IP Phones, this field designates the v4 address of the phone. For PSTN calls, this field designates the v4 address of the H.323 gateway. For intercluster calls, this field shows the v4 address of the remote phone. Default - 0. If the destination cannot be reached or the IP address of the destination is not v4, this field stays 0.
destMediaTransportAddress_Port	0, Positive Integer	Identifies the IP port number that is associated with the DestMediaTransportAddress_IP field. Default - 0. If the destination cannot be reached, this field stays 0.

Field Name	Range of Values	Description
destMediaCap_payloadCapability	0, Positive Integer	Identifies the codec type that the terminating party uses to transmit media. Unified Communications Manager currently uses the following payload capability values: 0, 1-16, 18-20, 25, 32, 33, 81-86. See topics related to codec types for a listing of the valid values. Default - 0. If the destination cannot be reached, this field stays 0.
destMediaCap_maxFramesPerPacket	0, Positive Integer	Identifies the number of milliseconds of data per packet that the terminating party of the call sends. This field normally gets set to 10, 20, or 30 for G.729 or G.711 codecs but can store any nonzero value. This field can specify zero if the media is never established. Default - 0. If the destination cannot be reached, this field stays 0.
destMediaCap_g723BitRate	0	This field is not used in the current release of Unified Communications Manager. Default - This field stays 0.
destVideoCap_Codec	0, 100 = H.261, 101 = H.263, 103 = H.264	Identifies the codec type that the terminating party uses to transmit video (H.261, H.263, or H.264). Default - 0. If the destination cannot be reached, this field stays 0.
destVideoCap_Bandwidth	0, Positive Integer	Identifies the bandwidth, and is measured in units of kbps. Default - 0. If the destination cannot be reached, this field stays 0.

Field Name	Range of Values	Description
destVideoCap_Resolution	0, 1 = SQCIF, 2 = QCIF, 3 = CIF, 4 = CIF4, 5 = CIF16 6 = H263 custom resolution 7 = W360P 8 = VGA 9 = W448P 10 = HD720P 11 = HD1080P 12 = CIF2	Indicates the transmitting resolution. In the case of H.264 codec or SIP device, this field refers to the max transmitting resolution the device can transmit for this call. Default - 0. If media is not established, this field stays 0.
destVideoTransportAddress _IP	0, Integer	Identifies the v4 IP address of the device that receives the call. Default - 0. If the destination cannot be reached or the IP address of the destination is not v4, this field stays 0.
destVideoTransportAddress_Port	0, Positive Integer	Identifies the video RTP port that is associated with the destVideoTransportAddress_IP field. Default - 0. If the destination cannot be reached, this field stays 0.

Field Name	Range of Values	Description
destRSVPAudioStat	0 - 5	Designates the status of the RSVP audio reservation from terminator to originator. 0 – No reservation. 1 – RSVP Reservation Failure condition at the call setup or feature invocation. 2 – RSVP Reservation Success condition at call setup or feature invocation. 3 – RSVP Reservation No Response (RSVP Agent) condition at call setup or feature invocation. 4 – RSVP Mid Call Failure Preempted condition (preempted after call setup). 5 – RSVP Mid Call Failure Lost Bandwidth condition (includes all mid call failures except MLPP preemption). Default – 0

Field Name	Range of Values	Description
destRSVPVideoStat	0 - 5	Designates the status of the RSVP video reservation from terminator to originator. 0 – No reservation. 1 – RSVP Reservation Failure condition at call setup or feature invocation. 2 – RSVP Reservation Success condition at call setup or feature invocation. 3 – RSVP Reservation No Response (RSVP Agent) condition at call setup or feature invocation. 4 – RSVP Mid Call Failure Preempted condition (preempted after call setup). 5 – RSVP Mid Call Failure Lost Bandwidth condition (includes all mid call failures except MLPP preemption). Default – 0
dateTimeConnect	0, Integer	Identifies the date and time that the call connects. The time gets stored as UTC. If the call is never answered, this value shows zero. Default - 0. If the call is never connected, this field stays 0.
dateTimeDisconnect	Integer	Identifies the date and time when the call is cleared. This field gets set even if the call never connects. The time gets stored as UTC. Default - Ensure that this field is populated.

Field Name	Range of Values	Description
lastRedirectDn	Text String	Specifies a numeric string of up to 25 characters. The numeric string can contain digits or a SIP URL. For forwarded calls, this field specifies the phone number of the next to last hop before the call reaches its final destination. If only one hop occurs, this number matches the OriginalCalledPartyNumber. If a blended address is used for call addressing, this field contains only the directory number portion of the blended address. For calls that are not forwarded, this field matches the OriginalCalledPartyNumber and the FinalCalledPartyNumber. For calls to a conference bridge, this field contains the actual identifier of the conference bridge, which is an alphanumeric string (for example, b0019901001). Default - Empty string " ". If the call is never redirected, or if the next to last hop address is a directory URI, this field remains empty.
pkid	Text String	Identifies a text string that the database uses internally to uniquely identify each row. This text string provides no meaning to the call itself. Default - A unique ID should always populate this field.

Field Name	Range of Values	Description
originalCalledPartyNumberPartition	Text String	Identifies unique partition name that is associated with the OriginalCalledPartyNumber field because Unified Communications Manager supports multiple Cisco Unified IP Phones with the same extension number in different partitions. For calls that egress through an H.323 gateway, this field uniquely specifies the partition name that is associated with the route pattern that points to the gateway. Default - Empty string " ". If the original called party does not have a partition, this field remains empty.
callingPartyNumberPartition	Text String	Identifies unique partition name that is associated with the CallingPartyNumber field because Unified Communications Manager supports multiple Cisco Unified IP Phones with the same extension number in different partitions. For calls that ingress through an H.323 gateway, this field remains blank. Default - Empty string " ". If the original called party does not have a partition, this field remains empty.

Field Name	Range of Values	Description
finalCalledPartyNumberPartition	Text String	Identifies unique partition name that is associated with the FinalCalledPartyNumber field because Unified Communications Manager supports multiple Cisco Unified IP Phones with the same extension number in different partitions. For calls that egress through an H.323 gateway, this field uniquely specifies the partition name that is associated with the route pattern that points to the gateway. Default - Empty string " ". If the final called party does not have a partition, this field remains empty.
lastRedirectDnPartition	Text String	Identifies unique partition name that is associated with the LastRedirectDn field because Unified Communications Manager supports multiple Cisco Unified IP Phones with the same extension number in different partitions. For calls that egress through an H.323 gateway, this field specifies the partition name that is associated with the route pattern that points to the gateway. Default - Empty string " ". If the last redirecting Party does not have a partition or the call was never redirected, this field stays empty.
duration	0, Positive integer	Identifies the difference between the Connect Time and Disconnect Time. This field specifies the time that the call remains connected, in seconds. This field remains zero if the call never connects or if it connects for less than 1 second. Default - 0

Field Name	Range of Values	Description
origDeviceName	Text String	Specifies the text string that identifies the name of the originating device. Default - Ensure that this field is populated.
destDeviceName	Text String	Specifies the text string that identifies the name of the destination device. Default - Empty string“ ”. If the original device does not have a name, this field stays empty.
origCallTerminationOnBehalfOf	0, Positive Integer	Specifies code that identifies why the originator was terminated. For example, if the originator of the call hangs up the phone, the OnBehalfOf code shows “12” for Device. If the call terminates because of a transfer, the OnBehalfOf code shows “10” for Transfer. See topics related to CDR field descriptions for a list of the codes. This release added new OnBehalfOf codes. Default - 0
destCallTerminationOnBehalfOf	0, Positive Integer	Specifies code that identifies why the destination was terminated. For example, if the destination of the call hangs up the phone, the OnBehalfOf code shows “12” for Device. If the call terminates because of a transfer, the OnBehalfOf code shows “10” for Transfer. See topics related to CDR field descriptions for a list of the codes. This release added new OnBehalfOf codes. Default - 0

Field Name	Range of Values	Description
origCalledPartyRedirectOnBehalfOf	0, Positive Integer	Specifies code that identifies the reason for redirection of the original called party. For example, if the original called party was redirected because of a conference, the OnBehalfOf code specifies “4.” See topics related to CDR field descriptions for a list of the codes. This release added new OnBehalfOf codes. Default - 0
lastRedirectRedirectOnBehalfOf	0, Integer	Specifies code that identifies the reason for redirection of the last redirected party. For example, if the last redirected party was redirected on behalf of a conference, the OnBehalfOf code specifies “4.” See topics related to CDR field descriptions for a list of the codes. This release added new OnBehalfOf codes. Default - 0
origCalledPartyRedirectReason	0, Integer	Identifies the reason for a redirect of the original called party. See topics related to redirect reason codes for a complete list of the codes. Default - 0
lastRedirectRedirectReason	0, Integer	Identifies the last redirect reason for redirection. See topics related to redirect reason codes for a complete list of the codes. Default - 0

Field Name	Range of Values	Description
destConversationID	0, Integer	Specifies a unique identifier that is used to identify the parties of a conference call. For conference chaining scenarios, the origConversationID and destConversationID fields identify which conferences are chained together. Default - 0
globalCallId_ClusterId	Text String	Specifies a unique ID that identifies a cluster of Unified Communications Managers. The field is generated at installation and is not used by Unified Communications Manager. The fields globalCallId_ClusterId + globalCallId_CMId + globalCallId_CallId make up this unique key. Default - This field should always be populated.
joinOnBehalfOf	0, Integer	Specifies code that identifies the reason for a join. For example, if the join takes place on behalf of a transfer, the OnBehalfOf code specifies “10.” See topics related to CDR field descriptions for a list of the codes. Default - 0
comment	Text String	Allows features to add text to the CDRs. This text can describe details about the call. For example, the following field flags malicious calls: Tag—CallFlag Value—MALICIOUS Default - Empty string “ ”.
authCodeDescription	Text String	Provides a description of the FAC. Default - Empty string “ ” or null.

Field Name	Range of Values	Description
authorizationLevel	0, Integer	Displays the level of the FAC. Default - 0
clientMatterCode	Text String	Displays the client matter code. Before the system extends a call, the user enters a client matter code that can be used for assigning account or billing codes to calls. Default - Empty string " " or null.
origDTMFMethod	0, Positive Integer	Displays the DTMF method that the originator uses. 0 - No DTMF - Use ANY matched DTMF. 1 - OOB - Use OOB if endpoints behind SIPTrunk support it. 2 - 2833 - Use RFC2833 if endpoints behind SIPTrunk support it. 3 - OOB and 2833 - Use both KPML and RFC2833 if endpoints behind SIPTrunk can support both. 4 - Unknown Default - 0 (No preference)
destDTMFMethod	0, Positive Integer	Displays the DTMF method that the destination uses. 0 - No DTMF - Use ANY matched DTMF.1 - OOB - Use OOB if endpoints behind SIPTrunk support it.2 - 2833 - Use RFC2833 if endpoints behind SIPTrunk support it.3 - OOB and 2833 - Use both KPML and RFC2833 if endpoints behind SIPTrunk can support both.4 - Unknown. Default - 0 (No preference)

Field Name	Range of Values	Description
callSecuredStatus	0, Positive Integer	Displays the highest security status that is reached during a call. For example, if the call is originally unsecured, and later the call changes to secured, the CDR contains 1 for "Secured" even though different portions of the call have different status values. 0 - Non-secured 1 - Authenticated (not encrypted) 2 - Secured (encrypted) Default - 0 (Non-secured)
origConversationID	Integer	Identifies the conference ID that is associated with the originating leg of the call. In most cases, this field equals 0. For conference chaining scenarios, the origConversationID and destConversationID fields identify which conferences are chained together. Default - 0
origMediaCap_Bandwidth	0, Positive Integer	Displays the media bandwidth that is used at the origination of the call. Default - 0
destMediaCap_Bandwidth	0, Positive Integer	Displays the media bandwidth that is used at the destination of the call. Default - 0
authorizationCodeValue	Text String	Displays the Forced Authorization Code (FAC) that is associated with the call. Default - Empty string " " or null.
outpulsedCallingPartyNumber	Text String	Comprises an alphanumeric string of up to 50 characters. The calling party number gets outpulsed from the device. This field gets populated only when normalization or localization takes place at the device. Default - Empty string " " or null.

Field Name	Range of Values	Description
outpulsedCalledPartyNumber	Text String	Comprises an alphanumeric string of up to 50 characters. The called party number gets outpulsed from the device. This field gets populated only when normalization or localization takes place at the device. Default - Empty string “ ”or null.
origIpv4v6Addr	Text string	Comprises an alphanumeric string of up to 64 characters. This field identifies the IP address of the device that originates the call signalling. The field can be either IPv4 or IPv6 format depending on the type of IP address that gets used for the call. For Cisco Unified IP Phones, this field is the address of the Cisco Unified IP Phone. For PSTN calls, this field is the address of the gateway. For intercluster calls, this field is the address of the remote Unified Communications Manager. The IP address is either in dotted decimal format or in colon separated hexadecimal format. Default - The IP address of the originating device as reported by the device or used for the call after media negotiation.

Field Name	Range of Values	Description
destIpv4v6Addr	Text string	Comprises an alphanumeric string of up to 64 characters. This field identifies the IP address of the device that terminates the call signalling. The field can be either in IPv4 or IPv6 format depending upon the type of IP address that gets used for the call. For Cisco Unified IP Phones, this field is the address of the Cisco Unified IP Phone. For PSTN calls, this field is the address of the gateway. For intercluster calls, this field is the address of the remote Unified Communications Manager. The IP address is either in dotted decimal format or in colon separated hexadecimal format. Default - Empty String " " or null. If the destination does not get reached, this field stays empty.
origVideoCap_Codec_Channel2	0, 100 = H.261, 101 = H.263, 103 = H.264,	Identifies the codec type that the originator uses to transmit video (H.261, H.263, or H.264) for the second video channel. Default - 0. If media does not get established, this field displays 0. Also, if H.239 and BFCP are not supported for this call, this field displays 0.
origVideoCap_Bandwidth_Channel2	0, Positive integer	Identifies the bandwidth measured in units of kbps for the second video channel. Default - 0. If media does not get established, this field displays 0. Also, if H.239 and BFCP are not supported for this call, this field displays 0.

Field Name	Range of Values	Description
origVideoCap_Resolution_Channel2	0, 1 = SQCIF, 2 = QCIF, 3 = CIF, 4 = CIF4, 5 = CIF16 6 = H263 custom resolution 7 = W360P 8 = VGA 9 = W448P 10 = HD720P 11 = HD1080P 12 = CIF2	Indicates the transmitting resolution for the second video channel. In the case of H.264 codec or SIP device, this field refers to the maximum transmitting resolution the device can transmit for this call. Default - 0. If media is not established, this field stays 0. Also, if H.239 and BFCP are not supported for this call, this field displays 0.
origVideoTransportAddress_IP_Channel2	0, Integer	Identifies the v4 IP address of the device that originates the call for the second video channel. Default - 0. If media does not get established, this field displays 0. Also, if H.239 and BFCP are not supported for this call, this field displays 0.
origVideoTransportAddress_Port_Channel2	0, Positive integer	Identifies the video RTP port associated with the origH239VideoTransportAddress_IP field for the second video channel. Default - 0. If media does not get established, this field displays 0. Also, if H.239 and BFCP are not supported for this call, this field displays 0.
origVideoChannel_Role_Channel2	0 = Presentation role, 1 = Live role, Positive integer	Identifies the H.239 video channel role of the device that originates. Default - 0. If media does not get established, this field displays 0. Also, if H.239 is not supported, this field displays 0.

Field Name	Range of Values	Description
destVideoCap_Codec_Channel2	0, 100 = H.261, 101 = H.263, 103 = H.264	Identifies the codec type that the terminating party uses to transmit video (H.261, H.263, or H.264) for the second video channel. Default - 0. If the destination cannot be reached, this field stays 0. Also, if H.239 and BFCP are not supported for this call, this field displays 0.
destVideoCap_Bandwidth_Channel2	0, Positive integer	Identifies the bandwidth measured in units of kbps for the second video channel. Default - 0. If media does not get established, this field displays 0. Also, if H.239 and BFCP are not supported for this call, this field displays 0.
destVideoCap_Resolution_Channel2	0, 1 = SQCIF, 2 = QCIF, 3 = CIF, 4 = CIF4, 5 = CIF16 6 = H263 custom resolution 7 = W360P 8 = VGA 9 = W448P 10 = HD720P 11 = HD1080P 12 = CIF2	Indicates the transmitting resolution for the second video channel. In the case of H.264 codec or SIP device, this field refers to the maximum transmitting resolution the device can transmit for this call. Default - 0. If media is not established, this field stays 0. Also, if H.239 and BFCP are not supported for this call, this field displays 0.
destVideoTransportAddress_IP_Channel2	0, Integer	Identifies the v4 IP address of the device that receives the call. Default - 0. If media does not get established, this field displays 0. Also, if H.239 and BFCP are not supported for this call, this field displays 0.

Field Name	Range of Values	Description
destVideoTransportAddress_Port_Channel2	0, Positive integer	Identifies the video RTP port associated with the destH239VideoTransportAddress_IP field. Default - 0. If media does not get established, this field displays 0. Also, if H.239 and BFCP are not supported for this call, this field displays 0.
destVideoChannel_Role_Channel2	0 = Presentation role, 1 = Live role, Positive integer	Identifies the H.239 video channel role of the device that receives the call. Default - 0. If media does not get established, this field displays 0. Also, if H.239 is not supported, this field displays 0.
incomingProtocolID	0 = Unknown, 1 = SIP, 2 = H323, 3 = CTI/JTAPI, 4 = Q931, Integer	Identifies the protocol (SIP, H.323, CTI/JTAPI, or Q.931) used between Cisco Unified CM and the upstream voice product in the call path.
incomingProtocolCallRef	Varchar(32)	Identifies the globally unique call reference identification for the protocol. The value is received from the upstream voice product. The value is alpha–numeric and truncated to 32 characters.
outgoingProtocolID	0 = Unknown, 1 = SIP, 2 = H323, 3 = CTI/JTAPI, 4 = Q931, Integer	Identifies the protocol (SIP, H.323, CTI/JTAPI, or Q.931) used between Cisco Unified CM and the downstream voice product in the call path.

Field Name	Range of Values	Description
outgoingProtocolCallRef	Varchar(32)	Identifies the globally unique call reference identification for the protocol. The value is passed to the next downstream voiced product. The value is alpha–numeric and truncated to 32 characters.
currentRoutingReason	Positive Integer	Displays the reason why the call was intercepted for the active call. This field is used with the external call control feature. See topics related to routing reason values for external call control for a list of reasons. Default value is 0.
origRoutingReason	Positive Integer	Displays the reason why the call was intercepted for the first time. This field is used with the external call control feature, See topics related to routing reason values for external call control for a list of reasons. Default value is 0.
lastRedirectingRoutingReason	Positive Integer	Displays why the call was intercepted for the last time. This field is used with the external call control feature. See topics related to routing reason values for external call control for a list of reasons. Default - Empty string.
huntPilotPartition	Text String	Indicates the partition for the hunt pilot DN. Default - Empty string.
huntPilotDN	Text String	Indicates the hunt pilot DN through which the call is routed. Default - Empty string.

Field Name	Range of Values	Description
calledPartyPatternUsage	Positive Integer	Indicates the pattern of the called party. Default value specifies 5 (PATTERN_ROUTE). • If the huntPilotDN is populated, use the huntPilotDN field value as the hunt pilot. • If the huntPilotDN is not available, check the pattern usage (7 =PATTERN_HUNT_PILOT) in the CDR table to identify the call type. If this call is a hunt list call, use the finalCalledPartyNumber as the huntPilotDN.
incomingICID	Text String	Specifies alphanumeric string up to 50 characters. This field is populated with the IMS Identifier(ICID) from the P-Charging Vector at the incoming call leg of the call. This field will be empty when the call leg has no IMS or SIP trunk with P-Charging-Vector enabled. Default = Empty String " "
incomingOrigIOI	Text String	Specifies alphanumeric string up to 50 characters. This field is populated with the originating Interoperator Identifier(IOI) from the P-Charging Vector at the incoming call leg of the call. This field will be empty when the call leg has no IMS or SIP trunk with P-Charging-Vector enabled. Default = Empty String " "

Field Name	Range of Values	Description
incomingTermIOI	Text String	Specifies alphanumeric string up to 50 characters. This field is populated with the terminating Interoperator Identifier(IOI) from the P-Charging Vector at the incoming call leg of the call. This field will be empty when the call leg has no IMS or SIP trunk with P-Charging-Vector enabled. Default = Empty String " "
outgoingICID	Text String	Specifies alphanumeric string up to 50 characters. This field is populated with the IMS Identifier(ICID) from the P-Charging Vector at the outgoing call leg of the call. This field will be empty when the call leg has no IMS or SIP trunk with P-Charging-Vector enabled. Default = Empty String " "
outgoingOrigIOI	Text String	Specifies alphanumeric string up to 50 characters. This field is populated with the originating Interoperator Identifier(IOI) from the P-Charging Vector at the outgoing call leg of the call. This field will be empty when the call leg has no IMS or SIP trunk with P-Charging-Vector enabled. Default = Empty String " "

Field Name	Range of Values	Description
outgoingTermIOI	Text String	Specifies alphanumeric string up to 50 characters. This field is populated with the terminating Interoperator Identifier(IOI) from the P-Charging Vector at the outgoing call leg of the call. This field will be empty when the call leg has no IMS or SIP trunk with P-Charging-Vector enabled. Default = Empty String " "
outpulsedOriginalCalledPartyNumber	Text String	Specifies alphanumeric string up to 50 characters. The Original called party number outpulsed from the device. Refer to section on Redirecting Number Transformation, on page 113 for details. Default = Empty String " "
outpulsedLastRedirectingNumber	Text String	Specifies alphanumeric string up to 50 characters. The Last Redirecting number outpulsed from the device. Refer to section on Redirecting Number Transformation, on page 113 for details. Default = Empty String " "
wasCallQueued	Positive Integer	Specifies whether the call has been put into a queue or not. A value of 0 means that the call is not put into any queue; 1 means the call has been put into a queue.
totalWaitTimeInQueue	Positive Integer	Specifies how long a caller has been put into a queue. The value is specified in second. The value is 0 if the call is never put into any queue.

Field Name	Range of Values	Description
callingPartyNumber_uri	Text String	Specifies an alphanumeric string of up to 254 characters that identifies the calling party if the calling party uses a directory URI for call addressing. If the calling party uses a blended address in the identity headers, this field contains the directory URI portion of the blended address. Default - Empty string " ". If the calling party does not use a directory URI, the field stays empty.
originalCalledPartyNumber_uri	Text String	Specifies a string of up to 254 alphanumeric characters that specifies the directory URI to which the original call was addressed, prior to any call forwarding, provided the call was addressed to a directory URI. If a blended address is used for the called party, this field specifies the directory URI portion of the blended address. Default - Empty string " ". If destination cannot be reached, or if the called party is a directory number, this field stays empty.

Field Name	Range of Values	Description
finalCalledPartyNumber_uri	Text String	Specifies an alphanumeric string of up to 254 characters that indicate the directory URI address to which the call finally gets presented, if the final address is a directory URI. If no forwarding occurs, this field shows the same directory URI as the originalCalledPartyNumber_uri field. If a blended address is used for the called number, this field specifies the directory URI portion of the blended address. For calls to a conference bridge, this field contains the actual identifier of the conference bridge, which is an alphanumeric string (for example, b0019901001). Default - Empty string " ". If destination cannot be reached, or if a directory number is used for called addressing, this field stays empty.

Field Name	Range of Values	Description
lastRedirectDn_uri	Text String	Specifies an alphanumeric string of up to 254 characters. For forwarded calls that use a directory URI for addressing, this field specifies the directory URI of the next to last hop before the call reaches its final destination. If only one hop occurs, this number matches the originalCalledPartyNumber_uri. If a blended address is used, this field contains only the directory URI portion of the blended address. For calls that are not forwarded, this field matches the originalCalledPartyNumber_uri and the finalCalledPartyNumber_uri. For calls to a conference bridge, this field contains the actual identifier of the conference bridge, which is an alphanumeric string (for example, b0019901001). Default - Empty string " ". If the call is never redirected, or if the address is a directory number, this field remains empty.
mobileCallingPartyNumber	Text string	Specifies the mobile cellular number if the original calling device is a mobile device. If the original calling device is not a mobile device, this field remains empty. Default - Empty string
finalMobileCalledPartyNumber	Text string	Specifies the mobile called party if the final called device is a mobile device. If the final called device is not a mobile device, this field remains empty. Default - Empty string

Field Name	Range of Values	Description
origMobileDeviceName	Text string	Specifies the device name of the calling party if the call is placed from a mobile device. If the mobile call uses a remote destination profile, the device name is the mobile number and remote destination profile name. For example, mobileNumber: RDP-name. If the mobile device uses a mobile identity, the device name is the mobile identity name. If the original device is not a mobile device, this field remains empty. Default - Empty string
destMobileDeviceName	Text string	Specifies the name of the destination mobile device. If the mobile device uses a remote destination profile the device name is the mobile number and remote destination profile name. For example, mobileNumber: RDP-name. If the mobile device uses a mobile identity, the device name is the mobile identity name. If the destination device is not a mobile device, this field remains empty. Default - Empty string
origMobileCallDuration	Positive integer	Specifies the call duration in the mobile network of the originating device if the calling party is a mobile device. If the calling party is not a mobile device, this field remains empty. Default - 0

Field Name	Range of Values	Description
destMobileCallDuration	Positive integer	Specifies the call duration in the mobile network for the destination device if the destination device is a mobile device. If the destination device is not a mobile device, this field remains empty. Default - 0
mobileCallType	Positive integer	Specifies the mobility feature that is invoked for this mobile call. Default - 0

Related Topics

Call Termination Cause Codes, on page 174
CDR Examples, on page 15
Cisco Call Management Record Field Descriptions, on page 191
Codec Types, on page 171
Convert Signed Decimal Value to IP Address, on page 12
Documentation Related to CDR, on page 4
Global Call Identifier, on page 8
Redirect Reason Codes, on page 179
Routing Reason Values for External Call Control, on page 168

Routing Reason Values for External Call Control

Unified Communications Manager supports the external call control feature, which enables an adjunct route server to make call-routing decisions for Unified Communications Manager by using the Cisco Unified Routing Rules Interface. When you configure external call control, Unified Communications Manager issues a route request that contains the calling party and called party information to the adjunct route server. The adjunct route server receives the request, applies appropriate business logic, and returns a route response that instructs Unified Communications Manager on how the call should get routed, along with any additional call treatment that should get applied.

The adjunct route server can instruct Unified Communications Manager to allow, divert, or deny the call, modify calling and called party information, play announcements to callers, reset call history so adjunct voicemail and IVR servers can properly interpret calling/called party information, and log reason codes that indicate why calls were diverted or denied.

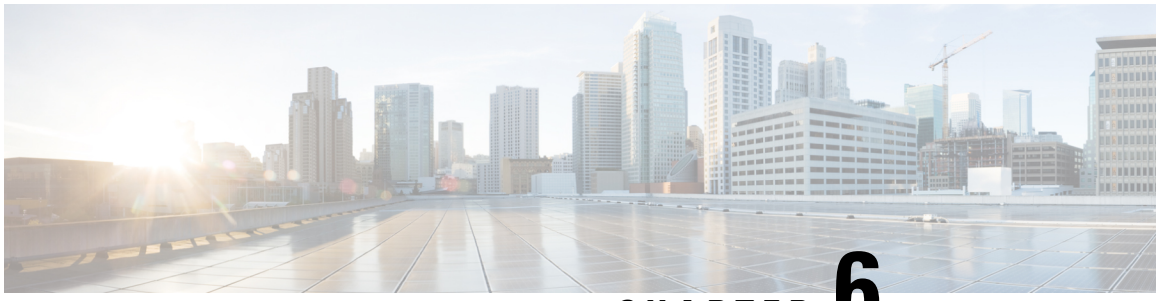
The following table includes the reasons that can display for the currentRoutingReason, origRoutingReason, or lastRedirectingRoutingReason fields.

Table 10: Routing Reason Values for External Call Control

Field Value	Reason	Description
0	PDPDecision_NONE	This value indicates that the route server did not return a routing directive to the Unified Communications Manager.
1	PDPDecision_Allow_Fulfilled	This value indicates that Unified Communications Manager allowed a call.
2	PDPDecision_Allow_Unfulfilled	This value indicates that Unified Communications Manager disallowed a call.
3	PDPDecision_Divert_Fulfilled	This value indicates that Unified Communications Manager diverted the call.
4	PDPDecision_Divert_Unfulfilled	This value indicates that Unified Communications Manager was not able to divert the call.
5	PDPDecision_Forward_Fulfilled	This value indicates that Unified Communications Manager forwarded the call.
6	PDPDecision_Forward_Unfulfilled	This value indicates that Unified Communications Manager was unable to forward the call.
7	PDPDecision_Reject_Fulfilled	This value indicates that Unified Communications Manager rejected the call.
8	PDPDecision_Reject_Unfulfilled	This value indicates that Unified Communications Manager was not able to reject the call.

Related Topics

CDR Examples, on page 15
Cisco Call Management Record Field Descriptions, on page 191
Redirect Reason Codes, on page 179



CHAPTER 6

Cisco Call Detail Records Codes

This chapter provides information about the codec types and codes that are used in the Call Detail Record fields.

Codec Types

The following table contains the compression and payload types that may appear in the codec fields.

Table 11: Codec Types

Value	Description
1	NonStandard
2	G711Alaw 64k
3	G711Alaw 56k
4	G711mu-law 64k
5	G711mu-law 56k
6	G722 64k
7	G722 56k
8	G722 48k
9	G7231
10	G728
11	G729
12	G729AnnexA

Value	Description
13	Is11172AudioCap
14	Is13818AudioCap
15	G.729AnnexB
16	G.729 Annex AwAnnexB
18	GSM Full Rate
19	GSM Half Rate
20	GSM Enhanced Full Rate
25	Wideband 256K
32	Data 64k
33	Data 56k
40	G7221 32K
41	G7221 24K
42	AAC
43	MP4ALATM_128
44	MP4ALATM_64
45	MP4ALATM_56
46	MP4ALATM_48
47	MP4ALATM_32
48	MP4ALATM_24
49	MP4ALATM_NA
80-	GSM
81	ActiveVoice
82	G726 32K
83	G726 24K
84	G726 16K
86	iLBC
89	iSAC
100	H261

Value	Description
101	H263
102	Vieo
103	H264
104	H264_SVC
105	T120
106	H224
107	T38Fax
108	TOTE
109	H265
110	H264_UC
111	XV150_MR_711U
112	NSE_VBD_711U
113	XV150_MR_729A
114	NSE_VBD_729A
115	H264_FEC
120	Clear_Chan
222	Universal_Xcoder
257	RFC2833_DynPayload
258	PassThrough
259	Dynamic_Payload_PassThru
260	DTMF_OOB
261	Inband_DTMF_RFC2833
299	NoAudio
300	v150_LC_ModemRelay
301	v150_LC_SPRT
302	v150_LC_SSE

Related Topics

CDR Examples, on page 15

Cisco Call Detail Records Field Descriptions, on page 127
Documentation Related to CDR, on page 4

Call Termination Cause Codes

The following tables contain call termination cause codes that may appear in the Cause fields in CDRs.

Note Cause Code is defined in call control as Natural number. It is a 32 bit unsigned (long) positive integer with values ranging from 0 to +4,294,967,295.

Table 12: Call Termination Cause Codes

Code	Description
0	No error
1	Unallocated (unassigned) number
2	No route to specified transit network (national use)
3	No route to destination
4	Send special information tone
5	Misdialed trunk prefix (national use)
6	Channel unacceptable
7	Call awarded and being delivered in an established channel
8	Preemption
9	Preemption—circuit reserved for reuse
16	Normal call clearing
17	User busy
18	No user responding
19	No answer from user (If "No Answer Ring duration" value is greater than the T301 Timer value and after T301 Timer expiry, Call Forwarding No Answer(CFNA) Feature would be invoked).
20	Subscriber absent
21	Call rejected
22	Number changed

Code	Description
26	Non-selected user clearing
27	Destination out of order
28	Invalid number format (address incomplete)
29	Facility rejected
30	Response to STATUS ENQUIRY
31	Normal, unspecified
34	No circuit/channel available
38	Network out of order
39	Permanent frame mode connection out of service
40	Permanent frame mode connection operational
41	Temporary failure
42	Switching equipment congestion
43	Access information discarded
44	Requested circuit/channel not available
46	Precedence call blocked
47	Resource unavailable, unspecified
49	Quality of Service not available
50	Requested facility not subscribed
53	Service operation violated
54	Incoming calls barred
55	Incoming calls barred within Closed User Group (CUG)
57	Bearer capability not authorized
58	Bearer capability not presently available
62	Inconsistency in designated outgoing access information and subscriber class
63	Service or option not available, unspecified
65	Bearer capability not implemented
66	Channel type not implemented

Code	Description
69	Requested facility not implemented
70	Only restricted digital information bearer capability is available (national use)
79	Service or option not implemented, unspecified
81	Invalid call reference value
82	Identified channel does not exist
83	A suspended call exists, but this call identity does not
84	Call identity in use
85	No call suspended
86	Call having the requested call identity has been cleared
87	User not member of CUG (Closed User Group)
88	Incompatible destination
90	Destination number missing and DC not subscribed
91	Invalid transit network selection (national use)
95	Invalid message, unspecified
96	Mandatory information element is missing
97	Message type nonexistent or not implemented
98	Message is not compatible with the call state, or the message type is nonexistent or not implemented
99	An information element or parameter does not exist or is not implemented
100	Invalid information element contents
101	The message is not compatible with the call state
102	Call terminated when timer expired; a recovery routine executed to recover from the error
103	Parameter nonexistent or not implemented - passed on (national use)
110	Message with unrecognized parameter discarded
111	Protocol error, unspecified
122	Precedence Level Exceeded

Code	Description
123	Device not Preemptable
125	Out of bandwidth (Cisco specific)
126	Call split (Cisco specific)
127	Interworking, unspecified
129	Precedence out of bandwidth
131	Call Control Discovery PSTN Failover (Cisco specific)
132	IME QOS Fallback (Cisco specific)
133	PSTN Fallback locate Call Error (Cisco specific)
134	PSTN Fallback wait for DTMF Timeout (Cisco specific)
135	IME Failed Connection Timed out (Cisco specific)
136	IME Failed not enrolled (Cisco specific)
137	IME Failed socket error (Cisco specific)
138	IME Failed domain blacklisted (Cisco specific)
139	IME Failed prefix blacklisted (Cisco specific)
140	IME Failed expired ticket (Cisco specific)
141	IME Failed remote no matching route (Cisco specific)
142	IME Failed remote unregistered (Cisco specific)
143	IME Failed remote IME disabled (Cisco specific)
144	IME Failed remote invalid IME trunk URI (Cisco specific)
145	IME Failed remote URI not E164 (Cisco specific)
146	IME Failed remote called number not available (Cisco specific)
147	IME Failed Invalid Ticket (Cisco specific)
148	IME Failed unknown (Cisco specific)

Table 13: Cisco-Specific Call Termination Cause Codes

Decimal Value Code	Hex Value Code	Description
262144	0x40000	Conference Full (was 124)
393216	0x60000	Call split (was 126)This code applies when a call terminates during a transfer operation because it was split off and terminated (was not part of the final transferred call). This code can help you to determine which calls terminated as part of a feature operation.
458752	0x70000	Conference drop any party/Conference drop last party (was 128)
16777257	0x1000029	CCM_SIP_400_BAD_REQUEST
33554453	0x2000015	CCM_SIP_401_UNAUTHORIZED
50331669	0x3000015	CCM_SIP_402_PAYMENT_REQUIRED
67108885	0x4000015	CCM_SIP_403_FORBIDDEN
83886081	0x5000001	CCM_SIP_404_NOT_FOUND
100663359	0x600003F	CCM_SIP_405_METHOD_NOT_ALLOWED
117440591	0x700004F	CCM_SIP_406_NOT_ACCEPTABLE
134217749	0x8000015	CCM_SIP_407_PROXY_AUTHENTICATION_REQUIRED
150995046	0x9000066	CCM_SIP_408_REQUEST_TIMEOUT
184549398	0xB000016	CCM_SIP__410_GONE
201326719	0xC00007F	CCM_SIP_411_LENGTH_REQUIRED
234881151	0xE00007F	CCM_SIP_413_REQUEST_ENTITY_TOO_LONG
251658367	0xF00007F	CCM_SIP_414_REQUEST_URI_TOO_LONG
268435535	0x1000004F	CCM_SIP_415_UNSUPPORTED_MEDIA_TYPE
285212799	0x1100007F	CCM_SIP_416_UNSUPPORTED_URI_SCHEME
83886207	0x1500007F	CCM_SIP_420_BAD_EXTENSION
369098879	0x1600007F	CCM_SIP_421_EXTENSION_REQUIRED
402653311	0x1800007F	CCM_SIP_423_INTERVAL_TOO_BRIEF
419430421	0x19000015	CCM_SIP_424_BAD_LOCATION_INFO

Decimal Value Code	Hex Value Code	Description
503316501	0x1E000015	CCM_SIP_429_PROVIDE_REFER_IDENTITY
1073741842	0x40000012	CCM_SIP_480_TEMPORARILY_UNAVAILABLE
1090519081	0x41000029	CCM_SIP_481_CALL_LEG_DOES_NOT_EXIST
1107296281	0x42000019	CCM_SIP_482_LOOP_DETECTED = 0x42000000 + EXCHANGE_ROUTING_ERROR
1124073497	0x43000019	CCM_SIP_483_TOO_MANY_HOOPS
1140850716	0x4400001C	CCM_SIP_484_ADDRESS_INCOMPLETE
1157627905	0x45000001	CCM_SIP_485_AMBIGUOUS
1174405137	0x46000011	CCM_SIP_486_BUSY_HERE
1191182367	0x4700001F	CCM_SIP_487_REQUEST_TERMINATED
1207959583	0x4800001F	CCM_SIP_488_NOT_ACCEPTABLE_HERE
1258291217	0x4B000011	CCM_SIP_491_REQUEST_PENDING
1291845649	0x4D000011	CCM_SIP_493_UNDECIPHERABLE
1409286185	0x54000029	CCM_SIP_500_SERVER_INTERNAL_ERROR
1442840614	0x56000026	CCM_SIP_502_BAD_GATEWAY
1459617833	0x57000029	CCM_SIP_503_SERVICE_UNAVAILABLE
2801795135	0xA700003F	CCM_SIP_503_SERVICE_UNAVAILABLE_SR_OPTION_NOAV
1476395110	0x58000066	CCM_SIP__504_SERVER_TIME_OUT
1493172351	0x5900007F	CCM_SIP_505_SIP_VERSION_NOT_SUPORTED
1509949567	0x5A00007F	CCM_SIP_513_MESSAGE_TOO_LARGE
2701131793	0xA1000011	CCM_SIP_600_BUSY_EVERYWHERE
2717909013	0xA2000015	CCM_SIP_603_DECLINE
2734686209	0xA3000001	CCM_SIP_604_DOES_NOT_EXIST_ANYWHERE
2751463455	0xA400001F	CCM_SIP_606_NOT_ACCEPTABLE

Redirect Reason Codes

The following table contains the available Redirect Reason Codes that may appear in a record.

Q.931 Standard Redirect Reason Codes	
Value	**Description**
0	Unknown
1	Call Forward Busy
2	Call Forward No Answer
4	Call Transfer
5	Call Pickup
7	Call Park
8	Call Park Pickup
9	CPE Out of Order
10	Call Forward
11	Call Park Reversion
15	Call Forward all
Nonstandard Redirect Reason Codes	
18	Call Deflection
34	Blind Transfer
50	Call Immediate Divert
66	Call Forward Alternate Party
82	Call Forward On Failure
98	Conference
114	Barge
129	Aar
130	Refer
146	Replaces
162	Redirection (3xx)
177	SIP-forward busy greeting
178	Call Forward Unregistered
207	Follow Me (SIP-forward all greeting)
209	Out of Service (SIP-forward busy greeting)

Q.931 Standard Redirect Reason Codes	
Value	**Description**
239	Time of Day (SIP-forward all greeting)
242	Do Not Disturb (SIP-forward no answer greeting)
257	Unavailable (SIP-forward busy greeting)
274	Away (SIP-forward no answer greeting)
303	Mobility HandIn
319	Mobility HandOut
335	Mobility Follow Me
351	Mobility Redial
354	Recording
370	Monitoring
399	Mobility IVR
401	Mobility DVOR
402	Mobility EFA
403	Mobility Session Handoff
415	Mobility Cell Pickup
418	Click to Conference
434	Forward No Retrieve
450	Forward No Retrieve Send Back to Parker
464	Call Control Discovery (indicates that the call is redirected to a PSTN failover number)
480	Intercompany Media Engine (IME)
496	IME Connection Timed Out
512	IME Not Enrolled
528	IME Socket Error
544	IME Domain Blacklisted
560	IME Prefix Blacklisted
576	IME Expired Ticket

Q.931 Standard Redirect Reason Codes	
Value	**Description**
592	IME Remote No Matching Route
608	IME Remote Unregistered
624	IME Remote IME Disabled
640	IME Remote Invalid IME Trunk URI
656	IME Remote URI not E164
672	IME Remote Called Number Not Available
688	IME Invalid Ticket
704	IME Unknown
720	IME PSTN Fallback
738	Presence Enabled Routing
752	Agent Greeting
783	NuRD
786	Native Call Queuing, queue a call
802	Native Call Queuing, de-queue a call
818	Native Call Queuing, redirect to the second destination when no agent is logged in
834	Native Call Queuing, redirect to the second destination when the queue is full
850	Native Call Queuing, redirect to the second destination when the maximum wait time in queue is reached

OnBehalfof Codes

The following table contains the available OnBehalfof Codes that may appear in a CDR record.

Table 14: OnBehalfof Codes

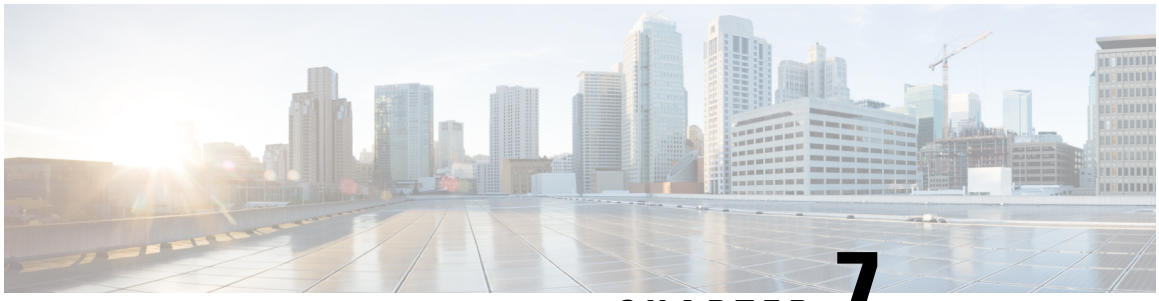
Value	Description
0	Unknown
1	CctiLine
2	Unicast Shared Resource Provider

Value	Description
3	Call Park
4	Conference
5	Call Forward
6	Meet-Me Conference
7	Meet-Me Conference Intercepts
8	Message Waiting
9	Multicast Shared Resource Provider
10	Transfer
11	SSAPI Manager
12	Device
13	Call Control
14	Immediate Divert
15	Barge
16	Pickup
17	Refer
18	Replaces
19	Redirection
20	Callback
21	Path Replacement
22	FacCmc Manager
23	Malicious Call
24	Mobility
25	Aar
26	Directed Call Park
27	Recording
28	Monitoring
29	CCDRequestingService
30	Intercompany Media Engine

Value	Description
31	FallBack Manager
32	Presence Enabled Routing
33	AgentGreeting
34	NativeCallQueuing
35	MobileCallType

PART III

Call Management Records



CHAPTER 7

Call Management Records

This chapter describes the format and logic of the call management records (CMRs) that the Unified Communications Manager system generates, and how to access the CMR files.

Call Management Records

The Unified Communications Manager system generates call management records (CMRs) . You can use this information for post-processing activities such as generating billing records and network analysis.

When you install your system, CMRs remain disabled by default. You can enable or disable CMRs at any time that the system is in operation. You do not need to restart Unified Communications Manager for the change to take effect. The system responds to all changes within a few seconds. The system enables CMR or diagnostic data separately from CDR data.

Related Topics

Cisco Call Management Record Field Descriptions, on page 191
Cisco Call Management Records K-Factor Data, on page 205
Documentation Related to CDR, on page 4
Example Cisco Call Management Records, on page 209

CMR Processing

The CMR records store information about the quality of the streamed audio and video of the call.

When Unified Communications Manager places or receives a call, the system generates a CDR record when the call terminates. The system writes the CDR to a flat file (text file). Inside the Unified Communications Manager, the call control process generates CDR records. The system writes records when significant changes occur to a given call, such as ending the call, transferring the call, redirecting the call, splitting the call, joining a call, and so forth.

When CMR records are enabled, the number of records that are written varies by type of call and the call scenario. When Diagnostics are enabled, the device generates CMR records for each call. The system writes

one CMR record for each IP phone that is involved in the call or for each Media Gateway Control Protocol (MGCP) gateway. The system sends these records to EnvProcessCdr where they get written to flat files.

Note For Hold and Resume calls on SCCP devices, multiple CMR Records are generated based on the number of times the call is put on hold.

The Unified Communications Manager generates CMR records but does not perform any post processing on the records. The system writes the records to comma-delimited flat files and periodically passes them to the CDR Repository. The CMR files represent a specific filename format within the flat file.

Filename Format

The following example shows the full format of the filename:
tag_clusterId_nodeId_datetime_seqNumber

- tag—Identifies the type of file, either CDR or CMR.
- clusterId—Identifies the cluster or server where the Unified Communications Manager database exists.
- nodeId—Identifies the node.
- datetime—Specifies UTC time in yyyymmddhhmm format.
- seqnumber—Specifies sequence number.

An example of the filename follows:

- `cmr_Cluster1_02_200404061011_6125`

Flat File Format

The CMR flat files have the following format:

- Line 1—List of field names in comma separated format.
- Line 2—List of field types in comma separated format.
- Line 3—Data in comma separated format.
- Line 4—Data in comma separated format.

The following example shows a flat file:

```
Line1-"cmrRecordType","globalCallID_callManagerId","globalCallID_callId","origLegCallIdentifier",...
Line2-INTEGER,INTEGER,INTEGER,INTEGER,...
Line3-1,1,388289,17586046,...
Line4-1,1,388293,17586054,...
```

Set Up CMRs

You can configure CMRs on the Service Parameters Configuration window in Cisco Unified CM Administration. To access the **Service Parameters Configuration** window, open Cisco Unified CM Administration and choose **System** > **Service Parameters**. Choose the **Advanced** button to display the complete list of Service Parameters. Select the **Call Diagnostics Enabled** parameter.

This parameter determines whether the system generates CMRs, also called call diagnostic records. Valid values specify Disabled (do not generate CMRs), Enabled Only When CDR Enabled Flag is True (generate CMRs only when the CDR Enabled Flag service parameter is set to True), or Enabled Regardless of CDR Enabled Flag (generates CMRs without regard to the setting in the CDR Enabled Flag service parameter). This represents a required field. The default value specifies Disabled.

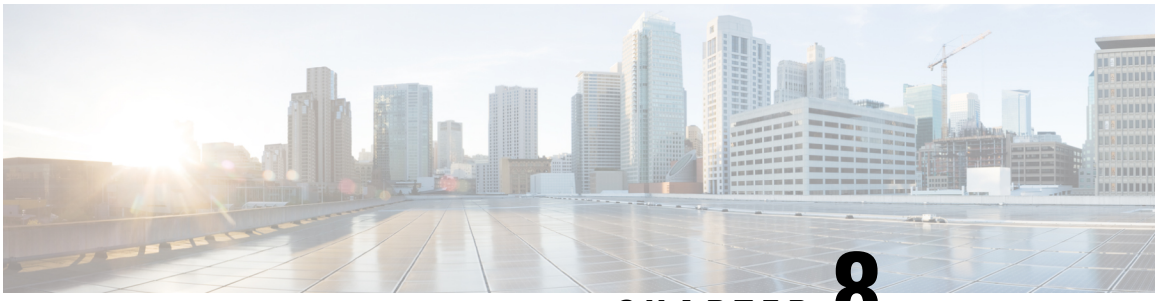
CPU Utilization

Cisco has performed basic testing to measure CPU utilization when CDRs and/or CMRs are enabled. The CPU utilization testing was measured on subscribers and was not measured on the publishers. Your actual results can vary because of the CDR Loader settings and the CDR Management settings for external billing servers. The following table displays the results of these tests.

Note Be aware that these tests were performed with Unified Communications Manager Release 8.0(1).

Table 15: CDR and CMR CPU Utilization

CDRs and CMRs Enabled/Disabled	Average % Increase in Cisco Unified CM CPU Utilization	Average % Increase in Total CPU Utilization	% Increase in Cisco Unified CM CPU	% Increase in Total CPU
CDRs disabled, CMRs disabled	6.17	11.15	-	-
CDRs enabled, CMRs disabled	6.99	12.10	13.18	8.57
CDRs disabled, CMRs enabled	6.38	11.24	3.43	0.86
CDRs enabled, CMRs enabled	7.71	13.04	24.92	17.02



CHAPTER 8

Cisco Call Management Record Field Descriptions

This chapter describes the field descriptions of the Call Management Records (CMRs).

• CMR Field Descriptions, on page 191

CMR Field Descriptions

The following table contains the fields, range of values, and field descriptions of the CMRs in the order in which they appear in the CMR.

Table 16: CMR Field Descriptions

Field Name	Range of Values	Description
cdrRecordType	0, 1, or 2	Specifies the type of this specific record. The following valid values apply: • 0—Start call detail record (not used) • 1—End call detail record • 2—CMR record Default - For CMRs, this field always specifies 2.

Field Name	Range of Values	Description
globalCallID_callManagerId	Positive Integer	Specifies a unique Unified Communications Manager identity. This field makes up half of the Global Call ID. The Global Call ID comprises the following fields: • globalCallId_callId • globalCallID_callManagerID All records that are associated with a standard call have the same Global Call ID in them. Default - Ensure that this field is populated.
globalCallId_callId	Positive Integer	Specifies a unique call identity value that gets assigned to each call. The system allocates this identifier independently on each call server. Values get chosen sequentially when a call begins. Each call, successful or unsuccessful, receives value assignment. This field makes up half the Global Call ID. The Global Call ID comprises the following two fields: • globalCallId_callId • globalCallID_callManagerID All records that are associated with a standard call have the same Global Call ID in them. Default - Ensure that this field is populated.
nodeId	Positive Integer	Specifies the server or node within the Unified Communications Manager cluster, where this record gets generated. Default - Ensure that this field is populated.

Field Name	Range of Values	Description
directoryNumber	Integer	Specifies the directory number of the device from which these diagnostics are collected. Default - Ensure that this field is populated.
callIdentifier	Positive Integer	Identifies the call leg to which this record pertains. Default - Ensure that this field is populated.
dateTimeStamp	Integer	Represents the approximate time that the device goes on the hook. Unified Communications Manager records the time when the phone responds to a request for diagnostic information. Default - Ensure that this field is populated.
numberPacketsSent	Integer	Designates the total number of Routing Table Protocol (RTP) data packets that the device transmits since starting transmission on this connection. The value remains zero if the connection is set to “receive only” mode. Default - 0
numberOctetsSent	Integer	Specifies the total number of payload octets (that is, not including header or padding) that the device transmits in RTP data packets since starting transmission on this connection. The value remains zero if the connection is set to “receive only” mode. Default - 0

Field Name	Range of Values	Description
numberPacketsReceived	Integer	Specifies the total number of RTP data packets that the device has received since starting reception on this connection. The count includes packets that are received from different sources if this is a multicast call. The value remains zero if the connection is set in "send only" mode. Default - 0
numberOctetsReceived	Integer	Specifies the total number of payload octets (that is, not including header or padding) that the device has received in RTP data packets since starting reception on this connection. The count includes packets that are received from different sources if this is a multicast call. The value remains zero if the connection is set in "send only" mode. Default - 0
numberPacketsLost	Integer	Designates the total number of RTP data packets that have been lost since the beginning of reception. This number designates the number of packets that were expected, less the number of packets that were received, where the number of packets that were received includes any that are late or duplicates. Thus, packets that arrive late do not get counted as lost, and the loss may be negative if duplicate packets exist. The number of packets that are expected designates the extended last sequence number that was received, as defined next, less the initial sequence number that was received. The value remains zero if the connection was set in "send only" mode. For detailed information, see RFC 1889. Default - 0

Field Name	Range of Values	Description
jitter	Integer	Provides an estimate of the statistical variance of the RTP data packet interarrival time that is measured in milliseconds and expressed as an unsigned integer. The interarrival jitter J specifies the mean deviation (smoothed absolute value) of the difference D in the packet spacing at the receiver, compared to the sender for a pair of packets. RFC 1889 contains detailed computation algorithms. The value remains zero if the connection was set in “send only” mode. Default - 0
latency	Integer	Designates value that is an estimate of the network latency, expressed in milliseconds. This value represents the average value of the difference between the NTP timestamp that the RTP Control Protocol (RTCP) messages indicate and the NTP timestamp of the receivers, measured when these messages are received. Unified Communications Manager obtains the average by summing all estimates then dividing by the number of RTCP messages that have been received. For detailed information, see RFC 1889. Default - 0 **Note** CMR records will not show latency for all phone loads. For example, for SIP 9.2.1 and 9.2.2, the latency will not show, as it has not been implemented in these loads.

Field Name	Range of Values	Description
pkid	Text String	Identifies a text string that the database uses internally to uniquely identify each row. This text string provides no meaning to the call itself. Default - The system always populates this field with a unique ID.
directoryNumberPartition	Text String	Identifies the partition of the directory number. Default - Empty string “”. This field may remain empty if no partition exists.
globalCallId_ClusterId	Text String	Designates a unique ID that identifies a single Unified Communications Manager, or a cluster of Unified Communications Managers. The system generates this field during installation, but Unified Communications Manager does not use it: globalCallId_ClusterId + globalCallId_callManagerId + globalCallId_callId. Default - Ensure that this field is populated.
deviceName	Text String	Identifies the name of the device. Default - Empty string “”. This field may remain empty if no device name exists.

Field Name	Range of Values	Description
varVQMetrics	Text String	Contains a variable number of voice quality metrics. This field comprises a string of voice quality metrics that are separated by a semicolon. The format of the string follows: fieldName=value;fieldName=value.precision This example shows voice quality data, but the names may differ. "MLQK=4.5000;MLQKav=4.5000;MLQKmn=4.5000;MLQKmx=4.5000;MLQKvr=0.0;CCR=0.0000;ICR=0.0000;ICRmx=0.0000;CS=0;SCS=0" **Note** See topics that are related to K-factor data stored in Unified Communications Manager CMRs for a complete list of K-Factor data.
duration	Integer	Specifies the duration value of audio session that is expressed in seconds. This is reported only for SIP phones.
videoContentType	Text String	Identifies the type of video stream. It can be “main”, “speaker” or “slides”. For an audio-only call, no video metrics are populated.
videoDuration	Integer	Specifies the duration value of the first video session, expressed in seconds.
numberVideoPacketsSent	Integer	Specifies the total number of RTP data packets that are transmitted by the device since starting transmission on this connection.
numberVideoOctetsSent	Integer	Specifies the total number of payload octets (that is, not including header or padding) transmitted in RTP data packets by the device since starting transmission on this connection.

Field Name	Range of Values	Description
numberVideoPacketsReceived	Integer	Specifies the total number of RTP data packets received by the device since starting reception on this connection.
numberVideoOctetsReceived	Integer	Specifies the total number of payload octets (that is, not including header or padding) received in RTP data packets by the device since starting reception on this connection.
numberVideoPacketsLost	Integer	Specifies the total number of RTP data packets that have been lost since the beginning of reception on this connection.
videoAverageJitter	Integer	Provides an estimate of the statistical variance of the RTP data packet interarrival time for this connection that is measured in milliseconds and expressed as an unsigned integer. For detailed information, see RFC 3550 for details.
videoRoundTripTime	Integer	This is a measure of average round trip time between the two endpoints for this connection. It is expressed in milliseconds. For detailed information, see RFC 3550 and RFC 3611 for detail
videoOneWayDelay	Text String	This is a measure of the average one-way delay (OWD) between the endpoints for this connection. This is only available if endpoints are time synchronized (same NTP source) and is measured in milli-seconds; otherwise “NA”

Field Name	Range of Values	Description
videoTransmissionMetrics	Text String	Contains a variable number of Cisco defined metrics that are related to RTP transmission on this connection. These metrics are delimited by a semicolon. The format of this string is: CiscoTxVM="TxCodec=xxx; TxBw=xxx;TxBwMax=xxx; TxReso=xxx;TxFrameRate=xxx" TxCodec identifies the type of video codec used for the transmitted video stream. TxBw identifies the actual bandwidth that is used for the transmitted video stream. TxBwMax identifies the maximum negotiated bandwidth for the transmitted video stream. TxReso identifies the resolution of the transmitted video stream (for example, 640x480). TxFrameRate identifies the average frame rate that is measured in frames per second for the transmitted video stream.

Field Name	Range of Values	Description
videoReceptionMetrics	TextString	Contains a variable number of Cisco defined metrics that are related to RTP reception on this connection. These metrics are delimited by a semicolon. The format of this string is: CiscoRxVM="CS=xxx; SCS=xxx;DSCP=xxx; DSCPunad=xxx; RxFramesLost=xxx;RxCodec=xxx; RxBw=xxx;RxBwMax=xxx; RxReso=xxx;RxFrameRate=xxx" CS identifies the concealed seconds metrics for the received video stream. SCS identifies the severely concealed seconds for the received video stream. DSCP is useful in verifying the DSCP value of the received video stream marked by endpoint. DSCPunad is useful in verifying the DSCP value of the received video stream marked by endpoint. RxCodec identifies the type of video codec used for received video stream. RxBw identifies the actual bandwidth used for the received video stream. RxBwMax identifies the maximum negotiated bandwidth for the received video stream. RxReso identifies the resolution of the received video stream (for example, 640x480). RxFrameRate identifies the average frame rate measured in frames per second for the received video stream.

Field Name	Range of Values	Description
videoContentType_channel2	Text String	Identifies the type of second video stream if it exists. If it does not exist, the remaining metrics for the second video stream will not be populated.
videoDuration_channel2	Integer	Specifies the duration of the second video session, expressed in seconds.
numberVideoPacketsSent_channel2	Integer	Specifies the total number of RTP data packets that are transmitted by the device since starting transmission on this connection.
numberVideoOctetsSent_channel2	Integer	Specifies the total number of payload octets (that is, not including header or padding) transmitted in RTP data packets by the device since starting transmission on this connection.
numberVideoPacketsReceived_channel2	Integer	Specifies the total number of RTP data packets received by the device since starting reception on this connection.
numberVideoOctetsReceived_channel2	Integer	Specifies the total number of payload octets (that is, not including header or padding) received in RTP data packets by the device since starting reception on this connection.
numberVideoPacketsLost_channel2	Integer	The total number of RTP data packets that have been lost since the beginning of reception on this connection.
videoAverageJitter_channel2	Integer	Provides an estimate of the statistical variance of the RTP data packet interarrival time for this connection that is measured in milliseconds and expressed as an unsigned integer. For more information, see RFC 3550.

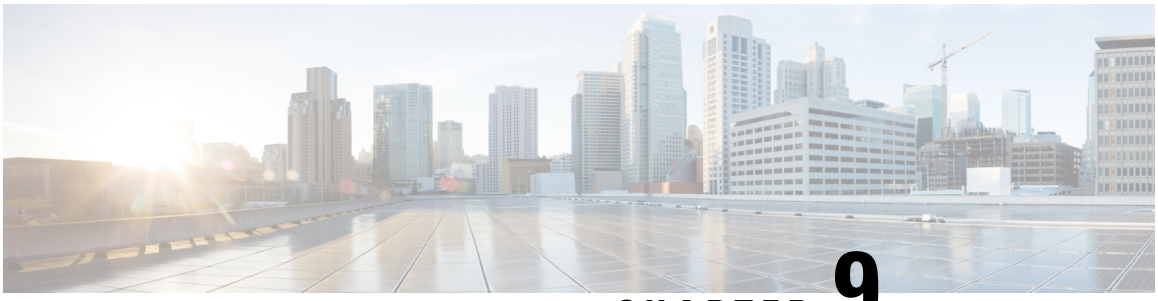
Field Name	Range of Values	Description
videoRoundTripTime_channel2	Integer	This is a measure of average round trip time between the two endpoints for this connection. It is expressed in milliseconds. For more information, see RFC 3550 and RFC 3611 for details.
videoOneWayDelay_channel2	Integer	This is a measure of the average one-way delay (OWD) between the endpoints for this connection. This is only available if endpoints are time synchronized (same NTP source) and is measured in milli-seconds; otherwise “NA”.
videoReceptionMetrics_channel2	Text String	Contains a variable number of Cisco defined metrics that are related to RTP transmission on this connection. These metrics are delimited by a semicolon. The format of this string is: CiscoTxVM="TxCodec=xxx; TxBw=xxx ;TxBwMax=xxx; TxReso=xxx;TxFrameRate=xxx" TxCodec identifies the type of video codec used for the transmitted video stream. TxBw identifies the actual bandwidth used for the transmitted video stream. TxBwMax identifies the maximum negotiated bandwidth for the transmitted video stream. TxReso identifies the resolution of the transmitted video stream (for example, 640x480). TxFrameRate identifies the average frame rate measured in frames per second for the transmitted video stream.

Field Name	Range of Values	Description
videoTransmissionMetrics_channel2	Text String	Contains a variable number of Cisco defined metrics that are related to RTP reception on this connection. These metrics are delimited by a semicolon. The format of this string is: CiscoRxVM="CS=xxx; SCS=xxx;DSCP=xxx; DSCPunad=xxx; RxFramesLost=xxx;RxCodec=xxx; RxBw=xxx;RxBwMax=xxx; RxReso=xxx;RxFrameRate=xxx" CS identifies the concealed seconds metrics for the received video stream. SCS identifies the severely concealed seconds for the received video stream. DSCP is useful in verifying the DSCP value of the received video stream that is marked by endpoint. DSCPunad is useful in verifying the DSCP value of the received video stream marked by endpoint. RxCodec identifies the type of video codec used for the received video stream. RxBw identifies the actual bandwidth that is used for the received video stream. RxBwMax identifies the maximum negotiated bandwidth for the received video stream. RxReso identifies the resolution of the received video stream (for example, 640x480). RxFrameRate identifies the average frame rate that is measured in frames per second for the received video stream.

Related Topics

Call Management Records, on page 187
Cisco Call Detail Records Field Descriptions, on page 127

Cisco Call Management Records K-Factor Data, on page 205
Documentation Related to CDR, on page 4
Example Cisco Call Management Records, on page 209



CHAPTER 9

Cisco Call Management Records K-Factor Data

This chapter provides information about the K-factor data that is present in the Cisco call management records (CMRs).

K-Factor Data

K-factor represents an endpoint mean opinion score (MOS) estimation algorithm that is defined in ITU standard P.VTQ. It represents a general estimator that is used to estimate the mean value of a perceptual evaluation of speech quality (PESQ) population for a specific impairment pattern.

MOS relates to the output of a well designed listening experiment. All MOS experiments use a five-point PESQ scale as defined in ITU standard P.862.1, which describes the PESQ as an objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs.

The MOS estimate provides a number that is inversely proportional to frame loss density. Clarity decreases as more frames are lost or discarded at the receiving end. Consider the loss or discarding of these frames as concealment. Concealment statistics measure packet (frame) loss and its effect on voice quality in an impaired network.

K-factor represents a weighted estimate of average user annoyance due to distortions that are caused by effective packet loss such as dropouts and warbles. It does not estimate the impact of delay-related impairments such as echo. It provides an estimate of listening quality (MOS-LQO) rather than conversational quality (MOS-CQO), and measurements of average user annoyance range from 1 (poor voice quality) to 5 (very good voice quality).

K-factor gets trained or conditioned by speech samples from numerous speech databases, where each training sentence or network condition that is associated with a P.862.1 value has a duration of 8 seconds. For more accurate scores, the system generates k-factor estimates for every 8 seconds of active speech.

Consider K-factor and other MOS estimators to be secondary or derived statistics because they warn a network operator of frame loss only after the problem becomes significant. Packet counts, concealment ratios, and concealment second counters represent primary statistics because they alert the network operator before network impairment has an audible impact or is visible through MOS.

The following table displays the K-factor date that is stored in the Unified Communications Manager CMRs.

Table 17: K-Factor Data Stored in Unified Communications Manager CMRs

Field Name	Phone Display Name	D&I User Interface Text and Description
CCR	Cum Conceal Ratio	Cumulative Conceal Ratio represents the cumulative ratio of concealment time over speech time that is observed after starting a call.
ICR	Interval Conceal Ratio	Interval Conceal Ratio represents an interval-based average concealment rate that is the ratio of concealment time over speech time for the last 3 seconds of active speech.
ICRmx	Max Conceal Ratio	Interval Conceal Ratio Max represents the maximum concealment ratio that is observed during the call.
CS	Conceal Secs	Conceal Secs represents the time during which some concealment is observed during a call.
SCS	Severely Conceal Secs	Severely Conceal Secs represents the time during which a significant amount of concealment is observed. If the concealment that is observed is usually greater than 50 milliseconds or approximately 5 percent, the speech probably does not seem very audible.
MLQK	MOS LQK	MOS Listening Quality K-factor provides an estimate of the MOS score of the last 8 seconds of speech on the reception signal path.
MLQKmn	Min MOS LQK	MOS Listening Quality K-factor Min represents the minimum score that is observed since the beginning of a call and represents the worst sounding 8-second interval.
MLQKmx	Max MOS LQK	MOS Listening Quality K-factor Max represents the maximum score that is observed since the beginning of a call and represents the best sounding 8-second interval.

Field Name	Phone Display Name	D&I User Interface Text and Description
MLQKav	Avg MOS LQK	MOS Listening Quality K-factor Avg8 represents the running average of scores that are observed since the beginning of a call.

The following table displays the devices that support K-factor (varQMetrics) in the CMR.

The K-factor support legend follows:

- X—Supported by phones that are running both SCCP and SIP
- S—SCCP feature only
- SI—SIP feature only
- G—Available on Cisco 5510 DSPs only

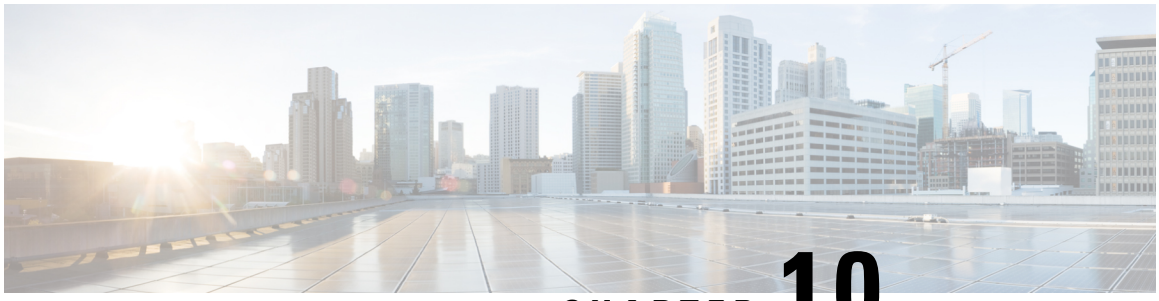
Table 18: Devices That Support K-factor (varVQMetrics) in CMRs

Device	K-factor (varVQMetrics) Support in CMR
Cisco Unified IP Phone 7906	X
Cisco Unified IP Phone 7911	X
Cisco Unified IP Phone 7921	X
Cisco Unified IP Phone7931	X
Cisco Unified IP Phone 7940	S
Cisco Unified IP Phone 7941	X
Cisco Unified IP Phone 7942-G	X
Cisco Unified IP Phone 7942-G/GE	X
Cisco Unified IP Phone 7945	X
Cisco Unified IP Phone7960	S
Cisco Unified IP Phone 7961	X
Cisco Unified IP Phone 7962-G	X
Cisco Unified IP Phone7962-G/GE	X
Cisco Unified IP Phone 7965	X
Cisco Unified IP Phone 7970	X
Cisco Unified IP Phone7971	X
Cisco Unified IP Phone7972-G/GE	X
Cisco Unified IP Phone7975	X

Device	K-factor (varVQMetrics) Support in CMR
3x MGCP Gateways	G
5x MGCP Gateways	G

Related Topics

Call Management Records, on page 187
Cisco Call Management Record Field Descriptions, on page 191
Documentation Related to CDR, on page 4
Example Cisco Call Management Records, on page 209



CHAPTER 10

Example Cisco Call Management Records

This chapter provides examples of call management records (CMRs).

CMR Examples

The following examples of CMRs get generated during a normal call (IP phone to IP phone). Normal calls log three records per call: one CDR and two CMRs (one for each endpoint).

These examples represent a call between directory number 1010 and 1014. See related topics for a sample of the CDR that gets generated during a normal call.

Example 1: SCCP to SCCP Phone

A successful call between two Cisco IP Phones generates 2 CMRs at the end of the call, one for each endpoint. This example has both endpoints as SCCP phones that do not support the new video metrics. They are left at default.

Note "Duration" field in CMR is filled only for SIP phones.

CMR 1

Field Names	Values
cdrRecordType	2
globalCallID_callManagerId	1
globalCallID_callId	96004
nodeId	1
directoryNum	1010
callIdentifier	28141535

Field Names	Values
dateTimeStamp	1202412060
numberPacketsSent	358
numberOctetsSent	61576
numberPacketsReceived	351
numberOctetsReceived	60372
numberPacketsLost	1
jitter	0
latency	0
pkid	e95df5b1-2914-4a03-befb-0f58bf16392d
directoryNumPartition	
globalCallIdClusterID	StandAloneCluster
deviceName	SEP003094C39BE7
varVQMetrics	MLQK=0.0000;MLQKav=0.0000; MLQKmn=0.0000;MLQKmx=0.0000;MLQKvr=0.95; CCR=0.0000;ICR=0.0000;ICRmx=0.0000;CS=0; SCS=0
duration	
videoContentType	
videoDuration	
numberVideoPacketsSent	
numberVideoOctetsSent	
numberVideoPacketsReceived	
numberVideoOctetsReceived	
numberVideoPacketsLost	
videoAverageJitter	
videoRoundTripTime	
videoOneWayDelay	
videoReceptionMetrics	
videoTransmissionMetrics	

Field Names	Values
videoContentType_channel2	
videoDuration_channel2	
numberVideoPacketsSent_channel2	
numberVideoOctetsSent_channel2	
numberVideoPacketsReceived_channel2	
numberVideoOctetsReceived_channel2	
numberVideoPacketsLost_channel2	
videoAverageJitter_channel2	
videoRoundTripTime_channel2	
videoOneWayDelay_channel2	
videoReceptionMetrics_channel2	
videoTransmissionMetrics_channel2	

CMR 2

Field Name	Values
cdrRecordType	2
globalCallID_callManagerId	1
globalCallID_callId	96004
nodeId	1
directoryNum	1004
callIdentifier	28141536
dateTimeStamp	1202412060
numberPacketsSent	352
numberOctetsSent	60544
numberPacketsReceived	356
numberOctetsReceived	61232
numberPacketsLost	1
jitter	0

Field Name	Values
latency	0
pkid	545ff25a-5475-4882-af09-c7b714802703
directoryNumPartition	
globalCallIdClusterID	StandAloneCluster
deviceName	SEP0007EBBA6376
varVQMetrics	MLQK=0.0000;MLQKav=0.0000; MLQKmn=0.0000; MLQKmx=0.0000;MLQKvr=0.95; CCR=0.0000; ICR=0.0000;ICRmx=0.0000;CS=0; SCS=0
duration	
videoContentType	
videoDuration	
numberVideoPacketsSent	
numberVideoOctetsSent	
numberVideoPacketsReceived	
numberVideoOctetsReceived	
numberVideoPacketsLost	
videoAverageJitter	
videoRoundTripTime	
videoOneWayDelay	
videoReceptionMetrics	
videoTransmissionMetrics	
videoContentType_channel2	
videoDuration_channel2	
numberVideoPacketsSent_channel2	
numberVideoOctetsSent_channel2	
numberVideoPacketsReceived_channel2	
numberVideoOctetsReceived_channel2	
numberVideoPacketsLost_channel2	

Field Name	Values
videoAverageJitter_channel2	
videoRoundTripTime_channel2	
videoOneWayDelay_channel2	
videoReceptionMetrics_channel2	
videoTransmissionMetrics_channel2	

Example 2: SIP to SIP Phone That Supports Main Video Metrics

The following CMR flat file is an example of SIP to SIP phone that supports video metrics.

Field Name	Values
cdrRecordType	2
globalCallID_callManagerId	1
globalCallID_callId	17001
nodeId	1
directoryNum	139098
callIdentifier	32216238
dateTimeStamp	1379591701
numberPacketsSent	170
numberOctetsSent	10370
numberPacketsReceived	169
numberOctetsReceived	12337
numberPacketsLost	0
jitter	2
latency	0
pkid	ea0cddd0-7ddd-4a4e-a697-ca405e39292c
directoryNumPartition	
globalCallId_ClusterID	StandAloneCluster
deviceName	SEPD0C7891411C3

Field Name	Values
varVQMetrics	MLQK=0.0000;MLQKav=0.0000;MLQKmn=0.0000;MLQKmx=0.0000;MLQKvr=;CCR=0.0000;ICR=0.0000;ICRmx=0.0000;CS=0;SCS=0
duration	3
videoContentType	main
videoDuration	3
numberVideoPacketsSent	140
numberVideoOctetsSent	126355
numberVideoPacketsReceived	141
numberVideoOctetsReceived	128214
numberVideoPacketsLost	0
videoAverageJitter	7
videoRoundTripTime	0
videoOneWayDelay	0
videoReceptionMetrics	RxCodec=H264;RxBw=377;RxReso=640x360;RxFrameRate=31;RxFramesLost=0
videoTransmissionMetrics	TxCodec=H264;TxBw=368;TxReso=640x360;TxFrameRate=30
videoContentType_channel2	
videoDuration_channel2	
numberVideoPacketsSent_channel2	
numberVideoOctetsSent_channel2	
numberVideoPacketsReceived_channel2	
numberVideoOctetsReceived_channel2	
numberVideoPacketsLost_channel2	
videoAverageJitter_channel2	
videoRoundTripTime_channel2	
videoOneWayDelay_channel2	
videoReceptionMetrics_channel2	

Field Name	Values
videoTransmissionMetrics_channel2	

Related Topics

Call Management Records, on page 187
CDR Examples, on page 15
Cisco Call Management Records K-Factor Data, on page 205
Cisco Call Management Record Field Descriptions, on page 191
Documentation Related to CDR, on page 4
Normal Calls (Cisco Unified IP Phone to Cisco Unified IP Phone), on page 102

INDEX

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V