Cisco Unified Communications Manager Call Detail Records Administration Guide

Release 7.0(1)

Cisco Unified Communications Manager Call Detail Records Administration Guide

C O N T E N T S

Preface

This preface describes the purpose, audience, organization, and conventions of this guide and provides information on how to obtain related documentation.

Note This document may not represent the latest Cisco product information available. You can obtain the most current documentation by accessing Cisco's product documentation page at this URL:

http://www.cisco.com/en/US/products/sw/voicesw/ps556/tsd_products_support_series_home.html

The preface covers these topics:

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.
- *Cisco Unified 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.

Organization

The following table provides the organization of this guide.

Chapter	Description
Chapter 1, "Understanding Cisco Call Detail Records"	Provides an overview of call detail records and an understanding of CDR management.
Chapter 2, "How CDRs Are Processed"	Describes the procedures for how CDRs are processed.
Chapter 3, "Types of Call Information Records"	Provides information on call information records.
Chapter 4, "CDR Examples"	Provides examples of call detail records.
Chapter 5, "Cisco Call Detail Records Field Descriptions"	Describes all call detail record fields.
Chapter 6, "Cisco Call Detail Records Codes"	Provides information on all CDR codes, including call termination cause codes, codec type codes, redirect reason codes, and onbehalfof codes.
Chapter 7, "Understanding Call Management Records"	Provides an overview of call management records (CMRs).
Chapter 8, "Cisco Call Management Records Field Descriptions"	Describes CMR fields.
Chapter 9, "Cisco Call Management Records K-Factor Data"	Describes K-Factor data information in the CMR record.
Chapter 10, "Cisco Call Management Record Examples"	Provides examples of CMRs.

Related Documentation

Refer to the following documents for further information about related Cisco IP telephony applications and products:

- *Installing Cisco Unified Communications Manager Release 7.0(1)*
- *Upgrading Cisco Unified Communications Manager Release 7.0(1)*
- *Cisco Unified Communications Manager Documentation Guide*
- *Release Notes for Cisco Unified Communications Manager Release 7.0(1)*
- *Cisco Unified Communications Manager System Guide*
- *Cisco Unified Communications Manager Features and Services Guide*
- *Cisco Unified Serviceability Administration Guide*
- *Cisco Unified Communications Manager Call Detail Records Administration Guide*
- *Cisco Unified Real-Time Monitoring Tool Administration Guide*
- *Troubleshooting Guide for Cisco Unified Communications Manager*
- *Cisco Unified IP Phone Administration Guide for Cisco Unified Communications Manager*
- *Cisco Unified Communications Manager Bulk Administration Guide*
- *Cisco Unified Communications Manager Security Guide*
- *Cisco Unified Communications Solution Reference Network Design (SRND)*

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.
⟶	This pointer highlights an important line of text in an example.

Convention	Description
^	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.**

Obtaining Documentation and Submitting a Service Request

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.

P A R T 1

Overview

CHAPTER 1

Understanding Cisco Call Detail Records

This chapter describes the format and logic of the call detail records (CDRs) that the Cisco Unified Communications Manager Release 7.0(1) 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 Cisco 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.

This chapter contains the following topics:

- CDR Management, page 1-1
- Upgrading Cisco Unified Communications Manager, page 1-3
- Backup of CDR Database, page 1-4
- Related Topics, page 1-4
- Related Documentation, page 1-4

CDR Management

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

- Collects the CDR/CMR files from the Cisco 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.

Cisco Unified Communications Manager Call Detail Records Administration Guide

- 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. The following sections describe these services:

- CDR Agent, page 1-2
- CDR Repository Manager, page 1-2
- CDR onDemand Service, page 1-3

CDR Agent

As part of the CDRM feature, a resident component on the server or node within a Cisco Unified Communications Manager installation acts as the CDR Agent. On the server or node where both Cisco Unified Communications Manager and the CDR Agent are running, Cisco 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 Cisco 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 Cisco 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 Cisco 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 Cisco 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 Cisco 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 file name 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 Release 7.0(1)*.

Upgrading Cisco Unified Communications Manager

When you upgrade from an earlier version of Cisco Unified Communications Manager to a later version of Cisco 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. Find these documents at
http://www.cisco.com/en/US/products/sw/voicesw/ps556/prod_installation_guides_list.html

Backup of CDR Database

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* at http://www.cisco.com/en/US/products/sw/voicesw/ps556/prod_maintenance_guides_list.html

Related Topics

- See the "Activating CAR" section in the *Configuring CDR Analysis and Reporting Tool* chapter found in the *CDR Analysis and Reporting Administration Guide.*
- *See the Configuring the CDR Repository Manager* chapter found in the *Cisco Unified Serviceability Administration Guide*.
- See the "CMR Processing" section on page 7-1.
- Chapter 2, "How CDRs Are Processed"
- Chapter 3, "Types of Call Information Records"

Related Documentation

The following documents contain additional information related to CDRs:

- *Cisco Unified Serviceability Administration Guide*
- *CDR Analysis and Reporting Administration Guide*
- *Cisco Unified Communications Manager Developers Guide for Release 7.0(1)*
- *Disaster Recovery System Administration Guide*

C H A P T E R 2

How CDRs Are Processed

This chapter provides information on how CDRs are processed and contains the following topics:

Record Processing

Cisco 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 Cisco 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 Cisco 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 Cisco 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 Cisco 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
```

Note   For Cisco Unified Communications Manager Business Edition installations, the value assigned to the clusterId equals 01.

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","origLegCallIdent
ifier",...
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

- Chapter 1, "Understanding Cisco Call Detail Records"
- Chapter 3, "Types of Call Information Records"
- Chapter 8, "Cisco Call Management Records Field Descriptions."

Related Documentation

The following documents contain additional information related to CDRs:

- *CDR Analysis and Reporting Administration Guide*
- *Cisco Unified Serviceability Administration Guide*

C H A P T E R 3

Types of Call Information Records

This chapter describes the two types of call information records that Cisco Unified Communications Manager generates.

Cisco 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 will generate 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 get written to ASCII files at the end of the call. Only completed calls and failed calls generate CDRs and CMRs. Cisco Unified Communications Manager does not perform any post processing on CDRs or CMRs.

This chapter contains the following topics:

- Global Call Identifier, page 3-2
- Number Translations, page 3-2
- Partitions and Numbers, page 3-3
- Timestamps, page 3-4
- Call Clearing Causes, page 3-5
- IP Addresses, page 3-5
- Related Topics, page 3-5
- Related Documentation, page 3-6

Global Call Identifier

The Cisco 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 Cisco Unified Communications Manager server, independent of calls running on other call servers in the cluster. Cisco Unified Communications Manager writes the GlobalCallID_callId value to a disk file for every 1,000th call. When Cisco 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 Cisco Unified Communications Manager restarts, the value for the next call in the CDR gets assigned 2001. The numbers continue sequentially from there until Cisco 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*.

Number Translations

The Cisco Unified Communications Manager can 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" even though 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 Table 3-1:

Table 3-1   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.
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 Cisco 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 Cisco 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.

Table 3-1   Partition/Extension Numbers in CDRs (continued)

Phone Number	Description
finalCalledPartyNumberPartition	This number identifies the partition name that is associated with the FinalCalledPartyNumber field. This field uniquely identifies this number because the Cisco 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.
lastRedirectDnPartition	This number identifies the partition name that is associated with the LastRedirectDn field. This field uniquely identifies this number because the Cisco 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.
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 CDR includes the UTC timestamps that are shown in Table 3-2:

Table 3-2   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.
dateTimeConnect	UTC	This field designates the time that the devices connect and speech begins. This field shows a zero if the call never connects.
dateTimeDisconnect	UTC	This field designates the time that the call disconnects. This field shows a zero if the call never connects.

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 6-2 on page 6-2 lists the call clearing cause code values per ITU specification Q.850. For On Net call legs, the Cisco Unified Communications Manager determines the cause code value. For Off Net call legs, the far-end switch determines the cause code value.

IP Addresses

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:

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

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.

Related Topics

- Chapter 1, "Understanding Cisco Call Detail Records"
- Chapter 2, "How CDRs Are Processed"
- Chapter 7, "Understanding Call Management Records"

Related Documentation

The following documents contain additional information related to CDRs:

- *CDR Analysis and Reporting Administration Guide*
- *Cisco Unified Serviceability Administration Guide*

P A R T 2

Call Detail Records

C H A P T E R 4

CDR Examples

This chapter describes examples of the call detail records (CDRs) that the Cisco Unified Communications Manager Release 7.0(1) 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 Cisco Unified Communications Manager for the change to take effect. The system responds to all changes within a few seconds. .

This chapter contains the following information:

AAC Calls

The Advanced Audio Codec (AAC) specifies a bandwidth voice codec that provides improved voice fidelity. This codec also provides equal or improved sound quality over older codecs with lower bit rates. AAC includes the following features:

- For AAC calls, the codec specifies Media_Payload_AAC 42.
- The maxFramesPerPacket specifies 1.

The system adds an audio bandwidth field to the CDR for AAC 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
G711Ulaw64k	64
G711Ulaw56k	56
G722_64k	64
G722_56k	56
G722_48k	48
G7231	7
G728	16
G729	8
G729AnnexA	8
G729AnnexB	8
G729AnnexAwAnnexB	8
XV150_MR_729A	8
NSE_VBD_729A	8
GSM_Full_Rate	13
GSM-Half_Rate	7
GSM_Enhanced_Full_Rate	13
Wide_Band_256k	256
Is11172AudioCap	0
Is13818AudioCap	0
Data64	64
Data56	56
GSM	13
G7221_32K	32
G7221_24K	24
AAC	256
ILBC	15k or 13k

AAC Calls CDR Example

This example applies to a call with AAC codec:

Field Names	AAC CDR
globalCallID_callId	121
origLegCallIdentifier	101

Cisco Unified Communications Manager Call Detail Records Administration Guide

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

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 were never connected or 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:

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	21
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	0
destTerminationOnBehalfOf	4	4	4	4	4	0
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=1003;ConfControllerDeviceName=SEP0003E333FAD1;ConfRequestorDn-1003; ConfRequestorDeviceName=SEP0003E333FAD1	ConfControllerDn=1003;ConfControllerDeviceName=SEP0003E333FAD1;ConfRequestorDn-1003;ConfRequestorDeviceName=SEP0003E333FAD1

Field Names	CDR7: Ed -> Conference Bridge (conference call)	CDR8 Dave -> Conference Bridge (conference call)	CDR9: Alice -> Conference Bridge (conference call)	CDR10: Bob -> Conference Bridge (conference call)	CDR11: Carol -> Conference Bridge (conference call)
globalCallID_callId	3	1	1	1	1
origLegCallIdentifier	24	25	11	12	14
destLegCallIdentifier	27	28	15	16	17
callingPartyNumber	1004	b0029901222	1000	1001	1002
originalCalledPartyNumber	b0029901222	b0029901001	b0029901001	b0029901001	b0029901001
finalCalledPartyNumber	b0029901222	b0029901001	b0029901001	b0029901001	b0029901001
lastRedirectDn	1003	1002	1001	1001	1001
origTerminationOnBehalfOf	0	0	0	0	0
destTerminationOnBehalfOf	0	0	0	0	0
lastRedirectRedirectReason	98	98	98	98	98
lastRedirectRedirectOnBehalfOf	4	4	4	4	4
origConversationID	0	2222			
destConversationID	2222	1111			
Comment	ConfControllerDn=1003;ConfControllerDeviceName=SEP0003E333FAD1;ConfRequestorDn-1003; ConfRequestorDeviceName=SEP0003E333FAD1	ConfControllerDn=1003;ConfControllerDeviceName=SEP0003E333FAD1;ConfRequestorDn-1003; ConfRequestorDeviceName=SEP0003E333FAD1			

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	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: Dave-> Conference Bridge (conference call)	CDR8: Ed -> Conference Bridge (conference call)	CDR9: Conference Bridge-> Conference Bridge	CDR-10: 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
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	1111	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

Removing a Party from a Linked Conference

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	CDR-10: 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	1000	1001	b0029901222	1000	1001	1004
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

Removing a Party (Controller) from a Linked Conference

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	CDR-10: 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	b0029901222	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
lastRedirectRedirectOnBehalf Of	4	4	10	4	4	4
origConversationID	0	0	2222	0	0	0
destConversationID	1111	1111	1111	2222	2222	2222
Comment	ConfControll erDn=1001;Co nfController DeviceName=S EP0003E333FE BD;ConfReque storDn-1001; ConfRequesto rDeviceName= SEP0003E333F EBD	ConfControll erDn=1001;Co nfController DeviceName=S EP0003E333FE BD;ConfReque storDn-1001; ConfRequesto rDeviceName= SEP0003E333F EBD	ConfControll erDn=1003;Co nfController DeviceName=S EP0003E333FA D1;ConfReque storDn-1003; ConfRequesto rDeviceName= SEP0003E333F AD1	ConfControll erDn=1003;Co nfController DeviceName=S EP0003E333FA D1;ConfReque storDn-1003; ConfRequesto rDeviceName= SEP0003E333F AD1	ConfControll erDn=1003;Co nfController DeviceName=S EP0003E333FA D1;ConfReque storDn-1003; ConfRequesto rDeviceName= SEP0003E333F AD1	ConfControll erDn=1003;Co nfController DeviceName=S EP0003E333FA D1;ConfReque storDn-1003; ConfRequesto rDeviceName= SEP0003E333F AD1

Removing the Linked Conference

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	CDR-10: 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	2222	0	0	0	0	0

destConversationID	1111	1111	1111	2222	2222	0
Comment	ConfControll erDn=1003;Co nfController DeviceName=S EP0003E333FA D1;ConfReque storDn-1003; ConfRequesto rDeviceName= SEP0003E333F AD1	ConfControll erDn=1001;Co nfController DeviceName=S EP0003E333FE BD;ConfReque storDn-1001; ConfRequesto rDeviceName= SEP0003E333F EBD	ConfControll erDn=1001;Co nfController DeviceName=S EP0003E333FE BD;ConfReque storDn-1001; ConfRequesto rDeviceName= SEP0003E333F EBD	ConfControll erDn=1003;Co nfController DeviceName=S EP0003E333FA D1;ConfReque storDn-1003; ConfRequesto rDeviceName= SEP0003E333F AD1	ConfControll erDn=1003;Co nfController DeviceName=S EP0003E333FA D1;ConfReque storDn-1003; ConfRequesto rDeviceName= SEP0003E333F AD1	ConfControll erDn=1003;Co nfController DeviceName=S EP0003E333FA D1;ConfReque storDn-1003; ConfRequesto rDeviceName= SEP0003E333F AD1

Field Names	CDR13: Dave -> Ed
globalCallID_callId	3
origLegCallIdentifier	21
destLegCallIdentifier	24
callingPartyNumber	1003
originalCalledPartyNumber	b0029901222
finalCalledPartyNumber	1004
lastRedirectDn	b0029901222
origTerminationOnBehalfOf	0
destTerminationOnBehalfOf	0
lastRedirectRedirectReason	98
lastRedirectRedirectOnBehalfOf	4
origConversationID	0
destConversationID	0
Comment	ConfControllerDn=10 03;ConfControllerDe viceName=SEP0003E33 3FAD1;ConfRequestor Dn-1003;ConfRequest orDeviceName=SEP000 3E333FAD1

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
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 CDR	Final Call 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
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	Barge 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			
origRedirectOnBehalfOf		15	15
lastRedirectRedirectOnBehalfOf		15	15
joinOnBehalfOf		15	15
destConversationID	0	16777250	16777251

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
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

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. This section contains the following CDR examples:

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
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
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.

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
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 will have 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 will have 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.

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.

Field Names	Original Call CDR	Pickup CDR
globalCallID_callId	11	11
origLegCallIdentifier	12345	12345
destLegCallIdentifier	12346	12347
callingPartyNumber	9728134987	9728134987
originalCalledPartyNumber	2001	2002
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.

Call Recording CDR Examples

1. The customer (9728134987) calls the agent (30000), and the agent answers. 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	Monitored Call CDR	Monitoring Call CDR
globalCallID_callId	7	10
origLegCallIdentifier	16777110	16777120
destLegCallIdentifier	16777111	16777121
callingPartyNumber	9728134987	30000
originalCalledPartyNumber	30000	90000
finalCalledPartyNumber	30000	90000
lastRedirectDn	30000	90000
origCause_Value	16	0
dest_CauseValue	0	0
origCalledPartyRedirectReason	0	354
lastRedirectRedirectOnBehalfOf	0	354
origCalledPartyRedirectOnBehalfOf		27
lastRedirectRedirectOnBehalfOf		27
destConversationID	0	16777111

2. The agent (30000) calls the customer (9728134987), and the customer answers. 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	Monitored Call CDR	Monitoring Call CDR
globalCallID_callId	71	100
origLegCallIdentifier	16777113	16777220
destLegCallIdentifier	16777114	16777221
callingPartyNumber	30000	30000
originalCalledPartyNumber	9728134987	90000
finalCalledPartyNumber	9728134987	90000
lastRedirectDn	9728134987	90000
origCause_Value	16	16
dest_CauseValue	0	0
origCalledPartyRedirectReason	0	354
lastRedirectRedirectOnBehalfOf	0	354
origCalledPartyRedirectOnBehalfOf		27
lastRedirectRedirectOnBehalfOf		27
destConversationID	0	16777113

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

Calling Party Normalization

This feature provides the support of the international escape code "+" to Cisco 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

Field Names	Values
destLegCallIdentifier	101
callingPartyNumber	+19725001212
outpulsedCallingPartyNumber	5001212
duration	60

2. 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

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
duration	0

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

Cisco Unified Communications Manager Call Detail Records Administration Guide

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
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

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
clientMatterCode	11111
duration	600

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 get 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 Cisco 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	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
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	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	

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 Cisco 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

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
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

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).

Table 1:

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
dest_CauseValue	16
origDTMFMethod	0
destDTMFMethod	0
duration	60

2. **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

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

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

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
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
lastRedirectRedirectOnBehalfOf	5
duration	30

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
G711Ulaw64k	64
G711Ulaw56k	56
G722_64k	64
G722_56k	56
G722_48k	48
G7231	7
G728	16
G729	8
G729AnnexA	8
G729AnnexB	8
G729AnnexAwAnnexB	8
XV150_MR_729A	8
NSE_VBD_729A	8
GSM_Full_Rate	13
GSM-Half_Rate	7
GSM_Enhanced_Full_Rate	13
Wide_Band_256k	256
Is11172AudioCap	0
Is13818AudioCap	0
Data64	64
Data56	56
GSM	13

G7221_32K	32
G7221_24K	24
AAC	256
ILBC	15k or 13k

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

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
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

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

Cisco Unified Communications Manager Call Detail Records Administration Guide

| 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
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

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
lastRedirectRedirectOnBehalfOf	16
duration	120

Local Route Groups and Called Party Transformation

In this release, Cisco 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
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

Malicious Calls

When a call gets identified as a malicious call (button press), the local Cisco 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"

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 Cisco 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.

Table 2:

	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

Mobility

Cisco Unified Communications Manager supports the following Mobility features:

- Hand-In
- Hand-Out
- Cell Pickup
- Interactive Voice Response (IVR)

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 the Mobility feature. The CAR Loader checks the following **OnBehalfOf** fields:

- **origCallTerminationOnBehalfOf**
- **destCallTerminationOnBehalfOf**
- **origCalledPartyRedirectOnBehalfOf**
- **lastRedirectRedirectOnBehalfOf**
- **joinOnBehalfOf**

If any of the preceding **OnBehalfOf** codes has the Mobility code of 24, the CDR has the Mobility call type that the CAR Loader determines. Four redirectResource codes apply for Mobility features: Hand-In (code 303), Hand-Out (code 319), Cell Pickup (code 335), and IVR (code 399).

Mobility CDR Examples

1. **Mobility Follow Me** - A dual-mode phone has the Enterprise number of 22285 and the cell number of 9728324124. 22202 calls 22285, and both 22285 and 9728324124 ring. The cell phone answers the call. The system generates a single CDR for this Follow Me call. The parties talk for 80 seconds.

Field Names	Follow Me Call CDR
globalCallID_callId	861
origLegCallIdentifier	22481077
destLegCallIdentifier	22481078
callingPartyNumber	22202
originalCalledPartyNumber	22285
finalCalledPartyNumber	9728324124
lastRedirectDn	22285
origCause_Value	16
dest_CauseValue	0
lastRedirectRedirectReason	0
lastRedirectRedirectOnBehalfOf	0
origTerminationOnBehalfOf	
destTerminationOnBehalfOf	
joinOnBehalfOf	0
duration	80

2. **Mobility HandIn** - A dual-mode phone with the Enterprise number of 22285 and the cell number of 9728324124 calls to the cell phone 9728324214. They talk for 39 seconds; then, the dual-mode phone gets carried into the Enterprise network, and the call gets switched from the cell network to the Enterprise network. The parties continue to talk for another 15 seconds.

Field Names	Call to cell #9728324214 CDR	HandIn Call to the Enterprise CDR
globalCallID_callId	864	864
origLegCallIdentifier	22481083	22481083
destLegCallIdentifier	22481085	22481087
callingPartyNumber	22202	22202
originalCalledPartyNumber	919728324124	22285
finalCalledPartyNumber	919728324124	22285
lastRedirectDn	919728324124	22285
origCause_Value	393216	0
dest_CauseValue	393216	16
lastRedirectRedirectReason	0	303
lastRedirectRedirectOnBehalfOf	0	24
origTerminationOnBehalfOf	24	24
destTerminationOnBehalfOf	24	12
joinOnBehalfOf	0	24
duration	39	15

3. **Mobility HandOut** - A dual-mode phone has the Enterprise number of 22285 and the cell number of 9728324124. The handout number (H-number) specifies 555123. A call goes to the Enterprise number 22285. They talk for 21 seconds; then, the dual-mode phone gets carried out of the Enterprise network and into the cell network. The call gets switched from the Enterprise network to the cell network (9728324124). The parties continue to talk for another 39 seconds.

Field Names	Enterprise Call to 22285 CDR	Server Call from cell phone to H-Number CDR	Handout Call CDR
globalCallID_callId	964	965	964
origLegCallIdentifier	22481083	22481095	22481093
destLegCallIdentifier	22481094	22481096	22481095
callingPartyNumber	22202	9728324124	22202
originalCalledPartyNumber	22285	555123	9728324124
finalCalledPartyNumber	22285	555123	9728324124
lastRedirectDn	22285	555123	9728324124
origCause_Value	393216	393216	0
dest_CauseValue	393216	393216	16

Field Names			
lastRedirectRedirectReason	0	0	319
lastRedirectRedirectOnBehalfOf	0	0	24
origTerminationOnBehalfOf	24	24	24
destTerminationOnBehalfOf	24	24	12
joinOnBehalfOf	0	0	24
duration	21	0	39

4. **Mobility Cell Pickup** - A dual-mode phone with the Enterprise number of 22285 and the cell number of 9728324124, establishes a call to the Enterprise number 22285. They talk for 40 seconds; then, **Cell Pickup** gets invoked. The call gets switched from the Enterprise phone to the cell phone. The parties continue to talk for another 111 seconds.

Field Names	Enterprise Call to 22285 CDR	Server Call to Cell Phone CDR	Final Handout Call CDR
globalCallID_callId	555	566	964
origLegCallIdentifier	22481111	22481222	22481111
destLegCallIdentifier	22481112	22481223	22481222
callingPartyNumber	22202		22202
originalCalledPartyNumber	22285	9728324124	9728324124
finalCalledPartyNumber	22285	9728324124	9728324124
lastRedirectDn	22285	9728324124	9728324124
origCause_Value	393216	0	0
dest_CauseValue	393216	0	16
lastRedirectRedirectReason	0	0	335
lastRedirectRedirectOnBehalfOf	0	0	24
origTerminationOnBehalfOf	24	24	24
destTerminationOnBehalfOf	24	24	12
joinOnBehalfOf	0	0	24
duration	40	0	111

5. **Mobility IVR** - A call comes into the Cisco Unified Communications Manager with string DID#RemoteDest#TargetNum#. The call gets redirected to the TargetNum. 9728131234 calls into an IVR, and data gets collected. The target destination specifies 812345, and the call gets redirected to 812345. The call remains connected for 60 seconds.

Field Names	Redirected Call CDR
globalCallID_callId	12345
origLegCallIdentifier	16677100
destLegCallIdentifier	16677102
callingPartyNumber	9728131234
originalCalledPartyNumber	8005559876
finalCalledPartyNumber	812345
lastRedirectDn	8005559876
origCause_Value	0
dest_CauseValue	16
lastRedirectRedirectReason	399
lastRedirectRedirectOnBehalfOf	24
origTerminationOnBehalfOf	0
destTerminationOnBehalfOf	0
duration	60

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

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 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).

Table 3:

Call	CallingPartyNumber	originalCalledPartyNumber
1	4001	4002

Table 3:

Call	CallingPartyNumber	originalCalledPartyNumber
2	4002	4003
3	4001	4003

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

Personal Assistant Calls

This section contains information about the following Personal Assistant Calls:

Personal Assistant Direct Call

A personal assistant direct call acts similar to the Blind Transfer from the Calling Party call type. See the "Blind Transfer from the Calling Party CDR Example" section on page 4-64.

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.

Table 4:

Calling Party Num	Orig LegCall 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

Personal Assistant Interceptor Going to Media Port and Transferring the Call

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

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.

Table 5:

Calling Party Num	Orig LegCall 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

Personal Assistant Interceptor Going Directly to Destination

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

Personal Assistant Interceptor Going Directly to Destination with No Rules 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, 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:

Table 6:

Calling Party Number	OrigLeg Call Identifier	Calling Party Number Partition	DestLeg Identifier	Final Called Party Number	Final Called 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

Personal Assistant Going Directly to Destination with Rule to Forward Calls to a Different Destination 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 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.

Table 7:

Calling Party Number	OrigLeg Call Identifier	Calling Party Number Partition	DestLeg Identifier	Final Called Party Number	Final Called 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 tables contain example CDRs for each of these scenarios:

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.

Table 8:

Calling Party Num	Orig LegCall 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.

Table 9:

Calling Party Num	Orig LegCall 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.

Table 10:

Calling Party Num	Orig LegCall 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

Table 10:

Calling Party Num	Orig LegCall 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	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.

Table 11:

Calling Party Num	Orig LegCall 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.

Table 12:

Calling Party Num	Orig LegCall 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.

Table 13:

Calling Party Num	Orig LegCall 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. For more information, see the "Conference Calls" section on page 4-29.

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.

Table 1

Calling Party Num	Orig LegCall 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.

Table 14:

Original Called Party 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

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 Cisco 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

Refer Calls

See the for an example of Refer with Replaces.

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 Cisco 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 Cisco 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
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 Cisco 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
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 Called Partition	Orig Cause	Dest Cause	DateTime Connect	Duration
2001	Accounts	2309	Marketing	0	16	973795815	0

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

Table 15:

	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
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
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
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 Cisco 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

FieldNames	Orig Call CDR	Setup Call CDR	Conference CDR 1	Conference CDR 2	Conference CDR 3	Final CDR
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
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	

Related Topics

- Chapter 5, "Cisco Call Detail Records Field Descriptions"
- Chapter 6, "Cisco Call Detail Records Codes"
- Chapter 10, "Cisco Call Management Record Examples"

Related Documentation

The following documents contain additional information related to CDRs:

- *Cisco Unified Serviceability Administration Guide*
- *CDR Analysis and Reporting Administration Guide*
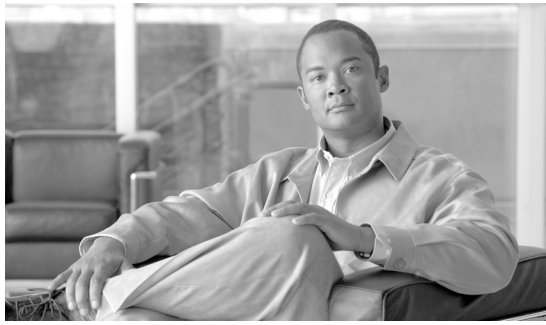
C H A P T E R 5

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. This chapter contains the following information:

- CDR Field Descriptions, page 5-1
- Related Topics, page 5-21
- Related Documentation, page 5-22

CDR Field Descriptions

Table 5-1 describes all fields in the current CDRs in the order in which they appear.

Table 5-1   CDR Field Descriptions

Field Name	Range of Values	Description
cdrRecordType	0, 1, 2	This field 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	This field designates a unique Cisco 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 this field always is populated.

Table 5-1   CDR Field Descriptions (continued)

Field Name	Range of Values	Description
globalCallID_callId	Positive Integer	This field 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 Cisco 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. See the "Global Call Identifier" section on page 3-2 for additional information.
		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.
		Default - Ensure this field always is populated.
origLegCallIdentifier	Positive Integer	This field 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 sub-calls, and consequently several CDRs (as during a call transfer), this value remains constant.
		Default - Ensure this field always is populated.
dateTimeOrigination	Integer	This field identifies the date and time when the user goes off 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 this field always is populated.
origNodeId	Positive Integer	This field 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 this field always is populated.

Table 5-1   CDR Field Descriptions (continued)

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	This field identifies the IP address of the device that originated the call signaling. For Cisco Unified IP Phones, this field specifies the address of the phone. For PSTN calls, this field specifies the address of the H.323 gateway. For intercluster calls, this field specifies the address of the remote Cisco Unified Communications Manager. The "IP Addresses" section on page 3-5 describes the IP address format. Default - This field gets populated based on these rules.

Table 5-1   *CDR Field Descriptions (continued)*

Field Name	Range of Values	Description
callingPartyNumber	Text String	This field specifies numeric string of up to 25 characters.
		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 Cisco Unified Communications Manager (such as translations at the gateway).
		For server calls, where Cisco 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	This field 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.
origCause_location	0 to 15	For clearing causes that are received over ISDN signaling links, this field specifies the Location field that is indicated in the ISDN release message. The "Call Termination Cause Codes" section on page 6-2 lists the valid values per Q.850.
		For clearing causes that are created internally by the Cisco Unified Communications Manager, this value specifies zero.
		Default - 0

Table 5-1   CDR Field Descriptions (continued)

Field Name	Range of Values	Description
origCause_value	0 to 129	For calls that are cleared by the originating party, this field reflects the reason for clearance.
		Cisco Unified Communications Manager currently uses the Q.850 codes and some Cisco Unified Communications Manager defined codes. The "Call Termination Cause Codes" section on page 6-2 lists them.
		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/conference), the CDR terminates, and this field gets set to 393216. This represents a proprietary value for this field.
		Default - 0
origPrecedenceLevel	0 to 4	For MLPP, each call leg includes a precedence level. This field represents the precedence level of the original leg.
		• Precedence 0 = FLASH OVERRIDE/ EXECUTIVE OVERRIDE
		• Precedence 1 = FLASH
		• Precedence 2 = IMMEDIATE
		• Precedence 3 = PRIORITY
		• Precedence 4 = ROUTINE
		Default - 4
origMediaTransportAddress_IP	0, Integer	This field identifies the IP address of the device that originates the media for the call.
		For Cisco Unified IP Phones, this field specifies the address of the phone.
		For PSTN calls, this field specifies the address of the H.323 gateway.
		For intercluster calls, this field specifies the address of the remote phone.
		The "IP Addresses" section on page 3-5 describes the IP address format.
		Default - 0. If media is not established, this field equals 0.

Table 5-1   *CDR Field Descriptions (continued)*

Field Name	Range of Values	Description
origMediaTransportAddress_Port	0, Positive Integer	This field identifies the IP port number that is associated with the OrigMediaTransportAddress_IP field. Default - 0. If media is not established, this field stays 0.
origMediaCap_payloadCapability	0, Positive Integer	This field identifies the codec type that the originator uses to transmit media. Cisco Unified Communications Manager currently uses the following payload capability values: 0, 1-16, 18-20, 25, 32, 33, 81-86. The "Codec Types" section on page 6-1 lists the valid values. Default - 0. If media is not established, this field stays 0.
origMediaCap_maxFramesPerPacket	0, Positive Integer	This field 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 Cisco Unified Communications Manager. This field will remain 0.
origVideoCap_Codec	0, 100 = H.261, 101 = H.263, 102 = Vieo	This field identifies the codec type that the originator uses to transmit video (H.261, H.263, or Vieo.) Default - 0. If media is not established, this field stays 0.
origVideoCap_Bandwidth	0, Positive Integer	This field identifies the bandwidth that is measured in units of kbps. Default - 0. If media is not established, this field stays 0.
origVideoCap_Resolution	0, 1 = SQCIF, 2 = QCIF, 3 = CIF, 4 = CIF4, 5 = CIF16	This field identifies the video resolution. Default - 0. If media is not established, this field stays 0.

Table 5-1   ***CDR Field Descriptions (continued)***

Field Name	Range of Values	Description
origVideoTransportAddress_IP	0, Integer	This field identifies the IP address of the device that originates the call. Default - 0. If media is not established, this field stays 0.
origVideoTransportAddress_Port	0, Positive Integer	This field identifies the video RTP port that is associated with the origVideoTransportAddress_IP field. Default - 0. If media is not established, this field stays 0.
origRSVPAudioStat	0 to 5	This field gives 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 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

Table 5-1   CDR Field Descriptions (continued)

Field Name	Range of Values	Description
origRSVPVideoStat	0 to 5	This field gives 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	This field 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	This field 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.

Table 5-1   CDR Field Descriptions (continued)

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	This field identifies the IP address of the device that terminates the call signaling.
		For Cisco Unified IP Phones, this field specifies the address of the phone.
		For PSTN calls, this field specifies the address of the H.323 gateway.
		For intercluster calls, this field specifies the address of the remote Cisco Unified Communications Manager.
		The "IP Addresses" section on page 3-5 describes the IP address format.
		Default - 0. If the destination cannot be reached, this field stays 0.
originalCalledPartyNumber	Text String	This field 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.
		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, this field stays empty.

Table 5-1   *CDR Field Descriptions (continued)*

Field Name	Range of Values	Description
finalCalledPartyNumber	Text String	This field specifies the 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. 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 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, this field stays empty.
finalCalledPartyUnicodeLoginUserID	Unicode – UTF_8	The final called party field 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. The "Call Termination Cause Codes" section on page 6-2 lists the valid values per Q.850. For clearing causes that Cisco Unified Communications Manager creates internally, this value equals zero. Default - 0. If the destination cannot be reached, this field stays 0.

Table 5-1   CDR Field Descriptions (continued)

Field Name	Range of Values	Description
destCause_value	0 to 129	For calls that the destination party cleared, this field reflects the reason for the clearance. The "Call Termination Cause Codes" section on page 6-2 lists 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/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.
destPrecedenceLevel	0 to 4	For MLPP, each call leg has a precedence level. This field represents the destination legs precedence level.
		• Precedence 0 = FLASH OVERRIDE
		• Precedence 1 = FLASH
		• Precedence 2 = IMMEDIATE
		• Precedence 3 = PRIORITY
		• Precedence 4 = ROUTINE
		Default - 4
destMediaTransportAddress_IP	0, Integer	This field identifies the IP address of the device that terminates the media for the call.
		For Cisco Unified IP Phones, this field designates the address of the phone.
		For PSTN calls, this field designates the address of the H.323 gateway.
		For intercluster calls, this field shows the address of the remote phone.
		The "IP Addresses" section on page 3-5 describes the IP address format.
		Default - 0. If the destination cannot be reached, this field stays 0.

Table 5-1   CDR Field Descriptions (continued)

Field Name	Range of Values	Description
destMediaTransportAddress_Port	0, Positive Integer	This field 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.
destMediaCap_payloadCapability	0, Positive Integer	This field identifies the codec type that the terminating party uses to transmit media. Cisco Unified Communications Manager currently uses the following payload capability values: 0, 1-16, 18-20, 25, 32, 33, 81-86. The "Codec Types" section on page 6-1 lists the valid values. Default - 0. If the destination cannot be reached, this field stays 0.
destMediaCap_maxFramesPerPacket	0, Positive Integer	This field 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 Cisco Unified Communications Manager. Default - This field stays 0.
destVideoCap_Codec	0, 100 = H.261, 101 = H.263, 102 = Vieo	This field identifies the codec type that the terminating party uses to transmit video (H.261, H.263, or Vieo). Default - 0. If the destination cannot be reached, this field stays 0.
destVideoCap_Bandwidth	0, Positive Integer	This field identifies the bandwidth, and is measured in units of kbps. Default - 0. If the destination cannot be reached, this field stays 0.

Table 5-1 **CDR Field Descriptions (continued)**

Field Name	Range of Values	Description
destVideoCap_Resolution	0, 1 = SQCIF, 2 = QCIF, 3 = CIF, 4 = CIF4, 5 = CIF16	This field identifies the video resolution. Default - 0. If the destination cannot be reached, this field stays 0.
destVideoTransportAddress _IP	0, Integer	This field identifies the IP address of the device that receives the call. Default - 0. If the destination cannot be reached, this field stays 0.
destVideoTransportAddress_Port	0, Positive Integer	This field 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.
destRSVPAudioStat	0 - 5	This field designates the status of the RSVP audio 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

Table 5-1   CDR Field Descriptions (continued)

Field Name	Range of Values	Description
destRSVPVideoStat	0 - 5	This field 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	This field 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	0, Integer	This field 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 - 0. If the call is never connected, this field stays 0.

Table 5-1   CDR Field Descriptions (continued)

Field Name	Range of Values	Description
lastRedirectDn	Text String	This field 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.
		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, this field remains empty.
pkid	Text String	This field 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.
originalCalledPartyNumberPartition	Text String	This field uniquely identifies the partition name that is associated with the OriginalCalledPartyNumber field because Cisco 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.

Table 5-1   CDR Field Descriptions (continued)

Field Name	Range of Values	Description
callingPartyNumberPartition	Text String	This field uniquely identifies the partition name that is associated with the CallingPartyNumber field because Cisco 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.
finalCalledPartyNumberPartition	Text String	This field uniquely identifies the partition name that is associated with the FinalCalledPartyNumber field because Cisco 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	This field uniquely identifies the partition name that is associated with the LastRedirectDn field because Cisco 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.

Table 5-1   CDR Field Descriptions (continued)

Field Name	Range of Values	Description
duration	0, Positive integer	This field 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 |
| origDeviceName | Text String | This field specifies the text string that identifies the name of the originating device.

Default - Ensure this field always is populated. |
| destDeviceName | Text String | This field 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 | This field 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 the "Related Topics" section on page 5-21 for a list of the codes. This release added new OnBehalfOf codes.

Default - 0 |
| destCallTerminationOnBehalfOf | 0, Positive Integer | This field specifies code that identifies why the destination 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 the "Related Topics" section on page 5-21 for a list of the codes. This release added new OnBehalfOf codes.

Default - 0 |

Table 5-1   CDR Field Descriptions (continued)

Field Name	Range of Values	Description
origCalledPartyRedirectOnBehalfOf	0, Positive Integer	This field 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 the "Related Topics" section on page 5-21 for a list of the codes. This release added new OnBehalfOf codes.
		Default - 0
lastRedirectRedirectOnBehalfOf	0, Integer	This field 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 the "Related Topics" section on page 5-21 for a list of the codes. This release added new OnBehalfOf codes.
		Default - 0
origCalledPartyRedirectReason	0, Integer	This field identifies the reason for a redirect of the original called party.
		See the "Redirect Reason Codes" section on page 6-6 for a complete list of the codes.
		Default - 0
lastRedirectRedirectReason	0, Integer	This field identifies the last redirect reason for redirection.
		See the "Redirect Reason Codes" section on page 6-6 for a complete list of the codes.
		Default: 0
destConversationID	0, Integer	This field 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

Table 5-1   CDR Field Descriptions (continued)

Field Name	Range of Values	Description
globalCallId_ClusterId	Text String	This field specifies a unique ID that identifies a cluster of Cisco Unified Communications Managers.
		The field is generated at installation and is not used by Cisco 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	This field 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 the "Related Topics" section on page 5-21 for a list of the codes.
		Default: 0
Comment	Text String	This field 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	This field provides a description of the FAC.
		Default: Empty string "" or null.
authorizationLevel	0, Integer	This field displays the level of the FAC.
		Default: 0
clientMatterCode	Text String	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. This field displays the client matter code.
		Default: Empty string "" or null.

Table 5-1   CDR Field Descriptions (continued)

Field Name	Range of Values	Description
origDTMFMethod	0, Positive Integer	This field 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	This field 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)
callSecuredStatus	0, Positive Integer	This field displays the highest security status that is reached during a call. For example, if the call is originally unsecured, then 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)

Table 5-1   CDR Field Descriptions (continued)

Field Name	Range of Values	Description
origConversationID	Integer	This field 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	This field displays the media bandwidth that is used at the origination of the call. Default: 0
destMediaCap_Bandwidth	0, Positive Integer	This field displays the media bandwidth that is used at the destination of the call. Default: 0
authorizationCodeValue	Text String	This field displays the Forced Authorization Code (FAC) that is associated with the call. Default: Empty string "" or null.
outpulsedCallingPartyNumber	Text String	This field 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.
outpulsedCalledPartyNumber	Text String	This field 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.

Related Topics

- Chapter 4, "CDR Examples"

- Chapter 8, "Cisco Call Management Records Field Descriptions"

Related Documentation

The following documents contain additional information that is related to CDRs:

- *Cisco Unified Serviceability Administration Guide*
- *CDR Analysis and Reporting Administration Guide*

P A R T 3

CDR Codes

C H A P T E R 6

Cisco Call Detail Records Codes

This chapter describes the codec types and codes that are used in the Call Detail Record fields. This chapter contains the following information:

Codec Types

Table 6-1 contains the compression and payload types that may appear in the codec fields.

Table 6-1   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
13	Is11172AudioCap
14	Is13818AudioCap

Table 6-1   Codec Types (continued)

Value	Description
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
80	GSM
81	ActiveVoice
82	G726_32K
83	G726_24K
84	G726_16K
86	iLBC
100	H261
101	H263
102	Vieo
103	H264
106	H224

Call Termination Cause Codes

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

- "Call Termination Cause Codes"
- "Cisco-Specific Call Termination Cause Codes"

Table 6-2   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

Table 6-2   Call Termination Cause Codes (continued)

Code	Description
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 (user alerted)
20	Subscriber absent
21	Call rejected
22	Number changed
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

Table 6-2   Call Termination Cause Codes (continued)

Code	Description
63	Service or option not available, unspecified
65	Bearer capability not implemented
66	Channel type not implemented
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
123	Device not Preemptable
125	Out of bandwidth (Cisco specific)
126	Call split (Cisco specific)
127	Interworking, unspecified
129	Precedence out of bandwidth

Table 6-3   *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
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

Table 6-3   *Cisco-Specific Call Termination Cause Codes (continued)*

Decimal Value Code	Hex Value Code	Description
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
1476395110	0x58000066	CCM_SIP__504_SERVER_TIME_OUT
1493172351	0x5900007F	CCM_SIP_505_SIP_VERSION_NOT_SUPPORTED
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

Table 6-4 contains the available Redirect Reason Codes that may appear in a record.

Table 6-4   *Redirect Reason Codes*

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

Table 6-4   Redirect Reason Codes (continued)

Q.931 Standard Redirect Reason Codes

82	Call Forward On Failure
98	Conference
114	Barge
129	Aar
130	Refer
146	Replaces
162	Redirection (3xx)
177	SIP-forward busy greeting
207	Follow Me (SIP-forward all greeting)
209	Out of Service (SIP-forward busy greeting)
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
354	Recording
370	Monitoring
399	Mobility IVR
415	Mobility Cell Pickup
418	Click to Conference

OnBehalfof Codes

Table 6-5 contains the available OnBehalfof Codes that may appear in a CDR record.

Table 6-5   OnBehalfof Codes

Value	Description
0	Unknown
1	CctiLine
2	Unicast Shared Resource Provider
3	Call Park
4	Conference
5	Call Forward
6	Meet-Me Conference

Table 6-5   OnBehalfof Codes (continued)

Value	Description
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

Related Topics

- Chapter 4, "CDR Examples"
- Chapter 5, "Cisco Call Detail Records Field Descriptions"

Related Documentation

The following documents contain additional information related to CDRs:

- *Cisco Unified Serviceability Administration Guide*
- *CDR Analysis and Reporting Administration Guide*

P A R T   4

Call Management Records

C H A P T E R 7

Understanding Call Management Records

This chapter describes the format and logic of the call management records (CMRs) that the Cisco Unified Communications Manager Release 7.0(1) system generates. You can use this information for post-processing activities such as generating billing records and network analysis. The chapter describes how to access the CMR files.

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 Cisco 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.

This chapter contains the following topics:

CMR Processing

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

When Cisco 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 Cisco 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.

The Cisco 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 Cisco 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

Note   For Cisco Unified Communications Manager Business Edition installations, the value that is assigned to the clusterId equals 01.

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","origLegCallIdent
ifier",...
Line2-INTEGER,INTEGER,INTEGER,INTEGER,...
Line3-1,1,388289,17586046,...
Line4-1,1,388293,17586054,...
```

Configuring CMRs

You can configure CMRs on the Service Parameters Configuration window in Cisco Unified Communications Manager Administration. To access the Service Parameters Configuration window, open Cisco Unified Communications Manager 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. Table 7-1 displays the results of these tests.

Note   Be aware that these tests were performed with Cisco Unified Communications Manager Release 7.0(1).

Table 7-1   CDR and CMR CPU Utilization

CDRs and CMRs Enabled/Disabled	Average % Increase in Cisco Unified CM CPU Utilization	Average % Increase in Total CPU Utilization	Average % I/O wait	% Increase in Cisco Unified CM CPU	% Increase in Total CPU	% Increase I/O wait
CDRs disabled, CMRs disabled	6.5	9.7	0.41	-	-	-
CDRs enabled, CMRs disabled	6.7	10.0	0.44	3.2	4.2	8.1
CDRs disabled, CMRs enabled	6.7	10.0	0.43	2.5	3.3	5.2
CDRs enabled, CMRs enabled	6.9	10.4	0.45	6.3	7.8	12.1

Related Topics

- Chapter 8, "Cisco Call Management Records Field Descriptions"
- Chapter 9, "Cisco Call Management Records K-Factor Data"
- Chapter 10, "Cisco Call Management Record Examples"

Related Documentation

The following documents contain additional information related to CMRs:

- *Cisco Unified Serviceability Administration Guide*
- *CDR Analysis and Reporting Administration Guide*
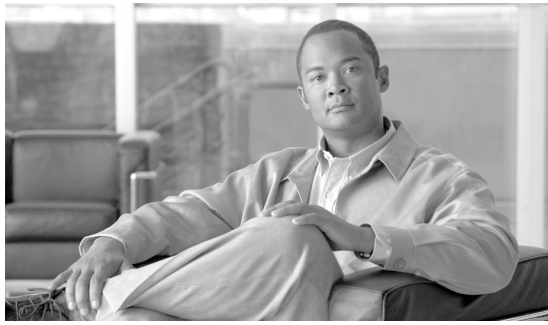
C H A P T E R 8

Cisco Call Management Records Field Descriptions

This chapter describes the field descriptions of the Call Management Records (CMRs). The chapter contains the following information:

- CMR Field Descriptions, page 8-1
- Related Topics, page 8-6
- Related Documentation, page 8-6

CMR Field Descriptions

Table 8-1 contains the fields, range of values, and field descriptions of the CMRs in the order in which they appear in the CMR.

Table 8-1   CMR Field Descriptions

Field Name	Range of Values	Description
cdrRecordType	0, 1, or 2	This field 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.

Table 8-1   CMR Field Descriptions (continued)

Field Name	Range of Values	Description
globalCallID_callManagerId	Positive Integer	This field specifies a unique Cisco 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 this field always is populated.
globalCallId_callId	Positive Integer	This field 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 this field always is populated.
nodeId	Positive Integer	This field specifies the server, or node within the Cisco Unified Communications Manager cluster, where this record gets generated. Default - Ensure this field always is populated.
callIdentifier	Positive Integer	This field identifies the call leg to which this record pertains. Default - Ensure this field always is populated.

Table 8-1   CMR Field Descriptions (continued)

Field Name	Range of Values	Description
directoryNumber	Integer	This field specifies the directory number of the device from which these diagnostics are collected. Default - Ensure this field always is populated.
dateTimeStamp	Integer	This field represents the approximate time that the device goes on hook. Cisco Unified Communications Manager records the time when the phone responds to a request for diagnostic information. Default - Ensure this field always is populated.
numberPacketsSent	Integer	This field 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	This field 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
numberPacketsReceived	Integer	This field 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

Table 8-1   *CMR Field Descriptions (continued)*

Field Name	Range of Values	Description
numberOctetsReceived	Integer	This field 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	This field 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 actually 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
jitter	Integer	This field provides an estimate of the statistical variance of the RTP data packet interarrival time, 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 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

Table 8-1   CMR Field Descriptions (continued)

Field Name	Range of Values	Description
latency	Integer	This field 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 indicates and the NTP timestamp of the receivers, measured when these messages are received. Cisco 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
pkid	Text String	This field 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	This field identifies the partition of the directory number. Default - Empty string, "". This field may remain empty if no partition exists.
deviceName	Text String	This field identifies the name of the device. Default - Empty string "". This field may remain empty if no device name exists.
globalCallId_ClusterId	Text String	This field designates a unique ID that identifies a single Cisco Unified Communications Manager, or a cluster of Cisco Unified Communications Managers. The system generates this field during installation, but Cisco Unified Communications Manager does not use it: globalCallId_ClusterId + globalCallId_callManagerId + globalCallId_callId. Default - Ensure this field always is populated.

Table 8-1   CMR Field Descriptions (continued)

Field Name	Range of Values	Description
varVQMetrics	Text String	This field 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=0.95;CCR=0.0000;ICR=0.0000;ICRmx=0.0000;CS=0;SCS=0"
		Note See Table 9-1 "K-Factor Data Stored in Cisco Unified Communications Manager CMRs" for a complete list of K-Factor data.

Related Topics

- Chapter 5, "Cisco Call Detail Records Field Descriptions"
- Chapter 7, "Understanding Call Management Records"
- Chapter 9, "Cisco Call Management Records K-Factor Data"
- Chapter 10, "Cisco Call Management Record Examples"

Related Documentation

The following documents contain additional information related to CMRs:

- *Cisco Unified Serviceability Administration Guide*
- *CDR Analysis and Reporting Administration Guide*

<Chapter>

C H A P T E R 9

Cisco Call Management Records K-Factor Data

This chapter describes the K-factor data that is present in the Cisco call management records (CMRs). This chapter contains the following information:

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.

Table 9-1 displays the K-factor date that is stored in the Cisco Unified Communications Manager CMRs.

Table 9-1   *K-Factor Data Stored in Cisco 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.
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.

Table 9-2 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 9-2 *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 Phone 7931	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 Phone 7960	S
Cisco Unified IP Phone 7961	X
Cisco Unified IP Phone 7962-G	X
Cisco Unified IP Phone 7962-G/GE	X
Cisco Unified IP Phone 7965	X
Cisco Unified IP Phone 7970	X
Cisco Unified IP Phone 7971	X
Cisco Unified IP Phone 7972-G/GE	X
Cisco Unified IP Phone 7975	X
3x MGCP Gateways	G
5x MGCP Gateways	G

Related Topics

- Chapter 7, "Understanding Call Management Records"
- Chapter 8, "Cisco Call Management Records Field Descriptions"
- Chapter 10, "Cisco Call Management Record Examples"

Related Documentation

The following documents contain additional information related to CDRs:

- *Cisco Unified Serviceability Administration Guide*
- *CDR Analysis and Reporting Administration Guide*

Cisco Call Management Record Examples

This chapter provides examples of call management records (CMRs). The chapter contains the following information:

- CMR Examples, page 10-1
- Related Topics, page 10-3
- Related Documentation, page 10-3

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. To see a sample of the CDR that gets generated during a normal call, see the "Normal Calls (Cisco Unified IP Phone to Cisco Unified IP Phone)" section on page 4-49.

CMR 1

Field Names	AAC CDR
cdrRecordType	2
globalCallID_callManagerid	1
globalCallID_callId	96004
nodeId	1
callIdentifier	28141535
directoryNumber	1010
dateTimeStamp	1202412060
numberPacketsSent	358
numberOctetsSent	61576
numberPacketsReceived	351
numberOctetsReceived	60372
numberPacketsLost	1

jitter	0
latency	0
pkid	e95df5b1-2914-4a03-befb-0f58bf16392d
directoryNumberPartition	
deviceName	SEP003094C39BE7
globalCallId_ClusterId	StandAloneCluster
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

CMR 2

Field Names	AAC CDR
cdrRecordType	2
globalCallID_callManagerid	1
globalCallID_callId	96004
nodeId	1
callIdentifier	28141536
directoryNumber	1004
dateTimeStamp	1202412060
numberPacketsSent	352
numberOctetsSent	60544
numberPacketsReceived	356
numberOctetsReceived	61232
numberPacketsLost	1
jitter	0
latency	0
pkid	545ff25a-5475-4882-af09-c7b714802703
directoryNumberPartition	
deviceName	SEP007EBBA6376
globalCallId_ClusterId	StandAloneCluster
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

Related Topics

- Chapter 4, "CDR Examples"
- Chapter 7, "Understanding Call Management Records"
- Chapter 8, "Cisco Call Management Records Field Descriptions"
- Chapter 9, "Cisco Call Management Records K-Factor Data"

Related Documentation

The following documents contain additional information related to CMRs:

- *Cisco Unified Serviceability Administration Guide*
- *CDR Analysis and Reporting Administration Guide*

INDEX

O

OnBehalfOf codes, table   **6-7**

on-net calls   **4-63**

on-net calls, CDR example   **4-63**

origCalledPartyRedirectOnBehalfOf, CDR field name   **5-18**

origCalledPartyRedirectReason, CDR field name   **5-18**

origCallTerminationOnBehalfOf, CDR field name   **5-17**

origCause_location, CDR field name   **5-4**

origCause_value, CDR field name   **5-5**

origConversationID, CDR field name   **5-21**

origDeviceName, CDR field name   **5-17**

origDTMFMethod, CDR field name   **5-20**

originalCalledPartyNumber, CDR field name   **5-9**

originalCalledPartyNumberPartition, CDR field name   **5-15**

original calling party on transfer   **4-33, 4-50**

original calling party on transfer, CDR example   **4-33, 4-50**

origIpAddr, CDR field name   **5-3**

origLegCallIdentifier, CDR field name   **5-2**

origMediaCap_Bandwidth, CDR field name   **5-21**

origMediaCap_g72.3BitRate, CDR field name   **5-6**

origMediaCap_maxFramesPerPacket, CDR field name   **5-6**

origMediaCap_payloadCapability, CDR field name   **5-6**

origMediaTransportAddress_IP, CDR field name   **5-5**

origMediaTransportAddress_Port, CDR field name   **5-6**

origNodeId, CDR field name   **5-2**

origPrecedenceLevel, CDR field name   **5-5**

origRSVPAudioStat, CDR field name   **5-7**

origRSVPVideoStat, CDR field name   **5-8**

origSpan, CDR field name   **5-3**

origVideoCap_Bandwidth, CDR field name   **5-6**

origVideoCap_Codec, CDR field name   **5-6**

origVideoCap_Resolution, CDR field name   **5-6**

origVideoTransportAddress_IP, CDR field name   **5-7**

origVideoTransportAddress_Port, CDR field name   **5-7**

outpulsedCalledPartyNumber, CDR field name   **5-21**

outpulsedCallingPartyNumber, CDR field name   **5-21**

P

partitions and numbers   **3-3**

personal assistant calls   **4-51**

personal assistant conferencing   **4-56**

personal assistant conferencing, CDR example   **4-56**

personal assistant direct call   **4-51**

personal assistant direct call, CDR example   **4-51**

personal assistant interceptor going directly to destination, CDR example   **4-52**

personal assistant interceptor going to media port and transferring the call   **4-52**

personal assistant interceptor going to media port and transferring the call, CDR example   **4-52**

personal assistant interceptor going to multiple destinations   **4-53**

personal assistant interceptor going to multiple destinations, CDR example   **4-53**

personal media interceptor going directly to destination   **4-52**

pickup   **4-21**

pickup, CDR example   **4-21**

pkid, CDR field name   **5-15**

pkid, CMR field name   **8-5**

precedence calls (MLPP)   **4-57**

precedence calls (MLPP), CDR example   **4-57**

R

redirected calls   **4-35**

redirection (3xx), CDR example   **4-58**

Redirection (3xx) calls   **4-58**

redirect reason codes, table   **6-6**

refer calls   **4-59**

removing a controller from a linked conference, CDR example   **4-12**

removing a party (controller) from a linked conference   **4-11**

removing a party from a linked conference   **4-9**

removing a party from a linked conference, CDR example   **4-10**

Cisco Unified Communications Manager Call Detail Records Administration Guide