

Cisco Unified CallManager Call Detail Record Definitions, Release 5.1(3)

This document describes the format and logic of the call detail records (CDRs) and call management records (CMRs) that the Cisco Unified CallManager Release 5.0 (and later) system generates. You can use this information for post-processing activities such as generating billing records and network analysis. This document describes how to access the CDR/CMR files and how to interpret fields in the files.

When you install your system, the system enables CDRs by default. 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 CallManager for the change to take effect. The system responds to all changes within a few seconds. The system enables CMRs (diagnostic data) separately from CDR data.

Contents

This document covers the following topics:

- New and Changed Information, page 2
- Configuring Cisco Unified CallManager CDR, page 2
- CDR Processing, page 4
- Cisco Unified CallManager CDR Overview, page 5
- Call Types, page 11
- Interpreting Cisco Personal Assistant Data in the CDRs, page 27
- Call Scenarios, page 33
- CDR Field Descriptions, page 63
- CMR Field Descriptions (Diagnostic), page 80
- K-Factor Data in CMRs, page 85
- Codec Types, page 86
- Call Termination Cause Codes, page 87
- Related Documentation, page 93
- Obtaining Documentation, Obtaining Support, and Security Guidelines, page 93

New and Changed Information

The Release Notes for the corresponding release of Cisco Unified CallManager describe new features or changes for CDRs/CMRs that are pertinent to a specified release.

Configuring Cisco Unified CallManager CDR

CDR Analysis and Reporting (CAR) comprises a group of complementary services, which you can activate in the Service Activation window in Cisco Unified CallManager Serviceability. Before you can launch CAR from the Tools menu in Cisco Unified CallManager Serviceability, you must activate the CAR services by using the following procedure.

Procedure

Step 1 Choose Tools > Service Activation.

The Service Activation window displays.

Step 2 From the Servers drop-down list box, choose the first node of the cluster.

The window displays the service names for the server that you chose, the service type, and the activation status of the services.



Note Activate the CAR services on only the first node, where the Cisco Unified CallManager database resides.

- Step 3 Check the check boxes next to the following CDR services:
 - Cisco CAR Web Service
 - Cisco SOAP-CDROnDemand (optional). If you are using a third-party billing application that accesses CDR data via an HTTPS/SOAP interface, activate this service.

P

- Tip Unchecking the check boxes next to the CDR services and clicking Update deactivates the services. If you deactivate the Cisco CAR Web Service, the system removes CAR from the Tools menu on the Cisco Unified CallManager Serviceability menu.
- Step 4 After you have finished making the appropriate changes, click Update.

You must also configure certain CDR service and enterprise parameters:

- Configuring CDR Service Parameters, page 3
- Configuring CDR Enterprise Parameters, page 4

Additional Information

See the "Getting Started With CDR Reporting And Analysis" chapter in the *Cisco Unified CallManager* CDR Reporting and Analysis Reporting Tool Administration Guide for additional information.

Configuring CDR Service Parameters

CAR relies on the data in the CDR and CMR records to generate both CAR and CDR reports. CAR requires that the CDR records be available in flat files on the CDR Repository node (the first node). Even if you do not use the CAR and CDR reporting services, you must enable certain Cisco Unified CallManager service parameters to ensure that CDR records are generated, and are generated in the manner that you can use for your particular system.

You can configure these parameters in the Service Parameters Configuration window in Cisco Unified CallManager Administration. To access the Service Parameters Configuration window, open Cisco Unified CallManager Administration and choose **System** > **Service Parameters**. Choose the **Advanced** button to display the complete list of Service Parameters. The following list of service parameters can affect CDR/CMR records:

- System Parameters
 - **CDR Enabled Flag**—This parameter determines whether CDRs are generated. Valid values specify True (CDRs are generated) or False (CDRs are not generated). For this required field, the default value specifies False. Enable this parameter on all servers in the cluster
 - CDR Log Calls With Zero Duration Flag—This parameter enables or disables the logging of CDRs for calls that were never connected or that lasted less than 1 second. Cisco Unified CallManager logs unsuccessful calls (calls that result in reorder, such as might occur because of a forwarding directive failure or calls that attempt to go through a busy trunk) regardless of this flag setting. This parameter represents a required field. The default value specifies False.
- Clusterwide Parameters (Device General)
 - Call Diagnostics Enabled—This parameter determines whether the system generates call management records (CMRs), also called 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.
 - **Display FAC in CDR**—This parameter determines whether the Forced Authorization Code (FAC) that is associated with the call displays in the CDR. Valid values specify True (display authorization code in CDRs) or False (do not display authorization code in CDRs) for this required field. The default value specifies False.
 - Show Line Group Member DN in finalCalledPartyNumber CDR Fields—This parameter determines whether the finalCalledPartyNumber field in CDRs shows the directory number (DN) of the line group member who answered the call or the hunt pilot DN. Valid values specify True (the finalCalledPartyNumber in CDRs will show the DN of the phone that answered the call) or False (the finalCalledPartyNumber in CDRs will show the hunt pilot DN). This parameter applies only to basic calls that are routed through a hunt list without feature interaction such as transfer, conference, call park, and so on. If a feature is involved in the call, the hunt pilot DN will show in the finalCalledPartyNumber field regardless of the setting in this parameter. This parameter does not apply to Cisco Unified CallManager Attendant Console. The default value for this required field specifies False.
- Clusterwide Parameters (Device Phone)
 - Add Incoming Number Prefix to CDR This parameter determines whether Cisco Unified CallManager adds the incoming prefix (as specified in the National Number Prefix, International Number Prefix, Subscriber Number Prefix, and Unknown Number Prefix service parameters) to the

calling party number in the CDRs for that call. If the destination of the call is a gateway, Cisco Unified CallManager will not add the prefix to the CDRs even if this parameter is enabled. The default value for this required field specifies False.

Configuring CDR Enterprise Parameters

Configure these CDR parameters on the Enterprise Parameters Configuration window in the Cisco Unified CallManager Administration. To access Enterprise Parameters Configuration windows, open Cisco Unified CallManager Administration and choose **System -> Enterprise Parameters**.

- CDR Parameters
 - CDR File Time Interval—This parameter specifies the time interval for collecting CDR data. For example, if this value is set to 1, each file will contain 1 minute of CDR data (CDRs and CMRs, if enabled). The external billing server and CAR database will not receive the data in each file until the interval expires (or sometime later, depending on the CAR Loader schedule setting). Consider how quickly you want access to the CDR data when you decide what interval to set for this parameter. Setting this parameter to 60 means that each file will contain 60 minutes worth of data, but that data will not be available until the 60-minute period has elapsed, and the records are written to the CAR database. The system sends CDR files to the configured billing server(s). The default value specifies 1. The minimum value specifies 1, and the maximum value specifies 1440. The unit of measure for this required field represents a minute.
 - Cluster ID—This parameter provides a unique identifier for the cluster. Because the parameter
 gets used in CDRs, collections of CDRs from multiple clusters can be traced to the sources. The
 default value specifies StandAloneCluster. The maximum length comprises 50 characters and
 provides a valid cluster ID that comprises any of the following characters: A-Z, a-z, 0-9, . -.
- CCM Web Services Parameters
 - Allowed CDRonDemand get_file Queries Per Minute—This parameter specifies the maximum number of CDRonDemand get_file queries that are allowed per minute for the system. For this required field, the default value specifies 10. The minimum value equals 1, and the maximum value equals 20.
 - Allowed CDRonDemand get_file_list Queries Per Minute—This parameter specifies the maximum number of CDRonDemand get_file_list queries that are allowed per minute for the system. For this required field, the default value specifies 20. The minimum value equals 1, and the maximum value equals 40.

CDR Processing

Cisco Unified CallManager 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 CallManager 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 CallManager, 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 CallManager 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 server where the Cisco Unified CallManager database exists
- nodeId—Identifies the node
- datetime—UTC time in yyyymmddhhmm format
- seqnumber—Sequence number

Two examples of filenames follow:

- cdr_Cluster1_01_200404021658_1
- cmr_Cluster1_02_200404061011_6125

Flat File Format

The CDR and CMR flat files have the following format:

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

The following example shows a flat file:

```
Line1-"cdrRecordType","globalCallID_callManagerId","globalCallID_callId","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.

Cisco Unified CallManager CDR Overview

The following sections provide a brief description of how CDRs are generated and managed in Cisco Unified CallManager:

• CDR Management, page 6

Types of Call Information Records, page 8

CDR Management

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

- Collects the CDR/CMR files from individual nodes within a cluster to the CDR Repository node.
- Maintains the CDR/CMR files on the CDR Repository node.
- 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 node.
- Sends CDR/CMR files from the CDR Repository node to up to three customer billing servers.
- Monitors disk usage of CDR/CMR files on the CDR Repository node.
- Periodically deletes CDR/CMR files that have been successfully delivered. You can configure the amount of storage that is used to store flat files. 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.

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

CDR Agent

As part of the CDRM feature, a resident component on every node within a Cisco Unified CallManager cluster acts as the CDR Agent. On a node where both Cisco Unified CallManager and the CDR Agent are running, Cisco Unified CallManager writes the CDRs into CDR flat files (CSV format) with a special control character ("_") that is prefixed to the filename by the call-processing module and indicates that the file is not available for transfer. If this control character is not present, the system assumes the file to be available for transfer and sends the file 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 CallManager nodes within a cluster continuously write CDRs to flat files, close existing flat files, and open new ones. The number of records that are written varies by the type of call and 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.

The CDR Agent periodically polls the files in a designated path (/var/log/active/cm/cdr, which is a softlink to /common/log/cdr), every 6 seconds to determine whether a CDR file is available for transfer to the CDR Repository node. Having a short interval provides an advantage because as soon as a file is available, the system can deliver it immediately to the CDR Repository node.

The CDR Agent uses a standard SFTP utility, sftp_connect.sh, to transfer CDR files from the Cisco Unified CallManager nodes to the CDR Repository node. The utility requires a batch file as input and generates a log file that indicates the results of the requested actions. The CDR Agent creates unique batch and log files for each transfer session.

In case of an SFTP failure, the component on Cisco Unified CallManager repeatedly tries to make new connections until successful. When CDR files are accumulated due to a lack of an SFTP connection, the system sends all leftover CDR files to the CDR Repository node immediately after connectivity is restored.

When the CDR Agent starts or restarts, it checks whether any CDR files remain from the previous life cycle and sends them over to the CDR Repository node.

Should SFTP fail to transfer CDR files to the CDR Repository node, the system raises an alarm.

CDR Repository Manager

Within a Cisco Unified CallManager cluster, one instance of the CDR Repository Manager runs on the CDR Repository node. It manages CDR files that are received from the Cisco Unified CallManager nodes and periodically sends the files to the specified customer/third-party billing servers via an (s)FTP connection.

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

If any external billing server is specified in CDRM configuration, the system creates a soft link to the file that is created in a directory that is designated to the destination. The file sender component of the CDR Repository Manager detects this soft link and sends the file to the destination with the specified method, either SFTP or FTP. If the delivery is successful, the system removes the soft link in the destination directory.

Every Cisco Unified CallManager node 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 on the CDR Repository node 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 how to configure the CDR Repository Manager and customer billing servers, see the "CDR Repository Manager Configuration" chapter in the *Cisco Unified Serviceability Administration Guide*.

CDR onDemand Service

The CDR onDemand Service, a SOAP/HTTPS-based service, runs on the CDR Repository 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 (s)FTP. The system can activate the CDR onDemand service on the CDR Repository node as 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 CallManager Developers Guide for Release 5.1(3)*.

Types of Call Information Records

Cisco Unified CallManager generates two different types of call information records: Call Detail Records (CDRs) and Call Management Records (CMRs). 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 and/or video of the call. More than one CMR can exist per CDR.

The CDRs relate to the CMRs via the two globalCallID columns:

- globalCallID_callManagerId
- globalCallId_callId

When the Call Diagnostics Enabled 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 CallManager does not perform any post processing on CDRs or CMRs.

This section contains the following topics:

- Global Call Identifier, page 8
- Number Translations, page 9
- Partitions and Numbers, page 9
- Timestamps, page 10
- Call Termination Cause Codes, page 11

Global Call Identifier

Cisco Unified CallManager allocates a global call identifier (GlobalCallID) each time that a Cisco Unified IP Phone is taken off hook or a call is received from a gateway.

The CDR table (Table 1) lists CDRs that are written at the end of a call in the order that they are written. GlobalCallIDs for active calls do not appear in the CDR table. Other global IDs also may not appear in the CDR table. For example, each call leg in a conference call gets assigned a GlobalCallID that the conference GlobalCallID overwrites. The original GlobalCallID does not appear in the CDR.

GlobalCalIID	Start Time	End Time
1	973795815	973795820
2	973795840	973795845
5	973795860	973795870
4	973795850	973795880

Table 1 Sample CDR Table

The CDR table does not contain an entry for GlobalCalIID 3 because that call was active when this record was taken. The table shows GlobalCalIID 5 listed before GlobalCalIII 4 because the GlobalCalIID 5 call ended before the GlobalCalIID 4 call ended.

Number Translations

The Cisco Unified CallManager 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 partition 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#."

In this release, 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 2:

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 CallManager 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.

Table 2 Partition/Extension Numbers in CDRs

Phone Number	Description				
originalCalledPartyNumberPartition	This number identifies the partition name that is associated with the OriginalCalledPartyNumber field. This field uniquely identifies this number because the Cisco Unified CallManager supports multiple Cisco Unified IP Phones with the same extension number in different partitions.				
	For calls that egress through a gateway, this field specifies the partition name that is associated with the route pattern that pointed to the gateway.				
finalCalledPartyNumberPartition	This number identifies the partition name that is associated with the FinalCalledPartyNumber field. This field uniquely identifies this number because the Cisco Unified CallManager 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 CallManager 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 2 Partition/Extension Numbers in CDRs (continued)

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:

Table 3	UTC Timestamps in CDRs	
---------	------------------------	--

Field	Description
dateTimeOrigination	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	This field designates the time that the devices connect and speech begins. This field shows a zero if the call never connects.
dateTimeDisconnect	This field designates the time that the call disconnects. This field shows a zero if the call never connects.

Call Termination Cause Codes

The CDR includes two call termination 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 termination cause code value shows zero.

The "Call Termination Cause Codes" section on page 87 lists the call termination cause code values per ITU specification Q.850. For On Net call legs, the Cisco Unified CallManager determines the call termination cause code value. For Off Net call legs, the far-end switch determines the call termination 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 1Convert the database display (-1139627840) to a hex value.
The hex value equals 0xBC12A8C0.
- Step 2Reverse 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.

Call Types

A successful call between two parties logs one CDR. Each CDR contains all fields, but some fields may not get used. If a field is not used, see the default values in the CDR definitions table. When supplementary services are involved in a call, additional CDRs may be written.

In addition to the CDR, a call may involve one CMR per endpoint. In a successful call between two parties who are each using an IP phone, the system writes two CMRs: one for the originator and one for the destination of the call.

This section describes the records that are written for different call types in the system.

- Successful On-Net Calls, page 12
- Abandoned Calls, page 12
- Calls with Busy or Bad Destinations, page 13
- Short Calls, page 13
- Forwarded or Redirected Calls, page 14
- Pickup Calls, page 14
- Transferred Calls, page 16
- Conference Calls, page 18
- Meet-Me Conferences, page 20
- Ad Hoc Conference Linking, page 21
- Precedence Calls (MLPP), page 23
- Malicious Calls, page 24
- Conference Drop Any Party, page 24
- Immediate Divert (to Voice Messaging System), page 25
- Video Calls, page 25
- Original Calling Party on Transfer, page 26

Successful On-Net Calls

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

The following table contains two examples:

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

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

Abandoned Calls

The logging of calls with zero duration represents an optional action. If the **CDR Log Calls With Zero Duration Flag** service parameter 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.

Abandoned Calls CDR Examples

The following table contains two examples:

- A—Extension 2001 goes off hook then on hook (when the CdrLogCallsWithZeroDurationFlag is set to **True**).
- B—Extension 2001 calls 2309, but 2001 hangs up (abandons) the call before it is answered.

	Calling Party	Calling Partition	Original Called Party	Original Called Partition	Orig Cause	Dest Cause	Duration
А	2001	Accounts			16	0	0
В	2001	Accounts	2309		16	0	0

Calls with Busy or Bad Destinations

The system logs these calls as a normal call with all relevant fields containing data. The Calling or Called Party Cause fields contain a cause code that indicates why the call was not connected, 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 (the CDR Log Calls With Zero Duration Flag set at True or False, a duration of zero, and a DateTimeConnect value of zero).

Calls with Busy or Bad Destinations CDR Examples

The following table contains three examples:

- A—Call to PSTN number, party is engaged (cause 17 = user busy).
- B—Call to PSTN number, number does not exist (cause 1 = number unavailable).
- C—Call to PSTN, fails because PSTN trunks are out of order (cause 38 = Network Out Of Order).

	Calling Party	Calling Partition	Original Called Party	Original Called Partition	Orig Cause	Dest Cause	Duration
А	2001	Accounts	902920262226	PSTN	0	17	0
В	2001	Accounts	902920100000	PSTN	0	1	0
С	2001	Accounts	902920262226	PSTN	0	38	0

Short Calls

A short call, with a CDR Log Calls With Zero Duration Flag 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 Call 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	Calling	Original	Original	Orig	Dest	DateTime	Duration
Party	Partition	Called Party	Called Partition	Cause	Cause	Connect	
2001	Accounts	2309	Marketing	0	16	973795815	0

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 is forwarded more than twice, the intermediate forwarding parties do not populate in the CDR.

Call forwarding can occur on several conditions (always, on busy, and on no answer). The condition under which the call is 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 is forwarded, the lastRedirectDn and lastRedirectDnPartition fields contain the directory number and partition of the last phone that forwarded or redirected the call.

Forward or Redirected Call CDR Examples

The following table contains two examples:

- A—Call from the PSTN to extension 2001, forwarded to 2309, where the call is answered
- B—Call from the PSTN to extension 2001, forwarded to 2309, which forwards to voice messaging system

	Calling Party	Original Called Party	Original Called Partition	Final Called Party	Final Called Partition	Last Redirect Party	Last Redirect Partition	Duration	OriginalCalled Party Redirect OnBehalfOf	Last Redirect Redirect OnBehalfOf
A	02920262227	2001	ACNTS	2309	MKTG	2001	ACNTS	120	5	5
В	02920262227	2001	ACNTS	6000	VMAIL	2309	MKTG	60	5	5

Pickup Calls

Cisco Unified CallManager includes two pickup modes: Pickup and Auto Pickup. The following sections describes these calls:

- Pickup Calls, page 14
- Auto Pickup, page 15

Pickup

Pickup calls work like forwarded calls. The CDRs for pickup 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 is picked up, the finalCalledPartyNumber and finalCalledpartyNumberPartition fields will differ and contain the Directory Number and partition of the phone that picked up the call. Also, when a call is picked up, the lastRedirectDn and lastRedirectDnPartition fields contain the directory number and partition of the last phone that redirected this call.

The origTermination, destTermination, lastRedirect, and Join OnBehalfOf fields contain 16 (Pickup) and the redirect reason field contains 5 (Pickup).

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

Pickup Call CDR Example

- 1. A call comes in from the PSTN to extensions 2000, 2001, and 2002, which are in the same pickup group.
- 2. Extension 2002 picks up the call that is ringing on 2001.
- 3. Extension 2002 answers the call, and the call connects between the PSTN caller and extension 2002.

							Orig	Dest	Last		
				Original	Final	Last	Termina-	Termina-	Redirect	Last	
Call	Orig		Dest	Called	Called	Redirect	tion On	tion On	On	Redirect	Join On
ID	Cause	Calling Party	Cause	Party	Party	Party	BehalfOf	BehalfOf	BehalfOf	Reason	BehalfOf
22	0	9728131234	16	2001	2002	2001	16	16	16	5	16

Auto Pickup

Auto Pickup works like call pickup with auto answer. The call connects automatically, so no need exists for the last answer softkey press. The system generates two CDRs for Auto Pickup, and these CDRs have the same Call ID.

The system generates the first CDR for the original call. This CDR will have the origTerminationOnBehalfOf and destTerminationOnBehalfOf fields equal to 16 (Pickup), which indicates that the call 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), which indicates that the system joined the call 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 Example

- 1. A call comes in from the PSTN to extension 2001; 2002 and 2002 are in the same pickup group.
- 2. Extension 2002 picks up the call that is ringing on 2001.
- 3. The call automatically connects between the PSTN caller and extension 2002.

Call ID	Orig Cause	Calling Party	Dest Cause	Original Called Party	Final Called Party	Last Redirect Party	Orig Termina- tion On BehalfOf	Dest Termina- tion On BehalfOf	Last Redirect On BehalfOf	Last Redirect Reason	Join On BehalfOf
11	126	9728131234	126	2001	2001	2001	16	16	0	0	0
11	0	9728131234	16	2002	2002	2001	16	16	16	5	16

Transferred Calls

A single CDR cannot show all the data necessary for a call transfer because it is too complex. Each time a call is transferred, the Cisco Unified CallManager terminates the CDR for that call and initiates a new CDR.

Calls that are transferred have multiple CDRs logged for them, as follows:

- 1. Original call from party A to party B.
- 2. Call from the transferring party (party A or B) to the transfer destination (party C).
- 3. Call from the transferred party (party A or B) to the destination (party C).

The first CDR represents the original placed call. The second CDR represents the setup call (consultative/announcement) that is used to initiate the transfer. The third CDR represents the transferred call itself. The first two CDRs have the origCause_value and destCause_value set to Split (126).

They also have the origCallTerminationOnBehalfOf and destCallTerminationOnBehalfOf fields set to Transfer (10) to indicate that these calls were involved in a transfer. The transferred leg of the call has the joinOnBehalfOf field set to Transfer (10) to indicate that this call resulted from a transfer; therefore, all legs of the transfer can be tied back to a single call.

Transferred Calls CDR Examples

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

Example 1

A calls B; A transfers B to C. The three logged calls follow:

- 1. Call from A to B
- 2. Call from A to C
- **3**. Call from B to C

If the call was a blind transfer, the call from A to C will have a duration of zero seconds. If the call was a consultation transfer, all calls will have non-zero durations. Original Called Party and Call Party Number fields match.

Example 2

A calls B; B transfers A to C. The three logged calls follow:

- 1. Call from A to B
- 2. Call from B to C
- 3. Call from A to C

If the call was a blind transfer, then the call from B to C will have a duration of zero seconds. If the call was a consultation transfer, then all calls will have non-zero durations. Original Called Party and Call Party Number fields match.

Example 3

A calls B; B transfers A to C on a blind transfer. C is Call Forwarded on No Answer to D. The calls that are logged follow:

- 1. Call from A to B
- **2**. Call from B to C
- 3. Call from A to D

Because the call was a blind transfer, the call from B to C has a duration of zero seconds. The call from A to D will have the Original Called Party field set to "C", and the Called Party Number field set to "D".

Transfer Without Consultation

The process of transferring a call, without consultation, involves the creation of three CDRs. The first CDR reflects the call between the original two parties (A and B), the second CDR represents the (zero length) call between the transferring party (A) and the new party (C), and the final CDR reflects the call between B and C.

No CDR reflects the time that a call is on hold. If a call is through a PSTN gateway, the call accrues charges that are not reflected in the CDRs while the call is on hold.

Transfer Without Consultation CDR Examples

The following table contains three examples:

- A—Call from extension 2001 to a PSTN number, talking for 120 seconds.
- B—Extension 2001 initiates a transfer without consultation (duration is zero) to extension 2002.
- C—Extension 2001 completes the transfer, dropping out of the call, and leaving a call between the other two parties.

	Calling Party	Calling Partition	Calling Leg	Original Called Party	Original Called Partition	Called Leg	Orig Cause	Dest Cause	OrigCall Term On BehalfOf	DestCall Term On BehalfOf	Join On BehalfOf	Duration
А	2001	ACNTS	101	3071111	PSTN	102	126	126	10	10	0	120
В	2001	ACNTS	103	2002	ACNTS	104	126	126	10	10	0	0
С	3071111	PSTN	102	2002	ACNTS	104	0	16	0	0	10	350

Transfer with Consultation

Transfer with consultation essentially acts identical to transfer without consultation, except the duration of the middle call is not zero.

As with a transfer without consultation, Cisco Unified CallManager creates three CDRs. The first CDR reflects the call between the original two parties (A and B), the second CDR represents the consultation call between the transferring party (A) and the new party (C), and the final CDR reflects the call between B and C.

Transfer with Consultation CDR Examples

The following table contains three examples:

- A—Call from extension 2001 to a PSTN number, talking for 120 seconds.
- B—Extension 2001 places the PSTN call on hold and calls extension 2002, talking for 30 seconds.

	Calling Party	Calling Partition	Calling Leg	Original Called Party	Original Called Partition	Called Leg	Orig Cause	Dest Cause	OrigCall Term On BehalfOf	DestCall Term On BehalfOf	Join On BehalfOf	Duration
А	2001	ACNTS	101	3071111	PSTN	102	126	126	10	10	0	120
В	2001	ACNTS	103	2002	ACNTS	104	126	126	10	10	0	30
С	3071111	PSTN	102	2002	ACNTS	104	0	16	0	0	10	350

• C—Extension 2001 completes the transfer, dropping out of the call, leaving a call between the other two parties.

Conference Calls

Three major operational factors exist for conference call CDRs:

1. When the 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 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 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 are 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 CallManager. 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"
- origCalledPtyRedirectOnBehalfOf—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 were 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 Calls CDR Examples

The following tables contain these examples:

- Call from 2001 to 2309.
- After 60 seconds, user 2001 presses the "conference" key on the Cisco Unified IP Phone and dials the PSTN number "3071111."
- 3071111 answers and talks for 20 seconds; 2001 then presses the conference key to complete the conference.
- The conference talks for 360 seconds.
- Each call leg shows as a call into the conference bridge. The call appears as a call *into* the bridge, regardless of the actual direction of the call.
- 3071111 hangs up and leaves 2001 and 2309 in the conference. Because only two participants remain in the conference, the conference features directly join the two, and they talk for another 55 seconds.

Calling Party	Calling Partition	Calling Leg	Original Called Party	Original Called Partition	Called Leg	Final Called Party	Final Called Partition	Last Redirect Party	Last Redirect Reason	Orig Conver sation Id
2001	ACNTS	101	2309	MKTG	102	2309	MKTG	2001	0	0
2001	ACNTS	101	2309	MKTG	115	b0029901001		b0029901001	0	1
2309	ACNTS	101	b0029901001		116	b0029901001		b0029901001	0	1
3071111	PSTN	101	b0029901001		117	b0029901001		b0029901001	0	1
2001	ACNTS	105	3071111	PSTN	106	3071111	PSTN	3071111	0	0
2001	ACNTS	101	2309	MKTG	102	2309	MKTG	b0029901001	98	0v

OrigCall Termination OnBehalfOf	DestCall Termination OnBehalfOf	Original CalledParty Redirect OnBehalfOf	Last Redirect OnBehalfOf	Join OnBehalfOf	Duration	Comment
4	4	0	0	0	60	
12	0	4	4	4	360	ConfControllerDn=2001;ConfController DeviceName=SEP0003E333FEBD
12	0	4	4	4	360	ConfControllerDn=2001;ConfController DeviceName=SEP0003E333FEBD
4	4	4	4	4	360	ConfControllerDn=2001;ConfController DeviceName=SEP0003E333FEBD
4	4	0	0	0	20	
12	42	0	4	4	55	ConfControllerDn=2001;ConfController DeviceName=SEP0003E333FEBD

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 Examples

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 CallManager, and calls to the conference bridge appear as forwarded calls; that is, User A phones the predetermined number (5001), and the call gets forwarded to a conference bridge port. The conference bridge port appears with a special number of the form "b0019901001."

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

	Calling Party	Calling Partition	Original Called Party	Original Called Partition	Final Called Party	Final Called Partition	Last Redirect Party	Last Redirect Partition	Duration
А	2001	Accounts	5001		b0019901001		b0019901001		70
В	2002	Accounts	5001		b0019901001		b0019901001		65
С	2003	Accounts	5001		b0019901001		b0019901001		80

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.

Two types of conference linking exist:

- Linear-No more than two ad hoc conferences can link directly to any participating conference.
- Nonlinear—Three or more ad hoc conferences that link directly to another conference. The system does not permit this type of linking by default because potentially negative impact on conference resources exists.

Linear Ad Hoc Conference Linking Using Join CDR Example

The following table contains example CDRs for this scenario:

- Alice (1000) calls Bob (1001). This represents an original call.
- Bob (1001) conferences in Carol (1002) This represents a consultation call.
- Dave (1003) calls Carol (1002). This represents an original call.
- Dave (1003) conferences in Ed (1004) This represents a consultation call.
- The system creates two separate conferences. Carol takes part in both conferences. At this point, the system generates CDR1, CDR2, CDR3, and CDR4.
- Carol (1002) joins the two conferences through a conference bridge (b002990122). At this point, the system generates CDR5..
- Dave (1003) joins the two conferences through a conference bridge (b002990122). At this point, the system generates CDR6.
- Ed (1004) leaves the conference. The system generates CDR7.
- Dave (b002990122) leaves the conference. The system generates CDR8.
- Alice (1000) leaves the conference. The system generates CDR9.
- Bob (1001) leaves the conference. The system generates CDR10.

Carol (1002) leaves the conference. The system generates CDR11.

Calling Party Number	globalCalIID- callid	Original Leg Call Identifier	Dest Leg Call Identifier	Original Called Party Number	Final Called Party Number	Last RedirectDn	OrigCall Termination OnBehalfOf
1000 (CDR1)	1	11	12	1001	1001	1001	4
1001 (CDR2)	2	13	14	1002	1002	1002	4
1003 (CDR3)	3	21	22	1002	1002	1002	4

Calling Party Number	globalCalIID- callid	Original Leg Call Identifier	Dest Leg Call Identifier	Original Called Party Number	Final Called Party Number	Last RedirectDn	OrigCall Termination OnBehalfOf
1003 (CDR4)	4	23	24	1004	1004	1004	4
1002 (CDR5)	3	22	25	b0029901222	b0029901222	1003	4
1003 (CDR6)	3	21	26	b0029901222	b0029901222	1003	0
1004 (CDR7)	3	24	27	b0029901222	b0029901222	1003	0
b002990 1222 (CDR8)	1	25	28	b0029901001	b0029901001	10020	0
1000 (CDR9)	1	11	15	b0029901001	b0029901001	1001	0
1001 (CDR10)	1	12	16	b0029901001	b0029901001	1001	0
1002 (CDR11)	1	14	17	b0029901001	b0029901001	1001	0

This is a continuation of the previous table.

Calling Party Number	DestCall Ter- mination OnBehalfOf	LastRedirect Redirect Rea- son	LastRedi- rect Redirect OnBehalfOf	Original Con- versationID	Destination Conversation ID	Comment
1000 (CDR1)	4	0	0	0	0	
1001 (CDR2)	4	0	0	0	0	
1003 (CDR3)	4	0	0	0	0	
1003 (CDR4)	4	0	0	0	0	
1002 (CDR5)	4	98	4	0	2222	ConfControllerDn=1003;ConfCon- trollerDevice- Name=SEP0003E333FAD1;ConfReque storDn-1003;ConfRequestorDevi- ceName=SEP0003E333 FAD1
1003 (CDR6)	0	98	4	0	2222	ConfControllerDn=1003;ConfCon- trollerDevice- Name=SEP0003E333FAD1;ConfReque storDn-1003;ConfRequestorDevi- ceName=SEP0003E333 FAD1
1004 (CDR7)	0	98	4	0	2222	ConfControllerDn=1003;Conf- ControllerDevice- Name=SEP0003E333FAD1;Conf RequestorDn-1003;ConfRequest- orDeviceName=SEP0003E333 FAD1
B002990 1222 (CDR8)	0	98	4	2222	1111	ConfControllerDn=1003;Conf- ControllerDevice- Name=SEP0003E333FAD1;Conf RequestorDn-1003;ConfRequest- orDeviceName=SEP0003E333 FAD1
1000 (CDR9)	0	98	4			
1001 (CDR10)	0	98	4			
1002 (CDR11)	0	98	4			

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 Calls CDR Example

The following table contains an example CDR for this scenario:

- User A (2001) calls another IP phone by dialing a precedence pattern (precedence level 2).
- User A (2001) calls another IP phone by dialing a precedence pattern (precedence level 3).
- User A receives a higher-level precedence call from another network (precedence level 1).
- The higher precedence level call preempts the first call.

Calling Party	Calling Partition	Origin Precedence Level	Original Called Party	Original Called Partition	Dest Precedence Level	Orig Cause	Dest Cause
2001	CMD	2	826001	FIRE	2	0	16
2001	CMD	3	836001	FIRE	3	0	16
9728552 001	GEN	1	6001	FIRE	1	16	0
2001	CMD	2	826001	FIRE	2	0	9
9728552 001	GEN	1	826001	FIRE	1	0	16

Malicious Calls

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

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"

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 was connected to a conference and dropped by this feature.

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



This table continues the Conference Drop Any Party CDR example .

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

Immediate Divert (to Voice Messaging System)

CDRs for Immediate Divert calls take place the same as forwarded calls except values exist for origCalledPartyRedirectOnBehalfOf and the lastRedirectRedirectOnBehalfOf fields.

Immediate Divert CDR Example

The following table contains an example CDR for this scenario:

Calling Party	Calling Partitio n	Original Called Party	Original Called Partition	Final Called Party	Final Called Partition	Last Redirect Party	Last Redirect Partition	Duration	OrigCalled Party Redirected OnBehalfOf	Last Redirect Redirect OnBehalfOf
02920262227		2001	ACNTS	2309	MKTG	2001	ACNTS	120	5	5
02920262227		2001	ACNTS	6000	VMAIL	2309	MKTG	60	5	5

Video Calls

The following table contains an example CDR for a video call for this scenario:

• Calling party 51234 calls the called party 57890.

- 100 = H.261
- 187962284 = 172.19.52.11
- 288625580 = 172.19.52.17
- 320 320K
- 2 = QCIF

Video Call CDR Example

Calling Party	Calling Partition	Calling Leg	Original Called Party	Original Called Partition	Called Leg	Orig VideoCap_ Codec	Orig VideoCap_ Bandwidth	Orig VideoCap_ Resolution	OrigVideo Transport Address_IP	OrigVideo Transport Address_Port
51234	CISCO	101	57890	CISCO	102	100	320	2	187962284	49208
						Dest	Dest	Dest	DestVideo	DestVideo
						VideoCap	VideoCap	VideoCap	Transport	Transport
						Codec	Bandwidth	Resolution	Address_IP	Address_Port
						100	320	2	288625580	49254

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 CallManager. See additional information at the "Configuring CDR Service Parameters" section on page 3.

Original Calling Party on Transfer CDR Example

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

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

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



No originalCallingParty field exists in the CDR.

Interpreting Cisco Personal Assistant Data in the CDRs

The Cisco Personal Assistant application can selectively handle incoming calls and assist with outgoing calls. This section provides a brief overview of Personal Assistant and describes the Personal Assistant call types with example CDR scenarios.

Personal Assistant provides the following features:

- **Rule-Based Call Routing**—Personal Assistant can forward and screen incoming calls based on rules that users devise. Personal Assistant can handle incoming calls according to caller ID, date and time of day, or the user meeting status based on the user calendar (such as office hours, meeting schedules, vacations, holidays, and so forth). Personal Assistant can also selectively route calls to other telephone numbers.
- Thus, Personal Assistant can route an incoming call to a desk phone, to a cell phone, home phone, or other phone, based on the call routing rules that users create. An incoming call can even generate an e-mail-based page.
- **Speech-Enabled Directory Dialing**—Personal Assistant allows users to dial a phone number by speaking the name of the called person. Personal Assistant then obtains the telephone number of that person from the corporate directory or personal address book.
- Speech-Enabled Voice-Mail Browsing—Users can use voice commands to browse, listen to, and delete voice-mail messages.
- **Speech-Enabled Simple Ad Hoc Conferencing**—Users can initiate conference calls by telling Personal Assistant to set up a conference call with the desired participants.

Personal Assistant Call Types

Personal Assistant provides the following call types:

- Personal Assistant Direct Call, page 27
- Personal Assistant Interceptor Going to Media Port and Transferring the Call, page 28
- Personal Assistant Interceptor Going Directly to Destination, page 28
- Personal Assistant Interceptor Going to Multiple Destinations, page 29
- Personal Assistant Conferencing, page 32

Personal Assistant Direct Call

A Personal Assistant direct call acts similar to the Transfer without Consultation call type. See the "Transfer Without Consultation" section.

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.



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.

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

In all cases, 2101 specifies the calling number.

Personal Assistant Interceptor Going to Media Port and Transferring the Call

This scenario acts similar to Transfer without Consultation and Forwarded Calls. See the sections on "Transfer Without Consultation" and "Forwarded or Redirected Calls".

The following table contains an example CDR for this scenario:

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

Calling Party Num	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 no rules and with rules.

Personal Assistant 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:

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.

Calling Party Number	Orig LegCall 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	16777248	PAManaged	16777250	2110	PA	2105	1023970922	21XX		5

Personal Assistant Interceptor Going to Multiple Destinations

This scenario can have several different cases. In each case, User B (2105) configured a rule to reach him at extension 2110 or 2120. This rule could 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), page 30
- Personal Assistant Direct Multiple Destinations: 2110 and 2120 (Call Accepted at Second Destination), page 30
- Personal Assistant Direct Multiple Destinations: 2110 and 2120 (Call Accepted at Third Destination), page 30
- Personal Assistant Intercept Multiple Destinations: 2110 and 2120 (Call Accepted at First Destination), page 31
- Personal Assistant Intercept Multiple Destinations: 2110 and 2120 (Call Accepted at Second Destination), page 31
- Personal Assistant Intercept Multiple Destinations: 2110 and 2120 (Call Accepted at Third Destination), page 32

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

- User A calls Personal Assistant and says, "call User B."
- User B answers the call at 2110 extension.

Calling Party Num	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.

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.



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

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
2002	16777287	Phones	16777288	2105	PAManaged	2105	1023971619	2105	PAManaged	0
2101	16777277	PAManaged	16777288	2105	PAManaged	2105	1023971627	2105	PAManaged	7
2105	16777289	PAManaged	0				1023971629			0

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

- User A calls Personal Assistant and says, "call User B."
- User B answers the call at extension 2110.

Calling Party Num	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.

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.



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

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.

Personal Assistant Conferencing CDR Example

The following table contains an example CDR for this scenario:

- User A calls Personal Assistant route point (2000) and says, "conference User B (2105) and User C (2110)."
- Personal Assistant conferences User B and C into User A conference.

Calling Party Num	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	

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

Call Scenarios

Each normal call between two parties logs one CDR. Each CDR contains all fields that are identified in the preceding table, but some fields may not be used. If a field is not used, it stays blank if it is an ASCII string field or shows "0" if it is a numeric field. When supplementary services are involved in a call, more CDRs may get written.

In addition to the CDR, be aware that one CMR per endpoint may be involved in a call. In a normal call between two parties with each using an IP phone, two CMRs get written, one for the originator and one for the destination of the call.

This section describes the records that are written for different call types, including all records for each call and important fields shown in summary tables for easy viewing and comparison.

- Normal Calls (IP Phone to IP Phone), page 34
- Abandoned Calls, page 35
- Calls With Busy or Bad Destinations (Unsuccessful Calls), page 36
- Forwarded Calls, page 37
- Call Pickup, page 39
- Legacy Call Pickup, page 40
- Transferred Calls, page 41
- Conference Calls, page 43
- Call Park, page 46
- Call Park Reversion, page 47
- Precedence Calls (MLPP), page 48
- Malicious Calls, page 49
- Immediate Divert (to Voice-messaging System), page 50
- Barge, page 51
- cBarge, page 53
- Video Calls, page 55
- Forced Authorization Code (FAC), page 55
- Client Matter Code (CMC), page 56
- Call Secured Status, page 57

- DTMF Method, page 58
- RSVP, page 59
- Redirection (3xx) Calls, page 60
- Replaces Calls, page 61
- Refer Calls, page 62

Normal Calls (IP Phone to 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.

Examples of Successful Calls

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

• A 60-second call terminated by the caller, notice that because the calling party hangs up, the orig_CauseValue specifies 16 (Normal Clearing).

FieldNames	Values
globalCallID_callId	1
origLegCallIdentifier	100
destLegCallIdentifier	101
callingPartyNumber	2001
originalCalledPartyNumber	2309
finalCalledPartyNumber	2309
lastRedirectDn	2309
origCause_Value	16
dest_CauseValue	0
duration	60

• A 60-second call cleared by the called party, notice that because the called party hangs up, the dest_CauseValue specifies 16 (Normal Clearing).

FieldNames	Values
globalCallID_callId	1
origLegCallIdentifier	100
destLegCallIdentifier	101
callingPartyNumber	2001
originalCalledPartyNumber	2309
finalCalledPartyNumber	2309
lastRedirectDn	2309
origCause_Value	0

dest_CauseValue	16
duration	60

Abandoned Calls

Consider the logging of calls with zero duration as optional. Normally, these records will not get logged. If logging calls with zero duration is enabled, all calls will generate a CDR.

If the call was abandoned, such as when a phone is taken off hook and placed back on hook, various fields will not contain data. In this case, the **originalCalledPartyNumber**, **finalCalledPartyNumber**, the partitions associated with them, **destIpAddr**, and the **dateTimeConnect** fields remain blank. All calls that were not connected will have a **duration** of zero seconds. When a call is abandoned, the cause code equals zero.

If the user dialed a Directory Number and then abandoned the call before it was connected, the origCalledPartyNumber and finalcalledPartyNumber fields and their associated partitions contain the directory number and partition to which the call would have been extended. The **destIPAddress** field remains blank, and the **duration** equals zero.

Examples of Abandoned Calls

• Extension 2001 goes off hook then on hook.

FieldNames	Values
globalCallID_callId	1
origLegCallIdentifier	100
destLegCallIdentifier	0
callingPartyNumber	2001
originalCalledPartyNumber	
finalCalledPartyNumber	
lastRedirectDn	
origCause_Value	16
dest_CauseValue	0
duration	0

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

FieldNames	Values	
globalCallID_callId	2	
origLegCallIdentifier	200	
destLegCallIdentifier	201	
callingPartyNumber	2001	
originalCalledPartyNumber	2309	
finalCalledPartyNumber	2309	
lastRedirectDn	2309	

origCause_Value	16
dest_CauseValue	0
duration	0

Calls With Busy or Bad Destinations (Unsuccessful Calls)

These calls will all get logged as a normal call with all relevant fields that contain data. The Calling or Called Party Cause field contains a cause code to indicate why the call was not connected, and the Called Party IP and Date/Time Connect fields remains blank. All unsuccessful calls get logged, even if zero duration calls are not being logged.

Examples of Unsuccessful Calls

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

FieldNames	Values
globalCallID_callId	3
origLegCallIdentifier	300
destLegCallIdentifier	301
callingPartyNumber	2001
originalCalledPartyNumber	9728134987
origCause_Value	0
dest_CauseValue	17
duration	0

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

FieldNames	Values
globalCallID_callId	4
origLegCallIdentifier	302
destLegCallIdentifier	303
callingPartyNumber	2001
originalCalledPartyNumber	9728134987
origCause_Value	1
dest_CauseValue	0
duration	0

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

FieldNames	Values
globalCallID_callId	5
origLegCallIdentifier	304
destLegCallIdentifier	305
callingPartyNumber	2001
originalCalledPartyNumber	9728134987
---------------------------	------------
origCause_Value	0
dest_CauseValue	38
duration	0

Forwarded Calls

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

- The originalCallPartyNumber 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 field gives the reason 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 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).

Forwarding Examples

• **CFA Example** - 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.

FieldNames	Values
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

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

FieldNames	Values
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

• **Multiple Hop CFNA & CFB Example** - 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.

FieldNames	Values
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

Call Pickup

Two types of call pickup exist in Cisco Unified CallManager: Pickup and Auto Pickup. The CDRs for both differ slightly for these two types of call pickup.

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 CDR will 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 indicates that the call was 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 indicates that the call was joined on behalf of the Pickup feature. The **lastRedirectReason** contains the redirect reason of 5 (Pickup).

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

Auto Pickup Example

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

	Original Call		
FieldNames	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

Legacy Call Pickup

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

- The originalCallPartyNumber contains the number of the original called party.
- The **finalCalledPartyNumber** specifies the number of the party that picked up the call.
- The **lastRedirectDn** field specifies the number that was ringing when the call was picked up.
- The **origCalledPartyRedirectReason** specifies the reason that the call was redirected the first time. For call pickup calls this field can contain **Call Pickup = 5**.
- The **lastRedirectRedirectReason** specifies the reason that the call was 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 redirect the call for the last redirect. For call pickup, this field specifies **Pickup = 16**.

Legacy 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 is ringing on 2001. 2002 answers the call, and the call connects between the PSTN caller and 2002. They talk for 2 minutes.

FieldNames	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

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.

The original call has the **origCause_value** and **destCause_value** 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.

The consultation call has the **origCause_value** and **destCause_value** 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.

The final transferred call has the **joinOnBehalfOf** field set to Transfer = 10 to indicate this call resulted from a transfer.

Transfer 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 - Call 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.

Original Call CDR	Consultation Call CDR	Final Transferred CDR
1	2	1
101	103	102
102	104	104
2001	2001	3071111
3071111	2002	2002
3071111	2002	2002
3071111	2002	2001
393216	393216	16
393216	393216	0
10	10	0
10	10	0
0	0	10
120	0	360
	Original Call 1 101 102 2001 3071111 3071111 393216 393216 10 10 10 100 100 100 100 100 100 100 100 100 120	Original CallConsultation Call CDR121011031021042001200130711112002307111120023071111200239321639321639321639321610101002000

• **Consultation Transfer from the calling party** - Call 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.

FieldNames	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 - Call 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 a 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 the 50000 and 50002.

FieldNames	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 - Call 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 the 50001 and 50002.

FieldNames	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

Conference Calls

Multiple records get logged for calls that are part of a conference. The number of CDRs that are 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 was used to join other parties to the conference, and one CDR for the last two parties that are connected in the conference. For a three-party ad-hoc conference, six CDRs would exist: one CDR for the original call, three CDRs for the parties that are connected to the conference, one CDR for each the 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 has special significance to the Cisco Unified CallManager, 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 are shown into the conference bridge, regardless of the actually direction. But 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 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 contains multiple conference controller information. This could happen in the case in which 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 connected to the conference includes 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 characteristics:

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

Conference Example

Call 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.



Each conference call leg gets shown as 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	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
last Redirect Redirect On Behalf Of	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	ConfController Dn = 2001; ConfControler Device Name = SEP0003E333FEBD
Conference CDR 1	ConfController Dn = 2001; ConfControler Device Name = SEP0003E333FEBD
Conference CDR 2	ConfController Dn = 2001; ConfControler Device Name = SEP0003E333FEBD
Conference CDR 3	ConfController Dn = 2001; ConfControler Device Name = SEP0003E333FEBD
Final CDR	

Call Park

Call Pickup will generate two CDRs, one for the original call that is parked and another for the call that is picked up or reverted. These CDRs will have the same globalCallID_callId.

Call Park Pickup

- When the call is parked, the call gets split. This generates a CDR for the original call. The **origTerminationOnBehalfOf** and **destTerminationOnBehalfOf** fields gets set to Call Park = 3 for this CDR.
- When the parked call is retrieved, the user goes off hook and enters the park code. This call gets joined with the parked call. Because the user who is picking up the call is joined with the parked call, the user gets treated as the originator of the call, and the parked user gets treated as the destination. This means that the **callingPartyNumber** of the call contains the directory number of the user who is picking up the call, and the **originalCalledNumber** and **finalCalledNumber** contains the directory number of the parked user. The **lastRedirectDn** contains the park code that is used to pick up the call. The **lastRedirectRedirectRedirectRedirectOnBehalfOf** should also specify Call Park = 3.

Call Park Example

• **Call Park Example** – 50003 calls 50002, 50002 presses the Park softkey. 50001 picks up the parked call by dialing the park code (44444).

FieldNames	Original Call that is parked	Parked call that is picked up
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 will expire and redirect 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 generates a CDR for the original call. The **origTerminationOnBehalfOf** and **destTerminationOnBehalfOf** fields get set to Call Park = 3 for this CDR, the same as Call Park Pickup scenario.

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

Call Park Reversion Example

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

FieldNames	Original Call that is	Reverted Call CDR
globalCalIID_calIId	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

Precedence Calls (MLPP)

With precedence calls everything basically stays the same for all calls: normal calls, forwarded calls, transferred calls, and so on. The differences include the precedence level fields are set in the CDR, and also when a call is preempted by a higher-level precedence call, the cause codes indicate the reason for the preemption.

Precedence Call Examples

• Call to another IP phone by dialing a precedence pattern (precedence level 2)

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

• Received precedence call from another network (precedence level 1)

Precedence Call CDR
102
11111
11112
9728552001
6001
16
0
1
1

• Call gets preempted by a higher precedence level call

FieldNames	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

Malicious Calls

When a call is identified as a malicious call (button press), the local network (Cisco Unified CallManager) flags the call. The "comment" gets used to flag the malicious call.

Malicious Call Example

• Customer call marked as malicious

FieldNames	Original call CDR
globalCallID_callId	1
origLegCallIdentifier	100
destLegCallIdentifier	101
callingPartyNumber	9728552001
origCalledPartyNumber	5555
origCause_Value	0
dest_CauseValue	16
Comment	callFlag=MALICIOUS

Immediate Divert (to Voice-messaging System)

Idivert can get 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** 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 will have the same **globalCallID_CallId** field. The first for the original connected and a second for the call redirected to the Cisco voice-messaging system. The first call will have the **origTerminationOnBehalfOf** and **destTerminationOnBehalfOf** field set to Immediate Divert = 14.
- The call that is redirected to the voice-messaging system will have the **origCalledPartyRedirectOnBehalfOf** and the **lastRedirectRedirectOnBehalfOf** fields set to Immediate Divert = 14.

IDivert Examples

• **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).



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

FieldNames	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

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 voice-messaging system (40000).

Because the call was connected before the redirect, two CDRs are generated: one for the original connected call, and another for the call diverted to the voice-messaging system.

	Original connected	
FieldNames	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

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 have a value of Barge = 15, and the redirect reason fields are Barge = 114.

Barge Examples

• **Barge Example 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.



٠

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

FieldNames	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

• **Barge Example 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.

FieldNames	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

• **Barge Example 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.

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

	Original Call		
FieldNames	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

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 Examples

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

Note

FieldNames	Orig Call CDP	CBarge Call	CBarge Call	CBarge Call	Final Call CDP
		CDINT	CDK 2	CDK 3	
globalCallID_callId	49	49	49	49	49
origLegCallIdentifier	1677346	1677348	1677347	1677346	1677347
destLegCallIdentifier	1677347	1677353	1677351	1677352	1677346
callingPartyNumber	40003	40001	40001	40003	40001
originalCalledPartyNumber	40001	b0029901001	b0029901001	b0029901001	40003
finalCalledPartyNumber	40001	b0029901001	b0029901001	b0029901001	40003
lastRedirectDn	40001	b0029901001	40001	40001	b0029901001
origCause_Value	393216	16	393216	393216	16
dest_CauseValue	393216	0	393216	393216	0
origCalledPartyRedirectReason	0	98	98	98	0
lastRedirectRedirectReason	0	98	98	98	98
destTerminationOnBehalfOf	4		4	4	4
origCalledRedirectOnBehalfOf		4	4	4	
lastRedirectRedirectOnBehalfOf		4	4	4	4
joinOnBehalfOf		4	4	4	4
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

Video Calls

This is an example CDR for a video call.

Video Call CDR Example

• **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.

FieldNames	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

Forced Authorization Code (FAC)

When FAC feature is invoked, the system writes the authorization description and level into the CDR. For security reasons, the actual authorization code will 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 Example

45000 calls 9728134987, the user gets prompted for a authorization code and enters 12345. FAC code 12345 is configured as level 1 and name Legal1. The caller answers the call and talks for 2 minutes.

FieldNames	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

Client Matter Code (CMC)

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

CMC Example

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

FieldNames	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

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, so that later the call changed to secured, the CDR contains 1 for "Secured" even though different portions of the call had different status values. The **callSecuredStatus** will identify the security status of the call.

Examples

• Encrypted Call Example - The call between 20000 and 20001 is encrypted. They talk for 5 minutes.

FieldNames	Values
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

• Authenticated Call Example - The call between 20000 and 20001 is authenticated (not encrypted). They talk for 10 minutes.

FieldNames	Values
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

DTMF Method

These fields identify the DTMF method that is used for the call.

DTMF Call Examples

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

FieldNames	Values
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

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

FieldNames	Values
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

RSVP

These fields identify the status of RSVP reservation for the call. Be aware that the Unified CM RSVP CDR status field value is concatenated and the last 32 status values are retained 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, and the call ended with successful RSVP reservation, the CDR shows the following string as the Unified CM RSVP reservation status for that particular stream: "2:5:2:5:2:5:2" (success:lost_bw:success:lost_bw:success).

RSVP Call Examples

• The example represents a call that is established with "Optional" policy, and the initial RSVP reservation is successful. The parties talk for 5 minutes.

FieldNames	Values
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

• The example represents a call is established with "Optional" policy, and the initial RSVP reservation is successful, then it loses its bandwidth reservation, but regains it after a retry. Parties talk for 1 minute.

FieldNames	Values
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

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 represent CCM Redirection = 19. The **origCalledPartyRedirectReason** and the **lastRedirectRedirectReason** represent Redirection = 162.

Redirection (3xx) Examples

• **Redirection Example** – Activate CFA on phone 10010 that is running SIP (registered to Unified CM) 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 a minute.

FieldNames	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

Replaces Calls

This example shows a CDR for a Replaces call.

Replaces Examples

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



When the transfer is complete, the system sends an Invite with Replaces to Cisco Unified CallManager.

FieldNames	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

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



When the transfer is complete, a Refer with Replaces gets sent to Cisco Unified CallManager.

		Consultation	Cincl Transformed
FieldNames	Original Call CDR	Consultation Call CDR	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

Refer Calls

See the "Replaces Calls" section on page 61 for an example of Refer with Replaces.

CDR Field Descriptions

Table 4 defines all fields in the current CDRs in the order in which they appear in the CDR.

Field Name Description **Range of Values** 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 Default - For CDRs, this field always specifies 1. globalCallID_callManagerId Positive Integer This field designates a unique Cisco Unified CallManager identity. The Global Call ID comprises two fields: globalCallID_callId and globalCallID_callManagerId All records that are associated with a standard call have the same Global Call ID in them. Default - You should always ensure that this field is populated. globalCallID_callId Positive Integer This field designates unique call identity value that is assigned to each call. This identifier gets allocated independently on each call server. Values get chosen sequentially when a call begins, and value gets assigned for each call, successful or unsuccessful. When Cisco Unified CallManager restarts, this values resets to 1. The Global Call ID comprises two fields: globalCallID_callId and globalCallID_callManagerId All records that are associated with a standard call have the same Global Call ID in them. Default - You should always ensure that this field is populated. origLegCallIdentifier Positive Integer This field identifies the originating 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 - You should always ensure that this field is populated.

Table 4CDR Field Descriptions

Field Name	Range of Values	Description
dateTimeOrigination	Integer	This field identifies the date and time when the user goes off hook or the date and time when the H.323 Setup message is received for an incoming call. The time gets stored as UTC.
		Default - You should always ensure that this field is populated.
origNodeId	Positive Integer	This field identifies the node within a cluster to which the originator of the call is registered at the time the call is made.
		Default - You should always ensure that this field is populated.
origSpan	0, Positive integer	For calls originating at a gateway, this field indicates the B channel number of the T1, PRI, or BRI trunk where the call is originated, 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 do not originate at a gateway, the value equals 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 CallManager.
		The "IP Addresses" section on page 11 describes the IP address format.
		Default - This field gets populated based on these rules.

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 were applied to the Calling Party Number before it arrives at the Cisco Unified CallManager (such as translations at the gateway).
		For server calls, where Cisco Unified CallManager originates a half call without a calling party, this field may be 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 remains 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 lists the valid values per Q.850.
		For clearing causes that are created internally by the Cisco Unified CallManager, this value specifies zero.
		Default - 0.

I

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 CallManager currently uses the Q.850 codes and some Cisco Unified CallManager defined codes. The "Call Termination Cause Codes" section on page 87 lists them.
		For calls cleared by the terminating party, this field equals 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 has a precedence level. This field represents the original leg precedence level.
		• Precedence 0 = FLASH OVERRIDE/ EXECUTIVE OVERRIDE
		• Precedence 1 = FLASH
		• Precedence 2 = IMMEDIATE
		• Precedence 3 = PRIORITY
		• Precedence 4 = ROUTINE
		Default - 4.
origMediaTransportAddress_IP	0, Integer	This field identifies the IP address of the device that originated 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 11 describes the IP address format.
		Default - 0. If media is not established, this field equals 0.

Field Name	Range of Values	Description
origMediaTransportAddress_Port	0, 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 equals 0.
origMediaCap_payloadCapability	0, Positive integer	This field identifies the codec type that the originator used to transmit media.
		Cisco Unified CallManager currently uses the following payload capability values: 0, 1-16, 18-20, 25, 32, 33, 81-86. The "Codec Types" section on page 86 lists the valid values.
		Default - 0. If media is not established, this field equals 0.
origMediaCap_maxFramesPerPacket	0, Positive integer	This field identifies the number of milliseconds of data per packet sent by the originating party. 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 equals 0.
origMediaCap_g723BitRate	0	Deprecated since Cisco CallManager Release 3.3.4.
		This field will always equal 0.
origVideoCap_Codec	0, 100 = H.261, 101 = H.263	This field identifies the codec type that the originator used to transmit video (H.261, H.263, or Vieo.)
	101 = 11.203, 102 = Vieo	Default - 0. If media is not established, this field equals 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 equals 0.
origVideoCap_Resolution	0,	This field identifies the video resolution.
	1 = SQCIF, 2 = QCIF, 2 = CIF,	Default - 0. If media is not established, this field equals 0.
	3 = CIF, 4 = CIF4, 5 = CIF16	
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 equals 0.

I

Field Name	Range of Values	Description
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 equals 0.
origRSVPAudioStat	0 to 5	This field gives the status of 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
origRSVPVideoStat	0 to 5	This field gives the status of 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

 Table 4
 CDR Field Descriptions (continued)

Field Name	Range of Values	Description
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 subcalls and, consequently, several CDRs (as during a call transfer), this value remains constant.
		Default - 0. If the destination cannot be reached, this field equals 0.
destNodeId	0, Positive Integer	This field identifies the 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 equals 0.
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 that do not terminate at a gateway, the value equals 0.
		Default - 0. If the destination cannot be reached, this field equals 0.
destIpAddr	Integer	This field identifies the IP address of the device that terminated 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 CallManager.
		The "IP Addresses" section on page 11 describes the IP address format.
		Default - 0. If the destination cannot be reached, this field equals 0.

I

Field Name	Range of Values	Description
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 the destination cannot be reached, this field remains empty.
finalCalledPartyNumber	Text String	This field specifies the number to which the call is finally 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 the destination cannot be reached, this field remains empty.
finalCalledPartyUnicodeLoginUserI D	Unicode – UTF_8	This field specifies the final called party login user ID. The format of this field specifies UTF_8.
		Default - Empty string " ". If the user ID does not exist, this field remains empty.
destCause_location	0 to 15	For clearing causes that are received over ISDN signaling links, this field provides the Location field that the ISDN release message indicates. The "Call Termination Cause Codes" section on page 87 lists the valid values per Q.850.
		For clearing causes that Cisco Unified CallManager created internally, this value equals zero.
		Default - 0. If the destination cannot be reached, this field equals 0.

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 87 lists the valid values per Q.850.
		For calls that the originating party cleared, this field equals 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 equals 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 terminated 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 11 describes the IP address format.
		Default - 0. If the destination cannot be reached, this field equals 0.
destMediaTransportAddress_Port	0, Positive Integer	This field identifies the IP port number associated with the DestMediaTransportAddress_IP field.
		Default - 0. If the destination cannot be reached, this field equals 0.

I

Field Name	Range of Values	Description
destMediaCap_payloadCapability	0, Positive Integer	This field identifies the codec type that the terminating party used to transmit media.
		Cisco Unified CallManager currently uses the following payload capability values: 0, 1-16, 18-20, 25, 32, 33, 81-86. The "Codec Types" section on page 86 lists the valid values.
		Default - 0. If the destination cannot be reached, this field equals 0.
destMediaCap_maxFramesPerPacket	0, Positive Integer	This field identifies the number of milliseconds of data per packet that the terminating party of the call sent. This field is normally set to 10, 20, or 30 for G.729 or G.711 codecs, but can store any nonzero value.
		This field can be zero if the media is never established.
		Default - 0. If the destination cannot be reached, this field equals 0.
destMediaCap_g723BitRate	0	Depreciated since Cisco Unified CallManager Release 3.3(4).
		Default - This field always equals 0.
destVideoCap_Codec	0, 100 = H.261,	This field identifies the codec type that the terminating party used to transmit video (H.261, H.263, or Vieo). Default - 0. If the destination cannot be reached, this field equals 0.
	101 = H.263, 102 = Vieo	
destVideoCap_Bandwidth	0, Positive Integer	This field identifies the bandwidth that is measured in units of kbps.
		Default - 0. If the destination cannot be reached, this field equals 0.
destVideoCap_Resolution	0, 1 = SQCIF, 2 = QCIF, 3 = CIF, 4 = CIF4, 5 = GUE1 (This field identifies the video resolution. Default - 0. If the destination cannot be reached, this field equals 0.
destVideoTransportAddress _IP	0, Integer	This field identifies the IP address of the
		Default - 0. If the destination cannot be reached, this field equals 0.
Field Name	Range of Values	Description
--------------------------------	------------------------	---
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 equals 0.
destRSVPAudioStat	0 - 5	This field designates the status of 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
destRSVPVideoStat	0 - 5	This field designates the status of 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

Field Name	Range of Values	Description
dateTimeConnect	0, Integer	This field identifies the date and time that the call connected. 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 equals 0.
dateTimeDisconnect	0, Integer	This field identifies the date and time when the call was cleared. This field gets set even if the call never connected. The time gets stored as UTC.
		Default - 0. If the call is never connected, this field equals 0.
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 was 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.

 Table 4
 CDR Field Descriptions (continued)

Field Name	Range of Values	Description
originalCalledPartyNumberPartition	Text String	This field uniquely identifies the partition name associated with the OriginalCalledPartyNumber field because Cisco Unified CallManager 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 associated with the route pattern that pointed to the gateway.
		Default - empty string "". If the original called party does not have a partition, this field remains empty.
callingPartyNumberPartition	Text String	This field uniquely identifies the partition name associated with the CallingPartyNumber field because Cisco Unified CallManager 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 associated with the FinalCalledPartyNumber field because Cisco Unified CallManager 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 associated with the route pattern that pointed to the gateway.
		Default - empty string "". If the final called party does not have a partition, this field remains empty.

Field Name	Range of Values	Description
lastRedirectDnPartition	Text string	This field uniquely identifies the partition name that is associated with the LastRedirectDn field because Cisco Unified CallManager 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 pointed to the gateway.
		Default - empty string "". If the last redirecting Party does not have a partition or the call was never redirected, this field remains empty.
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 connected or if it was connected for less than 1 second.
		Default - 0
origDeviceName	Text string	This field specifies a text string that identifies the name of the originating device.
		Default - Ensure this field is populated.
destDeviceName	Text string	This field specifies text string that identifies the name of the destination device.
		Default - empty string "". If the original device does not have a name, this field remains 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 specifies 12 for Device. If the call is terminated because of a transfer, the OnBehalfOf code specifies 10 for Transfer.
		See the "OnBehalfof Codes" section on page 92 for a list of the codes.
		Default - 0

Field Name	Range of Values	Description
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 specifies 12 for Device. If the call is terminated because of a transfer, the OnBehalfOf code specifies 10 for Transfer.
		See the "OnBehalfof Codes" section on page 92 for a list of the codes.
		Default - 0
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 "OnBehalfof Codes" section on page 92 for a list of the codes.
		Default - 0
lastRedirectRedirectOnBehalfOf	0, Positive 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 "OnBehalfof Codes" section on page 92 for a list of the 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 91 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 91 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.
		Default - 0

Field Name	Range of Values	Description
globalCallId_ClusterId	Text String	This field specifies a unique ID that identifies a cluster of Cisco Unified CallManagers.
		This field is generated at installation, but is not used by Cisco Unified CallManager. The following fields make up this unique key:
		GlobalCallId_ClusterId + GlobalCallId_CallManagerId + globalCallId_callId
		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 occurs on behalf of a transfer, the OnBehalfOf code specifies 10.
		See the "OnBehalfof Codes" section on page 92 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 authorization code. For security purposes, the authorization code is not written to the CDR; instead, the authorization description and level are written.
		Default: Empty string "" or null.
authorizationLevel	0, Integer	This field provides the level of the authorization code. For security purposes, the authorization does not get written to the CDR; instead, the authorization description and level are written.
		Default: 0
clientMatterCode	Text String	This field displays a unique code. Before the system extends a call, the user enters a client matter code that can be used for assigning account or billing codes to calls.
		Default: Empty string "" or null

Field Name	Range of Values	Description
origDTMFMethod	0, Positive integer	This field displays the DTMF method that is used by the originator. 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 is used by the destination.
		 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 reached during a call. For example, if the call is originally unsecured, then later the call is changed to secured, the CDR contains the value 1 for "Secured", even though different portions of the call had different status values.
		0 - Non-secured
		2 - Secured (encrypted)
		Default – 0 (Non-secured)
origConversationId	0, Integer	This field displays the unique identifier used to identify parties in a conference call. Default - 0
authorizationCodeValue	Text string	This field displays the Forced Authorization Code (FAC) associated with the call. The field contains a numeric string of up to 32 characters. Default - Empty string ""or null

Table 4	CDR Field Descriptions (continued)
---------	------------------------------------

CMR Field Descriptions (Diagnostic)

Table 5 contains the fields, range of values, and field descriptions of the CMRs in the order in which they appear in the CMR.

	Range of	
Field Name	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 equals 2.
globalCallID_callManagerId	Positive Integer	This field specifies a unique Cisco Unified CallManager 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.
		Default - Ensure this field always is populated.
globalCallId_callId	Positive Integer	This field specifies a unique call identity value that is assigned to each call. This identifier is allocated independently on each call server. Values are chosen sequentially when a call begins. Each call, successful or unsuccessful, receives a 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.
		Default - Ensure this field always is populated.

 Table 5
 CMR Field Descriptions

1

Field Name	Range of Values	Description
nodeId	Positive Integer	This field specifies the node within the Cisco Unified CallManager cluster where this record is 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.
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 has been on hook. Cisco Unified CallManager 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 has transmitted 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 has transmitted in RTP data packets since starting transmission on this connection. The value remains zero if the connection is set to "receive only" mode.
		Default - 0

Field Name	Range of Values	Description
numberPacketsReceived	Integer	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 to "send only" mode.
		Default - 0
numberOctetsReceived	Integer	This field specifies the total number of payload octets (that is, not including header or padding) that the device receives 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 to "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 packets that are late or duplicates. Thus, packets that arrive late do not get counted as lost, and the loss may be negative if duplicates 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 is set in "send only" mode. For detailed information, see RFC 1889. Default - 0

Field Name	Range of Values	Description
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 is set in "send only" mode.
		Default - 0
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 indicate and the NTP timestamp of the receivers, measured when these messages are received. Cisco Unified CallManager 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 - This field will always be
		populated with a unique ID.
directoryNumberPartition	Text String	This field identifies the partition of the directory number.
		Default - Empty string, "". This field may be empty if no partition exists.
deviceName	Text String	This field identifies the name of the device.
		Default - Empty string "". This field may be empty if there is no device name.

Field Name	Range of Values	Description
globalCallId_ClusterId	Text String	This field designates a unique ID that identifies a cluster of Cisco Unified CallManagers.
		This field gets generated during installation, but is not used by Cisco Unified CallManager: globalCallId_ClusterId + globalCallId_callManagerId + globalCallId_callId.
		Default - Ensure this field always is populated.
varVQMetrics	Text String	This field contains a variable number of voice quality metrics. This field represents 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 6 "K-Factor Data Stored in Cisco Unified CallManager CMRs" for a complete list of K-Factor data.

K-Factor Data in CMRs

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 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, k-factor estimates are generated 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 provide primary statistics because they alert the network operator before network impairment has an audible impact or is visible through MOS.

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, and 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 duration during which some concealment is observed during a call.
SCS	Severely Conceal Secs	Severely Conceal Secs. represents the duration during which a significant amount of concealment is observed. If the concealment observed is usually greater than 50 milliseconds or approximately 5 percent, the speech probably does not seem very audible.

Table 6 K-Factor Data Stored in Cisco Unified CallManager CMRs

Field Name	Phone Display Name	D&I User Interface Text and Description
MLQK	MOS LQK	MOS Listening Quality K-factor represents 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 6 K-Factor Data Stored in Cisco Unified CallManager CMRs (continued)

Codec Types

Table 7 contains the compression and payload types that may appear in the codec fields.

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
15	G.729AnnexB
16	G.729 Annex AwAnnexB
18	GSM Full Rate
19	GSM Half Rate

Table 7Codec Types

Value	Description
20	GSM Enhanced Full Rate
25	Wideband 256K
32	Data 64k
33	Data 56k
40	G7221 32K
41	G7221 24K
42	Media_Payload_AAC
80	GSM
81	ActiveVoice
82	G726_32K
83	G726_24K
84	G726_16K
100	H261
101	H263
102	Video
103	H264
106	H224

Call Termination Cause Codes

Table 8, Table 9, and Table 10 contain call termination cause codes that may appear in the Cause fields in CDRs.

 Table 8
 Call Termination Cause Codes

Code	Description
0	No error
1	Unallocated (unassigned) number
2	No route to specified transit network (national use)
3	No route to destination
4	Send special information tone
5	Misdialed trunk prefix (national use)
6	Channel unacceptable
7	Call awarded and being delivered in an established channel
8	Preemption
9	Preemption—circuit reserved for reuse
16	Normal call clearing

Code	Description
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	Meet-Me secure conference minimum security level not met
62	Inconsistency in designated outgoing access information and subscriber class
63	Service or option not available, unspecified
65	Bearer capability not implemented
66	Channel type not implemented
69	Requested facility not implemented
70	Only restricted digital information bearer capability available (national use)
79	Service or option not implemented, unspecified

 Table 8
 Call Termination Cause Codes (continued)

Code	Description
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 not compatible with call state, or the message type nonexistent or not implemented
99	An information element or parameter non-existent or not implemented
100	Invalid information element contents
101	Message 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
127	Interworking, unspecified
129	Precedence out of bandwidth

Table 0	Call Tampin ation Causes	Codes (continued)
Idule o	Call lennination cause	coues (continueu)

Table 9 Cisco-Specific Call Release Cause Codes

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 splitoff and terminated (was not part of the final transferred call). This can help determinewhich calls were terminated as part of a feature operation.
458752 0x70000	Drop any party/drop last party (was 128)

I

L

Code	Description
0x1000029	CCM_SIP_400_BAD_REQUEST
0x2000015	CCM_SIP_401_UNAUTHORRIZED
0x3000015	CCM_SIP_402_PAYMENT_REQUIRED
0x4000015	CCM_SIP_403_FORBIDDEN
0x5000001	CCM_SIP_404_NOT_FOUND
0x600003F	CCM_SIP_405_METHOD_NOT_ALLOWED
0x700004F	CCM_SIP_406_NOT_ACCEPTABLE
0x8000015	CCM_SIP_407_PROXY_AUTHENTICATION_REQUIRED
0x9000066	CCM_SIP_408_REQUEST_TIMEOUT
0xB000016	CCM_SIP_410_GONE
0xC00007F	CCM_SIP_411_LENGTH_REQUIRED
0xE00007F	CCM_SIP_413_REQUEST_ENTITY_TOO_LONG
0xF00007F	CCM_SIP_414_REQUEST_URI_TOO_LONG
0x1000004F	CCM_SIP_415_UNSUPPORTED_MEDIA_TYPE
0x1100007F	CCM_SIP_416_UNSUPPORTED_URI_SCHEME
0x1500007F	CCM_SIP_420_BAD_EXTENSION
0x1600007F	CCM_SIP_421_EXTENSION_REQUIRED
0x1800007F	CCM_SIP_423_INTERVAL_TOO_BRIEF
0x40000012	CCM_SIP_480_TEMPORARILY_UNAVAILABLE
0x41000029	CCM_SIP_481_CALL_LEG_DOES_NOT_EXIST
0x42000019	CCM_SIP_482_LOOP_DETECTED = 0x42000000+EXCHANGE_ROUTING_ERROR
0x43000019	CCM_SIP_483_TOO_MANY_HOOPS
0x4400001C	CCM_SIP_484_ADDRESS_INCOMPLETE
0x45000001	CCM_SIP_485_AMBIGUOUS
0x46000011	CCM_SIP_486_BUSY_HERE
0x4700001F	CCM_SIP_487_REQUEST_TERMINATED
0x4800001F	CCM_SIP_488_NOT_ACCEPTABLE_HERE
0x4B000011	CCM_SIP_491_REQUEST_PENDING
0x4D000011	CCM_SIP_493_UNDECIPHERABLE
0x54000029	CCM_SIP_500_SERVER_INTERNAL_ERROR
0x5500004F	CCM_SIP_501_NOT_IMPLEMENTED
0x56000026	CCM_SIP_502_BAD_GATEWAY
0x57000029	CCM_SIP_503_SERVICE_UNAVAILABLE
0x58000066	CCM_SIP_504_SERVER_TIME_OUT
0x5900007F	CCM_SIP_505_SIP_VERSION_NOT_SUPPORTED

Table 10 SIP Call Release Cause Codes

Code	Description
0x5A00007F	CCM_SIP_513_MESSAGE_TOO_LARGE
0xA1000011	CCM_SIP_600_BUSY_EVERYWHERE
0xA2000015	CCM_SIP_603_DECLINE
0xA3000001	CCM_SIP_604_DOES_NOT_EXIST_ANYWHERE
0xA400001F	CCM_SIP_606_NOT_ACCEPTABLE

Table 10 SIP Call Release Cause Codes (continued)

Redirect Reason Codes

Table 11 contains the available Redirect Reason Codes that may appear in a CDR record.

Q.931 Standard Redirect Reason Codes		
Value	Description	
0	Unknown	
1	Call Forward Busy	
2	Call Forward No Answer	
4	Call Transfer	
5	Call Pickup	
7	Call Park	
8	Call Park Pickup	
9	CPE Out of Order	
10	Call Forward	
11	Call Park Reversion	
15	Call Forward All	
Non Sta	ndard Redirect Reason Codes	
18	Call Deflection	
34	Blind Transfer	
50	Call Immediate Divert	
66	Call Forward Alternate Party	
82	Call Forward On Failure	
98	Conference	
114	Barge	
130	Refer	
146	Replaces	
162	Redirection (3xx)	
178	Not Known (SIP-forward busy greeting)	

Table 11Redirect Reason Codes

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)

Table 11 Redirect Reason Codes (continued)

OnBehalfof Codes

Table 12 contains the available OnBehalfof Codes that may appear in a CDR record.

Value	Description
0	Unknown
1	CctiLine
2	Unicast Shared Resource Provider
3	Call Park
4	Conference
5	Call Forward
6	Meet-Me Conference
7	Meet-Me Conference Intercepts
8	Message Waiting
9	Multicast Shared Resource Provider
10	Transfer
11	SSAPI Manager
12	Device
13	Call Control
14	Immediate Divert
15	Barge
16	Pickup
17	Refer
18	Replaces
19	Redirection
20	Callback
21	Path Replacement
22	FacCmc Manager
23	Malicious Call

Table 12 **OnBehalfof Codes**

1

Related Documentation

The following documents contain additional information related to CDRs:

- Cisco Unified CallManager CDR Analysis and Reporting Tool Administration Guide
- Cisco Unified CallManager Administration Guide
- Cisco Unified CallManager Serviceability Administration Guide
- Cisco Unified CallManager Serviceability System Guide
- Cisco Unified CallManager System Guide
- Disaster Recovery System Administration Guide

Obtaining Documentation, Obtaining Support, and Security Guidelines

For information on obtaining documentation, obtaining support, providing documentation feedback, security guidelines, and also recommended aliases and general Cisco documents, 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

CCVP, the Cisco logo, and the Cisco Square Bridge logo are trademarks of Cisco Systems, Inc.; Changing the Way We Work, Live, Play, and Learn is a service mark of Cisco Systems, Inc.; and Access Registrar, Aironet, BPX, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Enterprise/Solver, EtherChannel, EtherFast, EtherSwitch, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, IP/TV, iQ Expertise, the iQ logo, iQ Net Readiness Scorecard, iQuick Study, LightStream, Linksys, MeetingPlace, MGX, Networking, and TransPath are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or Website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0705R)

Copyright 2007.

Г