Cisco CallManager
Call Detail Record Definitions, Release 5.0(1)

This document describes the format and logic of the call detail records (CDRs) and call management records (CMRs) that the Cisco CallManager Release 5.0(1) system generates. An integration partner 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, how to interpret fields in the files, and some of the known issues.

When you install your system, CDRs and CMRs are disabled by default. You can enable or disable CDRs and/or CMRs at any time while the system is in operation. You do not need to restart Cisco CallManager for the change to take effect. The system responds to all changes within a few seconds.

Contents

This document covers the following topics:

Corporate Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

New and Changed Information

This section describes any new features or changes for CDRs/CMRs that are pertinent to the specified release of Cisco CallManager.

Cisco CallManager Release 5.0(1)

The major change in CDRs/CMRs in this release is the transition from an SQL database to flat files. If you upgrade from Cisco CallManager 4.x, the CDRs in the CAR database are converted to CSV files, which are then stored in the CDR Repository.

The following new or changed features are also included in Cisco CallManager Release 5.0(1):

CDR File Sequence Numbers

In a previous release, independent sequence numbers were added to the CDR and CMR file names. However, the independent sequence numbers cause a problem with synchronizing CDRs and CMRs, in that all CMR files might need to be searched to find the CMR corresponding to a given CDR.

In order to solve this issue, in this release a single sequence number is used for both CDR and CMR files. This means the CDR and the corresponding CMR will reside within a reasonable number of files and can be more easily correlated.

This single sequence number is maintained in an index file, "_Index.idx". The **".idx"** extension needs to be added to the index filename to avoid removal by cleanup utilities.

Example filenames:

cdr_StandAloneCluster_01_200504070229_0

cmr_StandAloneCluster_01_200504070229_0

cdr_StandAloneCluster_01_200504070230_1

cmr_StandAloneCluster_01_200504070230_1

Voice Quality Metrics

In addition to the existing voice quality data supported by the SCCP phone, new measurements are added in the new varVQMetrics field. This string field contains voice quality metrics separated by semicolons.

The format of the string is either fieldName=value or fieldName=value/precision.

QoS Support (RSVP)

Cisco CallManager Release 5.0(1) QoS Support (RSVP) feature adds four new CDR fields to reflect the status of RSVP bandwidth reservation per audio or video stream for both the origin and destination.

DTMF Method

A new CDR field is added per party to show what DTMF method is being used by that party for given call. The possible values for these two new CDR fields are:

0 – No DTMF – Use ANY matched DTMF.

1 - Preferred OOB – Use OOB if endpoints behind SIPTrunk support it.

2 - Preferred 2833 – Use 2833 if endpoints behind SIPTrunk support it.

3 - Preferred Both – Use both KPML and 2833 if endpoints behind SIPTrunk can support both.

Call Secured Status

A new Call Secured Status field is added to the CDR. This field contains an indication of the highest level of security reached during the call. The three possible values for this field are:

0—non-secure

1—authenticated (not encrypted)

2—secure (encrypted)

SIP Features

Three new SIP features (Refer, Replaces, and Redirection) are included in this release. There are no new fields for these features, but calls that involve these features will generate CDRs. The major difference is, depending on the feature and the call scenario, the party fields may contain SIP URIs instead of numbers. For example, if the redirecting number is watson@acm.com, then the lastRedirectDn field contains this URI instead of a number.

These new SIP features (Refer, Replaces, and Redirection) also have new onBehalfOf field values.

Pickup Feature

The Pickup feature is enhanced and separated from the Forwarding feature in Cisco CallManager 5.0(1). Specifically, the pickup by redirect and monitoring of pickup groups are moved from the Forwarding feature to the Pickup feature. These changes will cause a small difference in the CDR for pickup calls.

In previous releases, the redirect OnBehalfOf fields contained 5 (Call Forward) since the Forwarding feature redirected the call. In Release 5.0(1) the Pickup feature now redirects the call, so the redirect onBehalfOf fields change from 5 (Call Forward) to 16 (Call Pickup).

New Call Release Cause Codes

The Q.931 call release cause codes remain the same; however, the Cisco-specific call release cause codes have changed values and several new codes have been added. A new table of SIP call release cause codes has also been added. Please refer to the "Call Release Cause Codes" section on page 85 for complete details.

Cisco CallManager Release 4.1(3)

For this release, the content of the CDRs changed for the new Auto Pickup feature, but no new CDR fields were added. Enhancements to the existing Call Pickup and Group Call Pickup features provide the Auto Pickup feature.

You can enable and disable Auto Pickup by using the new service parameter Auto Pickup Enabled. By default, the system sets the Auto Pickup Enabled parameter to False. When the parameter is set to True, Auto Pickup applies to all types of Call Pickup.

Auto Pickup

The following list gives the three types of auto pickup:

- Auto Call
- Auto Group
- Auto Other

The new Auto Pickup feature generates only two CDRs: one CDR for the ringing call and another CDR for the final connected call that is picked up. Both CDRs have the same Call ID.

For the first CDR, the origTerminationOnBehalfOf and destTerminationOnBehalfOf fields get set to 16 (Pickup), which indicates that the call terminated on behalf of the Pickup feature.

For the second CDR, the lastRedirectOnBehalfOf and joinOnBehalfOf fields get set to 16 (Pickup), which indicates that the system joined the call on behalf of the Pickup feature. The lastRedirectDn indicates from where the call was picked up, that is, lastRedirectDn contains the party that was ringing when the call was picked up. The lastRedirectRedirectReason contains the redirect reason 5 (Pickup).

Pickup

The existing pickup features generate only one CDR. The origCalledPartyRedirectOnBehalfOf, lastRedirectRedirectOnBehalfOf, and joinOnBehalfOf fields get set to 5 (Call Forward), which indicates that the Call Forward feature redirected the call. The origCalledPartyRedirectReason and lastRedirectRedirectReason contain the redirect reason code of 5 (Pickup).

Cisco CallManager Release 4.1(2)

The following list provides the features or changes for CDRs in Cisco CallManager release 4.1(2):

- Forced Authorization Codes (FAC)—Forces the user to enter a valid authorization code prior to extending calls to classes of dialed numbers, such as external calls, toll calls, and international calls. Authorization information gets written to the Cisco CallManager database.
- Client Matter Codes (CMC)—Before extending a call, allows the user to enter a “client matter” code that the customer can use for assigning account or billing codes to calls that are placed. Client Matter Code information gets written to the Cisco CallManager database.

The 4.1(2) Cisco CallManager release provides three new CDR fields to support FAC and CMC:

- authCodeDescription
- authorizationLevel
- clientMatterCode

Cisco CallManager Release 4.0(1)

The following list provides the features or changes for CDRs in Cisco CallManager release 4.0(1):

- Identifies Multilevel Precedence and Preemption (MLPP)
  - Adds the field `origPrecendenceLevel` for the precedence level of the originating leg of the call
  - Adds the field `destPrecedenceLvel` for the precedence level of the destination leg of the call
  - MLPP utilizes existing cause codes 8, 9, 46, 50, and 128
- Identifies malicious calls by adding a new `Comment` field
- Drop any party feature utilizes existing cause fields: `origCause_value` and `destCause_value`
- The `OnBehalfOf` field contains a new code (value = 14) for the Immediate Divert feature and value = 15 for Barge.
- The following new fields support video in Cisco CallManager:
  - `origVideoCap_Codec`
  - `destVideoCap_Codec`
  - `origVideoCap_Bandwidth`
  - `destVideoCap_Bandwidth`
  - `origVideoCap_Resolution`
  - `destVideoCap_Resolution`
  - `origVideoTransportAddress_IP`
  - `origVideoTransportAddress_Port`
  - `destVideoTransportAddress_IP`
  - `destVideoTransportAddress_Port`
- Adds user login fields (`callingPartyLoginUserID` and `finalCalledPartyLoginUserID`) to ensure that the system associates a valid UserID with a newly created phone device and that it gets properly reported in CDRs
- Adds examples for different call scenarios including IDivert, Barge, and cBarge

Note In the following sections, references to CDR files imply both CDR and CMR files.

Cisco Call Manager CDR Configuration

The following sections define the current CDR configuration parameters.

Service Parameters

These values are set to False by default. You must enable these configuration items separately on every server in a cluster. You can configure these parameters on the Service Parameters Configuration page in the Cisco CallManager Administration. To access the Service Parameters Configuration page, open Cisco CallManager Administration and select **Service -> Service Parameter**s.

- **CdrEnabled –** This parameter enables or disables CDRs. If this parameter is false (the default value), CCM will not generate CDRs. If this parameter is true, CDRs are generated.
- **CdrLogCallsWithZeroDurationFlag –** This parameter enables logging of CDRs for calls which were never connected, or which lasted less than one second. If this parameter is false, only calls with talk time greater or equal to 1 second and/or failed calls generate CDRs. If this parameter is true, all call will generate a CDR. In this case, going off hook and immediately back on hook will generate a CDR.
- **CallDiagnosticsEnabled –** This parameter enables and disables CMRs. If the parameter is false, CCM will not generate CMRs. If this parameter is true, CMRs will be generated for all supported devices. Currently, only some IP phone and some MGCP devices support the diagnostic data.

Enterprise Parameters

You can configure these parameters on the Enterprise Parameters Configuration page in the Cisco CallManager Administration. To access the Enterprise Parameters Configuration page, open Cisco CallManager Administration and select **System -> Enterprise Parameter**s.

- **CDRFlatFileInterval –** This is a parameter for CallManager that determines the number of seconds to write to a CDR file before opening another one. For example, if this CDRFlatFileInterval is 1 minute (default), CCM will write a minute worth of CDRs into each file. If the CDRFlatFileInterval is 60 minutes, then CCM will write an hours worth of CDRs into each file.
- **Cluster ID –** This parameter uniquely identifies the cluster. The Cluster ID is used both in the filename or the CDR/CMR flat files and in the CDR itself.

Cisco CallManager CDR Overview

This section provides a brief description of how CDRs are generated and managed in Cisco CallManager Release 5.0(1). The changes and new features are described in the following subsections.

CDR Management

The new CDR Management (CDRM) feature is a background application designed and implemented to fit into the new appliance model within the Cisco CallManager (CM) system on the Linux-based OS. The CDRM feature supports the following capabilities:

- 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
- allows third party applications to retrieve CDR/CMR files on demand through a SOAP interface
- monitors disk usage of CDR/CMR files on the CDR Repository node
- periodically deletes CDR/CMR files that have been successfully delivered

CDRM consists of the CDR Agent, CDR Repository Manager, and CDR onDemand Service.

CDR Agent

As part of the CDRM feature, a resident component on every CM node within a Cisco CallManager cluster is the CDR Agent. On a CM node where both CallManager and the CDR Agent are running, CallManager writes the CDRs into CDR flat files (CSV format) with a special control character ("_") prefixed to the filename by the call processing module to indicate that the file is not available for transfer. If this control character is not present, then the file is assumed to be available for transfer and is sent to the designated CDR Repository node. Upon successful transfer, the local copy of the file is deleted.

Reliability is given the highest priority for the CDRM feature. CDRs are very important financial data, so the goal of this feature is to guarantee that no CDR is lost. The CM 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 if a CDR file is available for transfer to the CDR Repository node. The advantage of having a short interval is that as soon as a file is available it can be delivered immediately to the CDR Repository node.

The CDR Agent uses a standard SFTP utility, sftp_connect.sh, to transfer CDR files from the CM nodes to the CDR Repository node. The utility requires a batch file as input and generates a log file indicating 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 the CallManager repeatedly tries to make new connections until successful. When CDR files are accumulated due to a lack of an SFTP connection, all leftover CDR files will be sent to the CDR Repository node immediately after connectivity is restored.

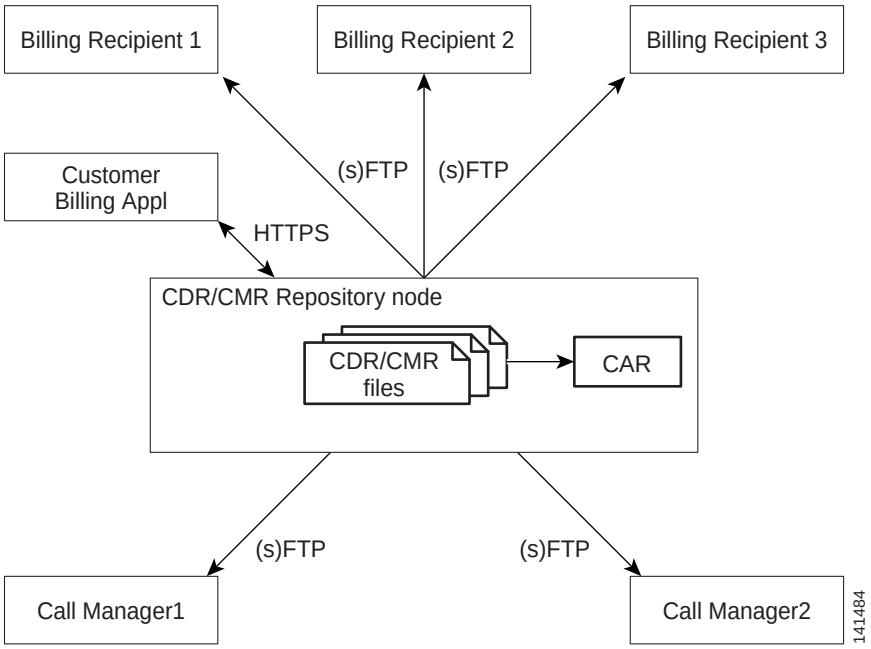
When the CDR Agent starts or restarts, it checks whether there are any CDR files remaining from the previous life cycle and sends them over to the CDR Repository node.

Should SFTP fail to transfer CDR files to the CDR Repository node, an alarm is raised.

CDR Repository Manager

Within a Cisco CallManager cluster, one instance of the CDR Repository Manager runs on the CDR Repository node. It manages CDR files received from CM nodes and periodically sends the files to the specified customer/third party billing servers via an (s)FTP connection, as illustrated in Figure 1.

Figure 1 CDR Management Feature



When the file arrives on the CDR Repository node, the CDR Repository Manager detects it. The file is archived in a directory 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, a soft link to the file is created in a directory 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 soft link in the destination directory is removed.

CDRM supports a BHCC (Busy Hour Call Completion) rate of 60K on a single CM node. Every CM node can generate one CDR file and one CMR file every minute for up to one hour. If there are 16 CM nodes within a Cisco CallManager cluster, then the minimum hard disk space requirement for the CDR Repository is 36GB. The maximum disk space used for storage of CDR files on the CDR Repository node is user-configurable through provisioning.

The File Manager component of the CDR Repository Manager is scheduled to run daily. When the File Manager runs, it deletes files with dates outside the configured preservation duration. It also checks if disk usage has exceeded the high water mark. If so, CDR files will be deleted until the low water mark is reached, starting with the oldest files. However, if any CDR file to be deleted has not been successfully sent to the specified billing server, it is left in the CDR Repository and a notification or alarm is raised. A flag file is created during the configured maintenance window, which denies access to the CDR files for third party billing applications and the CDR onDemand Service. The flag file is removed after the maintenance window has expired.

For detailed procedures for configuring the CDR Repository Manager and customer billing servers, see the “CDR Repository Manager Configuration” chapter in the *Cisco CDR Analysis and Reporting Tool Administration Guide*.

CDR onDemand Service

The CDR onDemand Service is a SOAP/HTTPS-based service that 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 one hour) and returns all lists fitting the time duration specified in the request.

The CDR onDemand Service can also handle requests for delivering a specific CDR file to a specified destination through (s)FTP. The CDR onDemand service can only be activated on the CDR Repository node as it has to access the CDR files in the repository. This service is prohibited during the maintenance window. For detailed information on the CDR onDemand Service, see the *Cisco CallManager Developers Guide for Release 5.0(1)*.

Types of Call Information Records

Cisco 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 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 CallManager does not perform any post processing on CDRs or CMRs.

Global Call Identifier

The Cisco CallManager allocates a global call identifier (GlobalCallID) each time that a Cisco 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.

Table 1 Sample CDR Table

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

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

Number Translations

The Cisco 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 shown in Table 2:

Table 2 Partition/Extension Numbers in CDRs

Phone Number	Description
callingPartyNumber	This party placed the call. For transferred calls, the transferred party becomes the calling party.
originalCalledPartyNumber	This number designates the originally called party, after any digit translations have occurred.
finalCalledPartyNumber	For forwarded calls, this number designates the last party to receive the call. For non-forwarded calls, this field shows the original called party.
lastRedirectDn	For forwarded calls, this field designates the last party to redirect the call. For non-forwarded calls, this field shows the last party to redirect (such as transfer and conference) the call.
callingPartyNumberPartition	This number identifies the partition name that is associated with the CallingPartyNumber field. This field uniquely identifies this number because the Cisco CallManager supports multiple Cisco 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 (continued)

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 CallManager supports multiple Cisco 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 CallManager supports multiple Cisco 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 CallManager supports multiple Cisco 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.

Timestamps

Timestamps within a CDR appear in universal coordinated time (UTC), which is the number of seconds since midnight on January 1, 1970. 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 Linux OS.

The CDR includes the UTC timestamps shown in Table 3:

Table 3 UTC Timestamps in CDRs

Field	Description
dateTimeOrigination	For outgoing calls, this field designates the time the device goes off hook. For incoming calls, this field designates the time the SETUP message is received.
dateTimeConnect	This field designates the time the devices connect and speech begins. This field shows a zero if the call never connects.
dateTimeDisconnect	This field designates the time the call disconnects. This field shows a zero if the call never connects.

Call Release Cause Codes

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

The "Call Release Cause Codes" section on page 85 lists the call release cause code values per ITU specification Q.850. For on-net call legs, the Cisco CallManager determines the call release cause code value. For off-net call legs, the far-end switch determines the call release cause code value.

IP Addresses

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

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

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

Step 1 Convert the database display (-1139627840) to a hex value.
The hex value equals 0xBC12A8C0.

Step 2 Reverse the order of the hex bytes, as shown below:
CO A8 12 BC

Step 3 Convert the four bytes from hex to decimal, as shown below:
192 168 18 188

Step 4 The IP address displays in the dotted decimal format:
192.168.18.188

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

Call Types

A successful call between two parties logs one CDR. Each CDR contains all fields, but some fields may not be 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, two CMRs are written: one for the originator and one for the destination of the call.

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

Successful On-Net Calls

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

The following table contains two examples:

- A—A 60-second call terminated by the caller
- B—A 60-second call cleared by the called party

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

Abandoned Calls

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

- All calls generate a CDR.
- If the call was 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.

The following table contains two examples:

- A—Extension 2001 going off hook then on hook (when the CdrLogCallsWithZeroDurationFlag is set at 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
A	2001	Accounts			16	0	0
B	2001	Accounts	2309		16	0	0

Calls with Busy or Bad Destinations

These calls will all be logged as a normal call with all relevant fields containing data. The Calling or Called Party Cause fields contain a cause code indicating why the call was not connected, and the Called Party IP and Date/Time Connect fields are blank. All unsuccessful calls are logged, even if zero duration calls are not being logged (CdrLogCallsWithZeroDurationFlag set at True or False, a duration of zero, and a DateTimeConnect value of zero).

The following table contains three examples:

- A—Call to PSTN number, party 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
B	2001	Accounts	902920262226	PSTN	0	17	0
C	2001	Accounts	902920100000	PSTN	0	1	0
D	2001	Accounts	902920262226	PSTN	0	38	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.

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

	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
B	02920262227	2001	ACNTS	6000	VMAIL	2309	MKTG	60	5	5

Pickup Calls

Cisco CallManager includes two pickup modes: Pickup and Auto Pickup. This section describes the CDRs for each mode.

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 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.

Call ID	Orig Cause	Calling Party	Dest Cause	Original Called Party	Final Called Party	Last Redirect Party	Orig Termination On BehalfOf	Dest Termination On BehalfOf	Last Redirect On BehalfOf	Last Redirect Reason	Join On 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 there is no need for the last answer softkey press. The system generates two CDRs for Auto Pickup, and these CDRs have the same Call ID.

The first CDR is generated 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 Termination On BehalfOf	Dest Termination 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 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 is the original placed call. The second CDR is the setup call (consultative/announcement) used to initiate the transfer. The third CDR is 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 this call resulted from a transfer. Therefore, all legs of the transfer can be tied back to a single call.

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

Example 1

A calls B, A transfers B to C. The three calls logged are:

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, then the call from A 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 are the same.

Example 2

A calls B, B transfers A to C. The three calls logged are:

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 are the same.

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 logged are:

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

Since the call was a blind transfer, then 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.

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
A	2001	ACNTS	101	3071111	PSTN	102	126	126	10	10	0	120
B	2001	ACNTS	103	2002	ACNTS	104	126	126	10	10	0	0
C	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 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.

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.
- C—Extension 2001 completes the transfer, dropping out of the call, 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
A	2001	ACNTS	101	3071111	PSTN	102	126	126	10	10	0	120
B	2001	ACNTS	103	2002	ACNTS	104	126	126	10	10	0	30
C	3071111	PSTN	102	2002	ACNTS	104	0	16	0	0	10	350

Conference Calls

Two 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 are connected 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 conference controller information is added 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 identifies 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.

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

The setup calls can be associated 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 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).

The following tables contain these examples:

- Call from 2001 to 2309.
- After 60 seconds, user 2001 presses the "conference" key on the Cisco IP Phone and dials the PSTN number "3071111."
- 3071111 answers and talks for 20 seconds; then, 2001 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 Conversation 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

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.

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
9728552001	GEN	1	6001	FIRE	1	16	0
2001	CMD	2	826001	FIRE	2	0	9
9728552001	GEN	1	826001	FIRE	1	0	16

Malicious Calls

When a call gets identified as a malicious call (button press), the local Cisco 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 calls that get terminated by this feature.

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

Note This table is continued on the following page.

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 Voicemail)

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

The following table contains an example CDR for this scenario:

Calling Party	Calling Partition	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

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 VideoCap_ Codec	Dest VideoCap_ Bandwidth	Dest VideoCap_ Resolution	DestVideo Transport Address_IP	DestVideo Transport Address_Port
100	320	2	288625580	49254

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 called person's name. 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

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.

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

In all cases, 2101 specifies the calling number.

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

Personal Assistant Interceptor Going to Media Port and Transferring the Call

This scenario acts similar to 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

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

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

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).

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

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

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

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.

Note 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.

Note 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

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	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.

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 identified above, but some fields may not be used. If a field is not used, it is blank if it is an ASCII string field, or “0” if it is a numeric field. When supplementary services are involved in a call, more CDRs may be written.

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

This section describes the records 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)

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 IP Phones generates a CDR at the end of the call.

A 60-second call terminated by the caller, notice since the calling party hangs up the orig_CauseValue is 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 since the called party hangs up the dest_CauseValue is 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

The logging of calls with zero duration is optional. Normally these records will not be 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 are blank. All calls that were not connected will have a **duration** of zero seconds. When a call is abandoned, the cause code is "0".

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 is blank and the **duration** is zero.

Examples of Abandoned Calls

- Extension 2001 going 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 be logged as a normal call with all relevant fields containing data. The Calling or Called Party Cause field contains a cause code indicating why the call was not connected, and the Called Party IP and Date/Time Connect fields is blank. All unsuccessful calls are logged, even if zero duration calls are not being logged.

Examples of Unsuccessful Calls

- Call to PSTN number, party 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, 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. Here are some of the important CDR fields for forwarded calls.

- The **originalCallPartyNumber** contains the number of the original called party.
- The **finalCalledPartyNumber** is the number that answered the call.
- The **lastRedirectDn** field is the number that performed the last redirect.
- The **origCalledPartyRedirectReason** is 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** is 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 redirect the call for the first redirect. For call forwarding, this field is 5 (Call Forward).
- The **lastRedirectRedirectOnBehalfOf** field identifies which feature redirect the call for the last redirect. For call forwarding, this field is 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 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 mail (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

There are two types of call pickup in Cisco CallManager: Pickup and Auto Pickup. The CDRs for both are slightly different for these two types of call pickup.

Auto Pickup

Auto Pickup is like call pickup with auto answer. The last answer softkey press is not needed. The call is automatically connected. Two CDRs are generated for Auto Pickup. These CDR will have the same Call ID.

- The first CDR is generated for the original call. This CDR will have the **origTerminationOnBehalfOf** and **destTerminationOnBehalfOf** fields equal to 16 (Pickup). This indicates 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 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 flavors for 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 are in the same pickup group. 2002 picks up the call ringing on 2001, the call is automatically connected between the PSTN caller and 2002. They talk for 2 minutes.

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

Legacy Call Pickup

Legacy Pickup calls are very similar to forwarded calls. Legacy Call Pickup uses the redirect call control primitive just like call forwarding. Here are some of the important CDR fields for Legacy Call Pickup calls.

- The **originalCallPartyNumber** contains the number of the original called party.
- The **finalCalledPartyNumber** is the number of the party that picked up the call.
- The **lastRedirectDn** field is the number was ringing when the call was picked up.
- The **origCalledPartyRedirectReason** is the reason the call was redirected the first time. For call pickup calls this field can contain (**Call Pickup = 5**).
- The **lastRedirectRedirectReason** is the reason the call was redirected the last time. For call call pickup this field can contain (**Call Pickup = 5**).
- The **origCalledPartyRedirectOnBehalfOf** field identifies which feature redirect the call for the first redirect. For call pickup, this field is (**Pickup = 16**).
- The **lastRedirectRedirectOnBehalfOf** field identifies which feature redirect the call for the last redirect. For call call pickup, this field is (**Pickup = 16**).
- Legacy Pickup Example
- **Legacy Pickup Example** - Call from the PSTN to extension 2001, 2001 and 2002 are in the same pickup group. 2002 picks up the call ringing on 2001, 2002 answers the call and the call is connected 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. There is one CDR for the original call, one of 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 are 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 are 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 are not an exhaustive set, and are intended to illustrate the records that would be generated under the stated circumstances. This is intended to 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** is the final transferred call where 2001 completes the transfer, drops out of the call, leaving a call between the PSTN and 2002.

FieldNames	Original Call CDR	Consultation Call CDR	Final Transferred CDR
globalCallID_callId	1	2	1
origLegCallIdentifier	101	103	102
destLegCallIdentifier	102	104	104
callingPartyNumber	2001	2001	3071111
originalCalledPartyNumber	3071111	2002	2002
finalCalledPartyNumber	3071111	2002	2002
lastRedirectDn	3071111	2002	2001
origCause_Value	393216	393216	16
dest_CauseValue	393216	393216	0
origTerminationOnBehalfOf	10	10	0
destTerminationOnBehalfOf	10	10	0
joinOnBehalfOf	0	0	10
duration	120	0	360

- **Consultation Transfer from the calling party** - 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 is completed. 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** is the final transferred call where 2001 completes the transfer, drops out of the call, leaving 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** is the final transferred call where 50001 completes the transfer, drops out of the call, leaving 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** is the final transferred call where 50000 completes the transfer, drops out of the call, leaving 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

Calls that are part of a conference have multiple records logged for them. The number of CDR generated is dependent on the number of parties in the conference. There is one CDR of 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 connected in the conference. Therefore, for a 3 party Ad-Hoc conference there would be six CDRs: one CDR for the original call, three CDRs for the parties connected to the conference, one CDR for each the setup call, and one CDR for the final two parties in the conference. The setup calls can be associated 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 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. All calls are shown “into” the conference bridge, regardless of the actually direction. But by examining the setup call CDRs, the original direction of each call can be determined.

The conference controller information can be found in the comment field of the CDR. The format of this information is:

Comment field = “ConfControllerDn=1000;ConfControllerDeviceName=SEP0003”

- The conference controller DN + conference controller device name uniquely identifies the conference controller. The device name is needed 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 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 will have the following fields information:

- The **finalCalledPartyNumber** field contains the conference bridge number “b0019901001”.
- The **origCalledPtyRedirectOnBehalfOf** field is set to (Conference = 4).
  - The **lastRedirectRedirectOnBehalfOf** field is set to (Conference = 4).
  - The **joinOnBehalfOf** field is set to (Conference = 4).
  - The **comment** field identifies the conference controller.
  - The **destConversationId** field is the same for all members in the conference. This field can be used 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:

- The **origCallTerminationOnBehalfOf** field is set to (Conference = 4).
- The **destCallTerminationOnBehalfOf** field is 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 leaving 2001 and 2309 in the conference. Since there are only two participants left in the conference, the conference features joins these two directly together and they talk for another 55 seconds.

Note Each of the conference call legs are shown as placing a call into the conference bridge. The call is 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
lastRedirectRedirectOnBehalfOf	0	0	4	4	4	4
joinOnBehalfOf	0	0	4	4	4	4
Conversation ID	0	1		1	1	0
duration	60	360		360	360	55

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

Call Park

Call Pickup will generate 2 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 is split. This generates a CDR for the original call. The **origTerminationOnBehalfOf** and **destTerminationOnBehalfOf** fields is 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 is joined with the parked call. Since the user picking up the is joined with the parked call, he/she is treated as the originator of the call and the parked user is treated as the destination. This means the **callingPartyNumber** of the call contains the directory number of the user picking up the call and the **originalCalledNumber** and **finalCalledNumber** contains the directoy number of the parked user. The **lastRedirectDn** contains the park code used to pickup the call. The **lastRedirectRedirectReason** is (Call Park Pickup = 8). The **lastRedirectRedirectOnBehalfOf** should also be (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 back to the called party. In this case, there are 2 CDRs generated. The first CDR is the same as Call Park Pickup scenario above, but the second CDR is slightly different. When the Call Pickup Reversion timer expires, the call is redirected back to the called party.

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

- When Call Park Reversion timer expires, the call is redirected back to the called party. The **origCalledPartyRedirectOnBehalfOf** and **lastRedirectRedirectOnBehalfOf** fields are (Call Park = 3). The **origCalledPartyRedirectReason** is (Call Park = 7) and the **lastRedirectRedirectReason** is (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 back to 50002 and 50002 answers.

FieldNames	Original Call that is parked	Reverted Call CDR
globalCallID_callId	2	2
origLegCallIdentifier	20863963	20863963
destLegCallIdentifier	20863964	20863967
callingPartyNumber	50003	50003
originalCalledPartyNumber	50002	50002
finalCalledPartyNumber	50002	50002
lastRedirectDn	50002	50002
origCause_Value	393216	0
dest_CauseValue	393216	16
origCalledPartyRedirectReason	0	7
lastRedirectRedirectReason	0	11
origCalledPartyRedirectOnBehalfOf	0	3
lastRedirectRedirectOnBehalfOf	0	3
origTerminationOnBehalfOf	3	3
destTerminationOnBehalfOf	3	12
joinOnBehalfOf	0	3
duration	7	60

Precedence Calls (MLPP)

With precedence calls everything is basically same for all calls (normal calls, forwarded calls, transferred calls, etc.). The difference is the precedence level fields is set in the CDR. 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)

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

- Call is 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 (CCM) flags the call. The “comment” is 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 Voicemail)

Idivert can be invoked in three different call states.

- The IDivert feature can be invoked while the incoming call is ringing. The CDR for the ringing case is very similar to call forwarding, but the **origCalledPartyRedirectOnBehalfOf** and the **lastRedirectRedirectOnBehalfOf** is (Immediate Divert = 14).
- The IDivert feature can be invoked while the call is connected or on hold. These scenarios generates 2 CDRs. Both CDRs will have the same **globalCallID_CallId** field. The first for the original connected and a second for the call redireted to voicemail. The first call will have the **origTerminationOnBehalfOf** and **destTerminationOnBehalfOf** field set to (Immediate Divert = 14).
- The call that is redirected to voicemail will have the **origCalledPartyRedirectOnBehalfOf** and the **lastRedirectRedirectOnBehalfOf** 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 voicemail (40000).

Note If the call is redirected by IDivert in the Alerting state only 1 CDR is 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 40001and 40001 answers the call. 40001 decides to divert the caller to voicemail and presses the IDivert softkey. 40003 is diverted to voicemail (40000).

  Since the call was connected before the redirect, 2 CDRs are generated.
  One for the original connected call and another for the call diverted to voicemail.

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

Barge

When a shared line uses the barge feature, the origCalledPartyNumber, finalCalledPartyNumber and lastRedirectDn are 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 are conferenced together. Then 40003 hangs up.

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

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 are 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 are conferenced together. Then 40001' (another shared line and phone) presses the Barge softkey. 40003 hangs up first.

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

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

cBarge

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

cBarge Examples

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

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

- **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 authorization description and level are written into the CDR. For security reasons, the actual authorization code will not be written to the CDR.

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

FAC Example

45000 calls 9728134987, the user is 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 CMC feature is invoked the client matter code is written into the CDR. The **clientMatterCode** field contains the client matter code entered by the caller.

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 reached during a call. For example, if the call is originally unsecured, then 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 identifies the DTMF method used for the call.

DTMF Call Examples

- **No Preference Example** - The DTMF method used during this call is No Preferece/Best Effort. This call is connected 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 used during this call is OOB Preferred. This call is 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 identifies the status of RSVP reservation for the call. The CCM 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 CCM RSVP reservation status for that particular stream: “2:5:2:5:2:5:2” (success:lost_bw:success:lost_bw:success:lost_bw:success).

RSVP Call Examples

- A call 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

- 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 is an example CDRs for a the redirection feature (3xx).

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

Redirection (3xx) Examples

- **Redirection Example** – Activate CFA on SIP phone 10010 (registered to CallManager) 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 is an example CDR for a Replaces call.

Replaces Examples

- **Invite with Replaces Example** – SIP phone 35010 calls SIP phone 35020, the transfer button is pressed on 35010 and a call is made to SCCP phone 3000, 3000 answers the call, then SIP phone 35010 completes the transfer. The final transferred call is between 35020 and 3000.

Note When the transfer is complete an Invite with Replaces is sent to 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** – SIP phone 35010 calls SCCP 3000, the transfer button is pressed on 35010 and a call is made to SCCP 3001, 3001 answers the call, then the SIP phone 35010 completes the transfer. The final transferred call is between 3000 and 3001.

Note When the transfer is complete a Refer with Replaces is sent to CallManager.

FieldNames	Original Call CDR	Consultation Call CDR	Final Transferred Call CDR
globalCallID_callId	5045245	5045246	5045245
origLegCallIdentifier	21822461	21822463	21822462
destLegCallIdentifier	21822462	21822464	21822464
callingPartyNumber	35010	35010	3000
originalCalledPartyNumber	3000	3001	3001
finalCalledPartyNumber	3000	3001	3001
lastRedirectDn	3000	3001	35010
origCause_Value	393216	393216	16
dest_CauseValue	393216	393216	0
origCalledPartyRedirectReason	0	0	130
lastRedirectRedirectReason	0	0	146
origCalledPartyRedirectOnBehalfOf	0	0	17
lastRedirectRedirectOnBehalfOf	0	0	18
origTerminationOnBehalfOf	17	18	12
destTerminationOnBehalfOf	17	18	17
joinOnBehalfOf	0	0	18
duration	25	4	25

Refer Calls

See the Replaces section for 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.

Table 4 CDR Field Descriptions

Field Name	Range of Values	Description
cdrRecordType	0, 1, 2	Defines the type of record. The following valid values apply: • 0—Start call detail record (not used) • 1—End call detail record (CDR) • 2—CMR Default - For CDRs, this field is always 1.
globalCallID_callManagerId	Positive Integer	Designates a unique Cisco CallManager identity. The Global Call ID consists of two fields: globalCallID_callId and globalCallID_callManagerId All records associated with a standard call have the same Global Call ID in them. Default - This field should always be populated.
globalCallID_callId	Positive Integer	Designates 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. A value is assigned for each call, successful or unsuccessful. When Cisco CallManager restarts, this values resets to 1. The Global Call ID consists of two fields: globalCallID_callId and globalCallID_callManagerId All records associated with a standard call have the same Global Call ID in them. Default - This field should always be populated.
origLegCallIdentifier	Positive Integer	Identifies the originating leg of a call. This value is unique within a cluster. If the leg of a call persists across several sub-calls, and consequently several CDRs (as during a call transfer), this value remains constant. Default - This field should always be populated.

Table 4 CDR Field Descriptions (continued)

Field Name	Range of Values	Description
dateTimeOrigination	Integer	Identifies the date and time when the user goes off hook or the date and time the H.323 Setup message is received for an incoming call. The time is stored as UTC. Default - This field should always be populated.
origNodeId	Positive Integer	Identifies the node within a cluster to which the originator of the call is registered at the time the call is made. Default - This field should always be populated.
origSpan	0, Positive integer	For calls originating at a gateway, this field identifies the port or span on the gateway where the call originated. For H.323 gateways, in which the span number is unknown, this field contains the call leg ID of the originator. For calls that did not originate at a gateway, the value is zero. Default - Populated based on these rules.
origIpAddr	Integer	Identifies the IP address of the device that originated the call signaling. For Cisco IP Phones, this field specifies the address of the Cisco IP 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 CallManager. The "IP Addresses" section on page 12 describes the IP address format. Default - Populated based on these rules.

Table 4 CDR Field Descriptions (continued)

Field Name	Range of Values	Description
callingPartyNumber	Text String	Specifies numeric string of up to 25 characters. For calls that originate at a Cisco 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 CallManager (such as translations at the gateway). For server calls, where Cisco CallManager originates a half call without a calling party, this field may be empty. CallingPartyNumber could contain a SIP URI. Default - Populated based on these rules.
callingPartyUnicodeLoginUserID	Unicode – UTF_8	Calling party's login user ID. The format of this field is UTF_8. Default - Empty string " ". If the user ID does not exist, this field is empty.
origCause_location	0 to 15	For clearing causes that are received over ISDN signaling links, specifies the Location field that is indicated in the ISDN release message. The "Call Release Cause Codes" section lists the valid values per Q.850. For clearing causes that are created internally by the Cisco CallManager, this value is zero. Default - 0.

Table 4 CDR Field Descriptions (continued)

Field Name	Range of Values	Description
origCause_value	0 to 129	For calls cleared by the originating party, this field reflects the reason for clearance. Cisco CallManager currently uses the Q.850 codes and some Cisco CallManager defined codes. These are listed in the "Call Release Cause Codes" section on page 85. Some of the non-standard cause codes have been changed in this release. For calls cleared by the terminating party, this field is zero. In addition to the standard values described in Q.850, when a call is placed on hold, the CDR terminates, and this field is set to 393216. This is 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 legs precedence level. • Precedence 0 = FLASH OVERRIDE/ EXECUTIVE OVERRIDE • Precedence 1 = FLASH • Precedence 2 = IMMEDIATE • Precedence 3 = PRIORITY • Precedence 4 = ROUTINE Default - 4.
origMediaTransportAddress_IP	0, Integer	Identifies the IP address of the device that originated the media for the call. For Cisco IP Phones, this field specifies the address of the Cisco IP 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 IP Phone. The "IP Addresses" section on page 12 describes the IP address format. Default - 0. If media is not established, this field is 0.

Table 4 CDR Field Descriptions (continued)

Field Name	Range of Values	Description
origMediaTransportAddress_Port	0, Integer	Identifies the IP port number associated with the OrigMediaTransportAddress_IP field. Default - 0. If media is not established, this field is 0.
origMediaCap_payloadCapability	0, Positive integer	Identifies the codec type that the originator used to transmit media. Cisco CallManager currently uses the following payload capability values (0, 1-16, 18-20, 25, 32, 33, 81-86). The "Codec Types" section on page 84 lists the valid values. Default - 0. If media is not established, this field is 0.
origMediaCap_maxFramesPerPacket	0, Positive integer	Identifies the number of milliseconds of data per packet sent by the originating party. This field is normally 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 is 0.
origMediaCap_g723BitRate	0	Deprecated since Cisco CallManager Release 3.3.4. This field will always be 0.
origVideoCap_Codec	0, 100 = H.261, 101 = H.263, 102 = Vieo	Identifies the codec type the originator used to transmit video (H.261, H.263, or Vieo.) Default - 0. If media is not established, this field is 0.
origVideoCap_Bandwidth	0, Positive Integer	Identifies the bandwidth measured in units of kbps. Default - 0. If media is not established, this field is 0.
origVideoCap_Resolution	0, 1 = SQCIF, 2 = QCIF, 3 = CIF, 4 = CIF4, 5 = CIF16	Identifies the Video resolution. Default - 0. If media is not established, this field is 0.

Table 4 CDR Field Descriptions (continued)

Field Name	Range of Values	Description
origVideoTransportAddress_IP	0, Integer	Identifies the IP address of the device that originates the call. Default - 0. If media is not established, this field is 0.
origVideoTransportAddress_Port	0, Positive Integer	Identifies the video RTP port associated with the origVideoTransportAddress_IP field. Default - 0. If media is not established, this field is 0.
origRSVPAudioStat	0 to 5	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	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	Identifies the terminating leg of a call. This value is unique within a cluster. If the leg of a call persists across several sub-calls and, consequently, several CDRs (as during a call transfer), this value remains constant. Default - 0. If the destination cannot be reached, this field is 0.
destNodeId	0, Positive Integer	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 is 0.
destSpan	0, Positive integer	For calls received at a gateway, this field identifies the port or span on the gateway where the call is received. For H.323 gateways, in which the span number is unknown, this field contains the call leg ID of the destination. For calls not terminating at a gateway, the value is zero. Default - 0. If the destination cannot be reached, this field is 0.
destIpAddr	Integer	Identifies the IP address of the device that terminated the call signaling. For Cisco IP Phones, this field specifies the address of the Cisco IP 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 CallManager. The "IP Addresses" section on page 12 describes the IP address format. Default - 0. If the destination cannot be reached, this field is 0.

Table 4 CDR Field Descriptions (continued)

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 on the Cisco CallManager, this number reflects the called number after the translations have been applied. Numeric string which could be either digits or a SIP URL. Default - empty string "". If destination cannot be reached, this field is 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 occurred, 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”). Numeric string which could be either digits or a SIP URL. Default - empty string ““. If destination cannot be reached, this field is empty.
finalCalledPartyUnicodeLoginUserID	Unicode – UTF_8	Final called party’s login user ID. The format of this field is UTF_8. Default - Empty string “ “. If the user ID does not exist, this field is empty.
destCause_location	0 to 15	For clearing causes that are received over ISDN signaling links, this is the Location field that the ISDN release message indicates. The “Call Release Cause Codes” section on page 85 lists the valid values per Q.850. Some of the non-standard cause codes have been changed in this release. For clearing causes that the Cisco CallManager created internally, this value equals zero. Default - 0. If the destination cannot be reached, this field is 0.

Table 4 CDR Field Descriptions (continued)

Field Name	Range of Values	Description
destCause_value	0 to 129	For calls that the destination party cleared, reflects the reason for the clearance. The "Call Release Cause Codes" section on page 85 lists the valid values per Q.850. Some of the non-standard cause codes have been changed in this release. For calls that the originating party cleared, this field is zero. In addition to the standard values described in Q.850, when a call is placed on hold, the CDR terminates, and this field is set to 393216. This is a proprietary value for this field. Default - 0. If the destination cannot be reached, this field is 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	Identifies the IP address of the device that terminated the media for the call. For Cisco IP Phones, this field designates the address of the Cisco IP 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 Cisco IP Phone. The "IP Addresses" section on page 12 describes the IP address format. Default - 0. If the destination cannot be reached, this field is 0.
destMediaTransportAddress_Port	0, Positive Integer	Identifies the IP port number associated with the DestMediaTransportAddress_IP field. Default - 0. If the destination cannot be reached, this field is 0.

Table 4 CDR Field Descriptions (continued)

Field Name	Range of Values	Description
destMediaCap_payloadCapability	0, Positive Integer	Identifies the codec type that the terminating party used to transmit media. Cisco CallManager currently uses the following payload capability values (0, 1-16, 18-20, 25, 32, 33, 81-86). The "Codec Types" section on page 84 lists the valid values. Default - 0. If the destination cannot be reached, this field is 0.
destMediaCap_maxFramesPerPacket	0, Positive Integer	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 is 0.
destMediaCap_g723BitRate	0	Depreciated since Cisco CallManager Release 3.3(4). Default - This field is always 0.
destVideoCap_Codec	0, 100 = H.261, 101 = H.263, 102 = Vieo	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 is 0.
destVideoCap_Bandwidth	0, Positive Integer	Identifies the bandwidth measured in units of kbps. Default - 0. If the destination cannot be reached, this field is 0.
destVideoCap_Resolution	0, 1 = SQCIF, 2 = QCIF, 3 = CIF, 4 = CIF4, 5 = CIF16	Identifies the video resolution. Default - 0. If the destination cannot be reached, this field is 0.
destVideoTransportAddress _IP	0, Integer	Identifies the IP address of the device that receives the call. Default - 0. If the destination cannot be reached, this field is 0.

Table 4 CDR Field Descriptions (continued)

Field Name	Range of Values	Description
destVideoTransportAddress_Port	0, Positive Integer	Identifies the video RTP port associated with the destVideoTransportAddress_IP field. Default - 0. If the destination cannot be reached, this field is 0.
destRSVPAudioStat	0 - 5	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	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”

Table 4 CDR Field Descriptions (continued)

Field Name	Range of Values	Description
dateTimeConnect	0, Integer	Identifies the date and time that the call connected. The time is stored as UTC. If the call is never answered, this value shows zero. Default - 0. If the call is never connected, this field is 0.
dateTimeDisconnect	0, Integer	Identifies the date and time when the call was cleared. This field gets set even if the call never connected. The time is stored as UTC. Default - 0. If the call is never connected, this field is 0.
lastRedirectDn	Text String	Specifies a numeric string of up to 25 characters. Numeric string could hold the 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 call was never redirected, this field is empty.
pkid	Text String	Identifies a text string that the database uses internally to uniquely identify each row. This text string provides no meaning to the call itself. Default - A unique ID should always populate this field.

Table 4 CDR Field Descriptions (continued)

Field Name	Range of Values	Description
originalCalledPartyNumberPartition	Text String	Identifies the partition name associated with the OriginalCalledPartyNumber field. This field uniquely identifies this number because the Cisco CallManager supports multiple Cisco 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 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 is empty.
callingPartyNumberPartition	Text String	Identifies the partition name associated with the CallingPartyNumber field. This field uniquely identifies this number because the Cisco CallManager supports multiple Cisco 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 is empty.
finalCalledPartyNumberPartition	Text String	Identifies the partition name associated with the FinalCalledPartyNumber field. This field uniquely identifies this number because the Cisco CallManager supports multiple Cisco 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 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 is empty.

Table 4 CDR Field Descriptions (continued)

Field Name	Range of Values	Description
lastRedirectDnPartition	Text String	Identifies the partition name associated with the LastRedirectDn field. This field uniquely identifies this number because the Cisco CallManager supports multiple Cisco 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 is empty.
duration	0, Positive integer	Identifies the difference between the Connect Time and Disconnect Time. This field specifies the time that the call remains connected, in seconds. This field remains zero if the call never connected or if it was connected for less than 1 second. Default - 0.
origDeviceName	Text String	Specifies text string that identifies the name of the originating device. Default - This field should always be populated.
destDeviceName	Text String	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 is empty.
origCallTerminationOnBehalfOf	0, Positive Integer	Specifies code that identifies why the originator was terminated. For example, if the originator of the call hangs up the phone, the OnBehalfOf code shows “12” for Device. If the call is terminated because of a transfer, the OnBehalfOf code shows “10” for Transfer. See the “OnBehalfof Codes” section on page 90 for a list of the codes. New OnBehalfOf codes have been added in this release. Default - 0.

Table 4 CDR Field Descriptions (continued)

Field Name	Range of Values	Description
destCallTerminationOnBehalfOf	0, Positive Integer	Specifies code that identifies why the destination was terminated. For example, if the originator of the call hangs up the phone, the OnBehalfOf code shows “12” for Device. If the call is terminated because of a transfer, the OnBehalfOf code shows “10” for Transfer. See the “OnBehalfof Codes” section on page 90 for a list of the codes. New OnBehalfOf codes have been added in this release. Default - 0.
origCalledPartyRedirectOnBehalfOf	0, Positive Integer	Specifies code that identifies the reason for redirection of the original called party. For example, if the original called party was redirected because of a conference, the OnBehalfOf code specifies “4.” See the “OnBehalfof Codes” section on page 90 for a list of the codes. New OnBehalfOf codes have been added in this release. Default - 0.
lastRedirectRedirectOnBehalfOf	0, Positive Integer	Specifies code that identifies the reason for redirection of the last redirected party. For example, if the last redirected party was redirected on behalf of a conference, the OnBehalfOf code specifies “4.” See the “OnBehalfof Codes” section on page 90 for a list of the codes. New OnBehalfOf codes have been added in this release. Default - 0.
origCalledPartyRedirectReason	0, Integer	Identifies the reason for a redirect of the original called party. See the “Redirect Reason Codes” section on page 89 for a complete list of the codes. New redirect reason values have been added in this release. Default - 0.

Table 4 CDR Field Descriptions (continued)

Field Name	Range of Values	Description
lastRedirectRedirectReason	0, Integer	Identifies the last redirect reason for redirection. See the "Redirect Reason Codes" section on page 89 for a complete list of the codes. New redirect reason values have been added in this release. Default - 0.
destConversationID	0, Integer	Specifies unique identifier that is used to identify the parties of a conference call. Default - 0.
globalCallId_ClusterId	Text String	Specifies a unique ID that identifies a cluster of Cisco CallManagers. This field is generated at installation, but is not used by Cisco 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	Specifies code that identifies the reason for a join. For example, if the join took place on behalf of a transfer, the OnBehalfOf code specifies "10." See the "OnBehalfof Codes" section on page 90 for a list of the codes. New OnBehalfOf codes have been added in this release. Default - 0.
Comment	TextString	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 "".

Table 4 CDR Field Descriptions (continued)

Field Name	Range of Values	Description
authCodeDescription	Text String	Description of the authorization code. For security purposes, the authorization code does not get written to the CDR; instead the authorization description and level are written. Default: Empty string "" or null.
authorizationLevel	0, integer	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	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 placed. Default: Empty string "" or null.
origDTMFMethod	0, Positive integer	DTMF method used by the originator side of the call. 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)

Table 4 CDR Field Descriptions (continued)

Field Name	Range of Values	Description
destDTMFMethod	0, Positive integer	DTMF method used by the destination side of the call. 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	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 1 for "Secured" even though different portions of the call had different status values. 0 - Non-secured 1 - Authenticated (not encypted) 2 - Secured (encrypted) Default – 0 (Non-secured)

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.

Table 5 CMR Field Descriptions

Field Name	Range of Values	Description
cdrRecordType	0, 1, or 2	Specifies the type of this specific record. The following valid values apply: • 0—Start call detail record (not used) • 1—End call detail record • 2—CMR Default - For CMRs, this field is always 2.
globalCallID_callManagerId	Positive Integer	Specifies a unique Cisco 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 in them. Default - This field should always be populated.
globalCallId_callId	Positive Integer	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 value assignment. This field makes up half the Global Call ID. The Global Call ID comprises the following two fields: • globalCallId_callId • globalCallID_callManagerID All records that are associated with a standard call have the same Global Call ID in them. Default - This field should always be populated.

Table 5 CMR Field Descriptions (continued)

Field Name	Range of Values	Description
nodeId	Positive Integer	Specifies the node within the Cisco CallManager cluster where this record was generated. Default - This field should always be populated.
callIdentifier	Positive Integer	Identifies the call leg to which this record pertains. Default - This field should always be populated.
directoryNumber	Integer	Specifies the directory number of the device from which these diagnostics were collected. Default - This field should always be populated.
dateTimeStamp	Integer	Represents the approximate time that the device went on hook. Cisco CallManager records the time when the phone responds to a request for diagnostic information. Default - This field should always be populated.
numberPacketsSent	Integer	Designates the total number of Routing Table Protocol (RTP) data packets that the device transmitted since starting transmission on this connection. The value remains zero if the connection was set in “receive only” mode. Default - 0.
numberOctetsSent	Integer	Specifies the total number of payload octets (that is, not including header or padding) that the device transmitted in RTP data packets since starting transmission on this connection. The value remains zero if the connection was set in “receive only” mode. Default - 0.
numberPacketsReceived	Integer	Specifies the total number of RTP data packets that the device received since starting reception on this connection. The count includes packets that were received from different sources if this is a multicast call. The value remains zero if the connection was set in “send only” mode. Default - 0.

Table 5 CMR Field Descriptions (continued)

Field Name	Range of Values	Description
numberOctetsReceived	Integer	Specifies the total number of payload octets (that is, not including header or padding) that the device received in RTP data packets since starting reception on this connection. The count includes packets that were received from different sources if this is a multicast call. The value remains zero if the connection was set in "send only" mode. Default - 0.
numberPacketsLost	Integer	Designates the total number of RTP data packets that have been lost since the beginning of reception. This number designates the number of packets that were expected, less the number of packets that were actually received, where the number of packets that were received includes any that are late or duplicates. Thus, packets that arrive late do not get counted as lost, and the loss may be negative if 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 was set in "send only" mode. For detailed information, see RFC 1889. Default - 0.
jitter	Integer	Provides an estimate of the statistical variance of the RTP data packet interarrival time, measured in milliseconds and expressed as an unsigned integer. The interarrival jitter J specifies the mean deviation (smoothed absolute value) of the difference D in packet spacing at the receiver, compared to the sender for a pair of packets. RFC 1889 contains detailed computation algorithms. The value remains zero if the connection was set in "send only" mode. Default - 0.

Table 5 CMR Field Descriptions (continued)

Field Name	Range of Values	Description
latency	Integer	Designates value that is an estimate of the network latency, expressed in milliseconds. This value represents the average value of the difference between the NTP timestamp that the RTP Control Protocol (RTCP) messages indicates and the NTP timestamp of the receivers, measured when these messages are received. Cisco 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	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	Identifies the partition of the directory number. Default - Empty string, “”. This field may be empty if no partition exists.
deviceName	Text String	Identifies the name of the device. Default - Empty string “”. This field may be empty if there is no device name.
globalCallId_ClusterId	Text String	Designates unique ID that identifies a cluster of Cisco CallManagers. This field is generated during installation, but is not used by Cisco CallManager: globalCallId_ClusterId + globalCallId_callManagerId + globalCallId_callId. Default - This field will always be populated.

Table 5 CMR Field Descriptions (continued)

Field Name	Range of Values	Description
varVQMetrics	Text String	This field contains a variable number of voice quality metrics. In this release, the kfactor and concealed seconds VQ (voice quality) metrics are added. This field is a string of voice quality metrics separated by a semicolon. The format of the string is: fieldName=value;fieldName=value/precision This is an example of voice quality data, but the names may be different. "MLQKav=14396/4096;MLQK=13926/4096;MLQKmn=12288/4096; MLQKmx=14396/4096;CRRav=0/65536;CRR=1024/65536; CRRmn=0/65536;CRRmx=3277/65536;CS=15;SCS=2" **Note** See Table 6 "K-Factor Data Stored in Cisco CallManager CMRs" for a complete list of K-Factor data.

K-Factor Data in CMRs

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

MOS is a term that 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 is a number inversely proportional to frame loss density. Clarity decreases as more frames are lost or discarded at the receiving end. The loss or discarding of these frames is termed "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 is 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 is trained or conditioned by speech samples from numerous speech databases, where each training sentence or network condition associated with a P.862.1 value has a duration of eight seconds. Hence, for more accurate scores, k-factor estimates are generated for every eight seconds of active speech.

K-factor and other MOS estimators are considered 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 are primary statistics because they alert the network operator before network impairment has an audible impact, or is visible through MOS.

Table 6 K-Factor Data Stored in Cisco CallManager CMRs

Field Name	Phone Display Name	D&I User Interface Text and Description
CCR	Cum Conceal Ratio	Cumulative Conceal Ratio. Is the cumulative ratio of concealment time over speech time observed after starting a call.
ICR	Interval Conceal Ratio	Interval Conceal Ratio. Is an interval-based average concealment rate, and is the ratio of concealment time over speech time for the last three seconds of active speech.
ICRmx	Max Conceal Ratio	Interval Conceal Ratio Max. Is the maximum concealment ratio observed during the call.
CS	Conceal Secs	Conceal Secs. Is the duration of time during which some concealment is observed during a call.
SCS	Severely Conceal Secs	Severely Conceal Secs. Is the duration of time during which a significant amount of concealment is observed. If the concealment observed is usually greater than fifty milliseconds or approximately five percent, the speech is probably not very audible.
MLQK	MOS LQK	MOS Listening Quality K-factor. An estimate of the MOS score of the last eight seconds of speech on the reception signal path.
MLQKmn	Min MOS LQK	MOS Listening Quality K-factor Min. The minimum score observed since the beginning of a call, and represents the worst sounding eight second interval.
MLQKmx	Max MOS LQK	MOS Listening Quality K-factor Max. The maximum score observed since the beginning of a call, and represents the best sounding eight second interval.
MLQKav	Avg MOS LQK	MOS Listening Quality K-factor Avg8. The running average of scores observed since the beginning of a call.

Codec Types

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

Table 7 Codec Types

Value	Description
1	NonStandard
2	G711Alaw 64k
3	G711Alaw 56k
4	G711mu-law 64k
5	G711mu-law 56k
6	G722 64k
7	G722 56k
8	G722 48k
9	G7231
10	G728
11	G729
12	G729AnnexA
13	Is11172AudioCap
14	Is13818AudioCap
15	G.729AnnexB
16	G.729 Annex AwAnnexB
18	GSM Full Rate
19	GSM Half Rate
20	GSM Enhanced Full Rate
25	Wideband 256K
32	Data 64k
33	Data 56k
40	G7221 32K
41	G7221 24K
80	GSM
81	ActiveVoice
82	G726_32K
83	G726_24K
84	G726_16K
100	H261
101	H263
102	Vieo

Table 7 Codec Types (continued)

Value	Description
103	H264
106	H224

Call Release Cause Codes

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

Table 8 Call Release Cause Codes

Code	Description
0	No error
1	Unallocated (unassigned) number
2	No route to specified transit network (national use)
3	No route to destination
4	Send special information tone
5	Misdialed trunk prefix (national use)
6	Channel unacceptable
7	Call awarded and being delivered in an established channel
8	Preemption
9	Preemption—circuit reserved for reuse
16	Normal call clearing
17	User busy
18	No user responding
19	No answer from user (user alerted)
20	Subscriber absent
21	Call rejected
22	Number changed
26	Nonselected 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

Table 8 Call Release Cause Codes (continued)

Code	Description
41	Temporary failure
42	Switching equipment congestion
43	Access information discarded
44	Requested circuit/channel not available
46	Precedence call blocked
47	Resource unavailable, unspecified
49	Quality of Service not available
50	Requested facility not subscribed
53	Service operation violated
54	Incoming calls barred
55	Incoming calls barred within Closed User Group (CUG)
57	Bearer capability not authorized
58	Bearer capability not presently available
62	Inconsistency in designated outgoing access information and subscriber class
63	Service or option not available, unspecified
65	Bearer capability not implemented
66	Channel type not implemented
69	Requested facility not implemented
70	Only restricted digital information bearer capability available (national use)
79	Service or option not implemented, unspecified
81	Invalid call reference value
82	Identified channel does not exist.
83	A suspended call exists, but this call identity does not.
84	Call identity in use
85	No call suspended
86	Call having the requested call identity has been cleared.
87	User not member of (CUG)
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

Table 8 Call Release Cause Codes (continued)

Code	Description
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 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 is terminated during a transfer operation because it was split off and terminated (was not part of the final transferred call). This can help determine which calls were terminated as part of a feature operation.
458752 0x70000	Drop any party/drop last party (was 128)

L

Table 10 SIP Call Release Cause Codes

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

Table 10 SIP Call Release Cause Codes (continued)

Code	Description
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
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

Redirect Reason Codes

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

Table 11 Redirect Reason Codes

Q.931 Standard Redirect Reason Codes	
Value	**Description**
0	Unknown
1	Call Forward Busy
2	Call Forward No Answer
4	Call Transfer
5	Call Pickup
7	Call Park
8	Call Park Pickup
9	CPE Out of Order
10	Call Forward
11	Call Park Reversion
15	Call Forward All
Non Standard 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)
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)

OnBehalfof Codes

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

Table 12 OnBehalfof Codes

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

Related Documentation

The following documents contain additional information related to CDRs:

- *Cisco CallManager CDR Analysis and Reporting Tool Administration Guide*
- *Cisco CallManager Serviceability Administration*
- *Cisco CallManager Serviceability System Guide*
- *Cisco CallManager System Guide*

Obtaining Documentation

Cisco documentation and additional literature are available on Cisco.com. Cisco also provides several ways to obtain technical assistance and other technical resources. These sections explain how to obtain technical information from Cisco Systems.

Cisco.com

You can access the Cisco website at this URL:

http://www.cisco.com

You can access the most current Cisco documentation at this URL:

http://www.cisco.com/univercd/home/home.htm

You can access international Cisco websites at this URL:

http://www.cisco.com/public/countries_languages.shtml

Documentation DVD

Cisco documentation and additional literature are available in a Documentation DVD package, which may have shipped with your product. The Documentation DVD is updated regularly and may be more current than printed documentation. The Documentation DVD package is available as a single unit.

Registered Cisco.com users (Cisco direct customers) can order a Cisco Documentation DVD (product number DOC-DOCDVD=) from the Ordering tool or Cisco Marketplace.

Cisco Ordering tool:

http://www.cisco.com/en/US/partner/ordering/

Cisco Marketplace:

http://www.cisco.com/go/marketplace/

Ordering Documentation

You can find instructions for ordering documentation at this URL:

http://www.cisco.com/univercd/cc/td/doc/es_inpck/pdi.htm

You can order Cisco documentation in these ways:

- Registered Cisco.com users (Cisco direct customers) can order Cisco product documentation from the Ordering tool:

  http://www.cisco.com/en/US/partner/ordering/

- Nonregistered Cisco.com users can order documentation through a local account representative by calling Cisco Systems Corporate Headquarters (California, USA) at 408 526-7208 or, elsewhere in North America, by calling 1 800 553-NETS (6387).

Documentation Feedback

You can send comments about technical documentation to bug-doc@cisco.com.

You can submit comments by using the response card (if present) behind the front cover of your document or by writing to the following address:

Cisco Systems
Attn: Customer Document Ordering
170 West Tasman Drive
San Jose, CA 95134-9883

We appreciate your comments.

Cisco Product Security Overview

Cisco provides a free online Security Vulnerability Policy portal at this URL:

http://www.cisco.com/en/US/products/products_security_vulnerability_policy.html

From this site, you can perform these tasks:

- Report security vulnerabilities in Cisco products.
- Obtain assistance with security incidents that involve Cisco products.
- Register to receive security information from Cisco.

A current list of security advisories and notices for Cisco products is available at this URL:

http://www.cisco.com/go/psirt

If you prefer to see advisories and notices as they are updated in real time, you can access a Product Security Incident Response Team Really Simple Syndication (PSIRT RSS) feed from this URL:

http://www.cisco.com/en/US/products/products_psirt_rss_feed.html

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

A summary of U.S. laws governing Cisco cryptographic products may be found at: http://www.cisco.com/wwl/export/crypto/tool/stqrg.html.

If you require further assistance please contact us by sending email to export@cisco.com.

Reporting Security Problems in Cisco Products

Cisco is committed to delivering secure products. We test our products internally before we release them, and we strive to correct all vulnerabilities quickly. If you think that you might have identified a vulnerability in a Cisco product, contact PSIRT:

- Emergencies—security-alert@cisco.com
- Non emergencies—psirt@cisco.com

Tip We encourage you to use Pretty Good Privacy (PGP) or a compatible product to encrypt any sensitive information that you send to Cisco. PSIRT can work from encrypted information that is compatible with PGP versions 2.*x* through 8.*x*.

Never use a revoked or an expired encryption key. The correct public key to use in your correspondence with PSIRT is the one that has the most recent creation date in this public key server list:

http://pgp.mit.edu:11371/pks/lookup?search=psirt%40cisco.com&op=index&exact=on

In an emergency, you can also reach PSIRT by telephone: 1-877-228-7302 or 1-408-525-6532.

Obtaining Technical Assistance

For all customers, partners, resellers, and distributors who hold valid Cisco service contracts, Cisco Technical Support provides 24-hour-a-day, award-winning technical assistance. The Cisco Technical Support Website on Cisco.com features extensive online support resources. In addition, Cisco Technical Assistance Center (TAC) engineers provide telephone support. If you do not hold a valid Cisco service contract, contact your reseller.

Cisco Technical Support Website

The Cisco Technical Support Website provides online documents and tools for troubleshooting and resolving technical issues with Cisco products and technologies. The website is available 24 hours a day, 365 days a year, at this URL:

http://www.cisco.com/techsupport

Access to all tools on the Cisco Technical Support Website requires a Cisco.com user ID and password. If you have a valid service contract but do not have a user ID or password, you can register at this URL:

http://tools.cisco.com/RPF/register/register.do

Note Use the Cisco Product Identification (CPI) tool to locate your product serial number before submitting a web or phone request for service. You can access the CPI tool from the Cisco Technical Support Website by clicking the **Tools & Resources** link under Documentation & Tools. Choose **Cisco Product Identification Tool** from the Alphabetical Index drop-down list, or click the **Cisco Product Identification Tool** link under Alerts & RMAs. The CPI tool offers three search options: by product ID or model name; by tree view; or for certain products, by copying and pasting **show** command output. Search results show an illustration of your product with the serial number label location highlighted. Locate the serial number label on your product and record the information before placing a service call.

Submitting a Service Request

Using the online TAC Service Request Tool is the fastest way to open S3 and S4 service requests. (S3 and S4 service requests are those in which your network is minimally impaired or for which you require product information.) After you describe your situation, the TAC Service Request Tool provides recommended solutions. If your issue is not resolved using the recommended resources, your service request is assigned to a Cisco TAC engineer. The TAC Service Request Tool is located at this URL:

http://www.cisco.com/techsupport/servicerequest

For S1 or S2 service requests or if you do not have Internet access, contact the Cisco TAC by telephone. (S1 or S2 service requests are those in which your production network is down or severely degraded.) Cisco TAC engineers are assigned immediately to S1 and S2 service requests to help keep your business operations running smoothly.

To open a service request by telephone, use one of the following numbers:

Asia-Pacific: +61 2 8446 7411 (Australia: 1 800 805 227)
EMEA: +32 2 704 55 55
USA: 1 800 553-2447

For a complete list of Cisco TAC contacts, go to this URL:

http://www.cisco.com/techsupport/contacts

Definitions of Service Request Severity

To ensure that all service requests are reported in a standard format, Cisco has established severity definitions.

Severity 1 (S1)—Your network is “down,” or there is a critical impact to your business operations. You and Cisco will commit all necessary resources around the clock to resolve the situation.

Severity 2 (S2)—Operation of an existing network is severely degraded, or significant aspects of your business operation are negatively affected by inadequate performance of Cisco products. You and Cisco will commit full-time resources during normal business hours to resolve the situation.

Severity 3 (S3)—Operational performance of your network is impaired, but most business operations remain functional. You and Cisco will commit resources during normal business hours to restore service to satisfactory levels.

Severity 4 (S4)—You require information or assistance with Cisco product capabilities, installation, or configuration. There is little or no effect on your business operations.

Obtaining Additional Publications and Information

Information about Cisco products, technologies, and network solutions is available from various online and printed sources.

- Cisco Marketplace provides a variety of Cisco books, reference guides, and logo merchandise. Visit Cisco Marketplace, the company store, at this URL:

  http://www.cisco.com/go/marketplace/

- *Cisco Press* publishes a wide range of general networking, training and certification titles. Both new and experienced users will benefit from these publications. For current Cisco Press titles and other information, go to Cisco Press at this URL:

  http://www.ciscopress.com

- *Packet* magazine is the Cisco Systems technical user magazine for maximizing Internet and networking investments. Each quarter, Packet delivers coverage of the latest industry trends, technology breakthroughs, and Cisco products and solutions, as well as network deployment and troubleshooting tips, configuration examples, customer case studies, certification and training information, and links to scores of in-depth online resources. You can access Packet magazine at this URL:

  http://www.cisco.com/packet

- *iQ Magazine* is the quarterly publication from Cisco Systems designed to help growing companies learn how they can use technology to increase revenue, streamline their business, and expand services. The publication identifies the challenges facing these companies and the technologies to help solve them, using real-world case studies and business strategies to help readers make sound technology investment decisions. You can access iQ Magazine at this URL:

  http://www.cisco.com/go/iqmagazine

- *Internet Protocol Journal* is a quarterly journal published by Cisco Systems for engineering professionals involved in designing, developing, and operating public and private internets and intranets. You can access the Internet Protocol Journal at this URL:

  http://www.cisco.com/ipj

- World-class networking training is available from Cisco. You can view current offerings at this URL:

  http://www.cisco.com/en/US/learning/index.html

This document is to be used in conjunction with the documents listed in the "Related Documentation" section.