



Call Details Records Overview

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

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

- [CDR Management, on page 1](#)
- [CDR Database Backup and Recovery, on page 3](#)
- [Record Processing, on page 3](#)
- [Call Information Record Types, on page 5](#)

CDR Management

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

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

mark. The post-processing applications can later retrieve the buffered historical data to re-get any lost, corrupted, or missing data. The CDRM feature, which is not aware of the flat file format, does not manipulate the file contents.



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

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

CDR Agent

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

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



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

CDR Repository Manager

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

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

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

empty files and sends the file to the destination with the specified method. If the delivery is successful, the system removes the empty file in the destination directory.

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

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

CDR onDemand Service

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

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

For Cisco Unified Communications Manager Release 12.x and later releases, CDR onDemand Service is not enabled by default. If you want to enable the CDR onDemand service, the service should be activated manually. Execute the following command at the root level to activate the CDR onDemand service:

```
/usr/local/cm/bin/soapervicecontrol2.shCDRonDemandServiceCDRonDemanddeploy8443.
```

CDR Database Backup and Recovery

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

For more information, see *Administration Guide for Cisco Unified Communications Manager*.

Record Processing

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

When Unified Communications Manager places or receives a call, the system generates a CDR record when the call terminates. The system writes the CDR to a flat file (text file). Inside the Unified Communications

Manager, the Call Control process generates CDR records. The system writes records when significant changes occur to a given call, such as ending the call, transferring the call, redirecting the call, splitting the call, joining a call, and so forth.

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

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

Filename Format

The following example shows the full format of the filename:

tag_clusterId_nodeId_datetime_seqNumber

- **tag**—Identifies the type of file, either CDR or CMR.
- **clusterId**—Identifies the cluster or server where the Unified Communications Manager database resides.
- **nodeId**—Identifies the node
- **datetime**—UTC time in yyyyymmddhhmm format
- **seqnumber**—Sequence number

Two examples of the filenames follow:

```
cdr_Cluster1_01_200404021658_1
cmr_Cluster1_02_200404061011_6125
```

Flat File Format

The CDR and CMR flat files have the following format:

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

The following example shows a flat file:

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



Note If the value of the CDR Log Calls With Zero Duration Flag parameter is True, the system writes all calls to a flat file. For more information, see *Configuring CDR Service Parameters* section in this book.

Call Information Record Types

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

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

CDRs relate to the CMRs via two globalCallID columns:

- globalCallID_callManagerId
- globalCallId_callId

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

Global Call Identifier

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

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



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

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

If the GlobalCallID_callId field is missing from the CDR record, CAR generates an error for that particular record. For more information, see *CDR Error Reports* in this guide.



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

Number Translations

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

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



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

Partitions and Numbers

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

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

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

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

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

Table 1: Partition/Extension Numbers in CDRs

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

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

Timestamps

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

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

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

Table 2: UTC Timestamps in CDRs

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

Call Clearing Causes

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

[Call Termination Cause Codes](#) lists the call clearing cause code values per ITU specification Q.850. For On Net call legs, the Unified Communications Manager determines the cause code value. For Off Net call legs, the far-end switch determines the cause code value.

Convert Signed Decimal Value to IP Address

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



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

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

Procedure

- Step 1** Convert the database display (-1139627840) to a hex value.
The hex value equals 0xBC12A8C0.
- Step 2** Reverse the order of the hex bytes, as shown below:
CO A8 12 BC
- Step 3** Convert the four bytes from hex to decimal, as shown below:
192 168 18 188
- Step 4** The IP address displays in the dotted decimal format:
192.168.18.188
-

What to do next

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

