SipXecs 4.2

Call Detail Recording

Developer Guide

Author: Raymond Dans
Last modified: 20. Aug. 2009 - version 1.1
First created: 25. May. 2009

Table of Contents

Call Detail Recording
Developer Guide

1. Introduction

The SipXecs Call Detail Records (CDRs) subsystem generates details of calls and stores them in a local database . This information is used for many purposes among which includes:

- Call Accounting - Billing
- System Utilization
- Resource Utilization

SipXecs provides a System Administrator with the ability to view and report on records in the system, however, in some cases it is desirable to have more control over the use of this data.

This document provides information for developers wishing to access and interpret the raw CDR data stored in the system.

It is assumed that the developer has sufficient knowledge of the Structured Query Language (SQL) in order to retrieve the necessary records and information in the database, SIP terminology to understand the data and basic SQL database table structuring.

2. Call Detail Records

CDRs are stored in a Postgres database named “SIPXCDR” and can be retrieved or viewed in either its raw (full) format or in a restricted view.

The raw format of the data (table cdrs) contains all of the stored information for each call in the system. A publicly available schema for the table “cdrs” can be found on the master system under */etc/sipxpbx/cdr/schema.sql*.

The restricted view of the data (view view_cdrs) holds commonly used information. The advantage to using the view, should it provide all the necessary information for your needs, is that it protects you against possible future changes to the database schema.

2.1 Table cdrs

The records in the “cdrs” table contain the following information as used in SCS release 3.0:

Field Name	Type	Description
Id	Serial8	Primary Key – uniquely identifies the record.
call_id	Text	SIP Call Identifier – unique per call.
from_tag	Text	SIP tag provided by call

Field Name	Type	Description
		originator.
to_tag	Text	SIP tag provided by call recipient.
caller_aor	Text	Call originator Address of Record (AOR)
callee_aor	Text	Call recipient Address of Record(AOR).
start_time	Timestamp	Database timestamp in GMT of call start.
connect_time	Timestamp	Database timestamp in GMT of call connect.
end_time	Timestamp	Database timestamp in GMT of call end.
termination	Char(1)	Call status. Indicates success or failure of call at end.
failure_status	Int2	SIP failure code for call failure.
failure_reason	Text	SIP text reason for call failure. Empty if call successful.
call_direction	Char(1)	Not currently used.

2.2 View view_cdrs

Field Name	Type	Description
Id	Serial8	Primary Key – uniquely identifies the record.
caller_aor	Text	Call originator Address of Record (AOR)
callee_aor	Text	Call recipient Address of Record(AOR).
start_time	Timestamp	Database timestamp in GMT of call start.
connect_time	Timestamp	Database timestamp in GMT of call connect.
end_time	Timestamp	Database timestamp in UTC (GMT) of call end.
duration	Timestamp	(end_time - connect_time)
termination	Char(1)	Call status. Indicates success or failure of call at end.
failure_status	Int2	SIP failure code for call failure.
failure_reason	Text	SIP text reason for call failure.

Field Name	Type	Description
		Empty if call successful.
call_direction	Char(1)	Not currently used.
reference	Text	Comma separated list of related call id(s).
caller_contact	Text	Contact URI for Caller
callee_contact	Text	Contact URI for call recipient
caller_internal	Boolean	Indication of whether the caller is internal or not (true = internal)
callee_route	Text	Comma separated list of call tags to help identify type of call.

2.3 CDR Field Information

To properly interpret the data for a call record, it is necessary to understand the use of each field and any relationship to other fields in the record. The following provides a more complete description of each field and its interpretation.

2.3.1 Id

Record Id automatically assigned by the database system when a record is inserted. Unique across all records and provides no real value when processing by an application.

2.3.2 call_id

A globally unique identifier for this call. It is generated and supplied by the caller typically by combining a random string and the host name or IP address.

2.3.3 from_tag

A globally unique and cryptographically random string generated by the caller. Used in combination with the call_id and to_tag to uniquely identify a dialog between the 2 endpoints of the call. This field is typically not used by applications when processing CDR records.

2.3.4 to_tag

A globally unique and cryptographically random string generated by the callee. Used in combination with the call_id and from_tag to uniquely identify a dialog between the 2 endpoints of the call. This field is typically not used by applications when processing CDR records.

2.3.5 caller_aor

Address of Record (AOR) of the initiator of the call. It is a SIP Uniform Resource Identifier (URI) that points to a domain with a location service. Frequently thought of as the public address of the user (e.g. sip:19941@abccompany.com).

On an outgoing call, this field is typically used for determining who to charge the call against.

On an incoming call, this field is typically used for determining who called into the system. If the incoming call is from a gateway, this field usually contains the Caller Id of

2.3.6 callee_aor

Address of Record (AOR) of the recipient of the call. It is a SIP Uniform Resource Identifier (URI) that points to a domain with a location service. Frequently thought of as the public address of the user (e.g. sip:2154@farendcompany.com)

2.3.7 start_time

Database timestamp in UTC(GMT) time without a time zone indicating the start time of the call. Value is of the format:

<4 digit year>-<2 digit month>-<2 digit day> <2 digit hour in a 24 hour clock>:<2 digit minutes>:<2 digit seconds>.<3 digit micro seconds>

e.g. 2009-05-19 17:53:47.039

It is up to the application to modify this time according to the time zone for the system.

Note: This value only indicates the time when a call was initiated and not when it was actually answered.

2.3.8 connect_time

Database timestamp in UTC(GMT) time without a time zone indicating the time the call connected between the 2 parties. Value is of the format:

<4 digit year>-<2 digit month>-<2 digit day> <2 digit hour in a 24 hour clock>:<2 digit minutes>:<2 digit seconds>.<3 digit micro seconds>

e.g. 2009-05-19 17:53:53.219

It is up to the application to modify this time according to the time zone for the system.

Note: The difference between the connect_time and start_time can be thought of as the amount of time the call was “ringing”.

2.3.9 end_time

Database timestamp in UTC(GMT) time without a time zone indicating the time the call ended between the 2 parties. Value is of the format:

<4 digit year>-<2 digit month>-<2 digit day> <2 digit hour in a 24 hour clock>:<2 digit minutes>:<2 digit seconds>.<3 digit micro seconds>

e.g. 2009-05-19 17:56:53.219

It is up to the application to modify this time according to the time zone for the system.

Note: Call duration can be calculated by calculating the difference between end_time and connect_time.

2.3.10 Termination

A single character code indicating the success or failure of the call.

'C' = Call Completed successfully.

'F' = Call Failed. Application must examine the failure_status and failure_reason for additional information indicating a reasoning for the failure.

'T' = Call Transferred.

'U' = Call Completion status is Unknown. This status is used for calls whose tracking we have lost. In those cases, we tag the status as unknown and close off the record.

Note: Unanswered or Abandoned calls are flagged as failed calls.

2.3.11 Failure_status

In a call failure scenario (termination = 'F'), this field may contain the SIP failure code. These failure codes are 3-digit integers in the 4xx,5xx and 6xx range (where xx is 00 to 99).

2.3.12 Failure_reason

In a call failure scenario (termination= 'F'), this text field contains a short english description of the failure_code.

e.g. failure code 408 = “Request Timeout”

2.3.13 call_direction

Not currently implemented or populated.

2.3.14 reference

For SIP, every call generates a new call id. In scenarios such as a Transfer (ie. Blind, Consultative, ..) multiple SIP calls (and subsequently call id(s)) occur, thereby generating multiple CDR records.

In order to be able to link calls into a logical group to represent a single call in the traditional sense, the reference field can be used to trace in **reverse order** the path of the call in the system.

The reference field contains a comma separated list of SIP call id(s) with a parameter (rel) indicating the type of relationship to the current record's call id. Two possible values for the relationship field can occur (refer and xfer). The “xfer” value occurs in a consultative transfer and refers to the consult call. The “refer” value indicates the call that was transferred.

2.3.15 caller_contact

Contact for the call originator. It is a SIP Uniform Resource Identifier (URI) that provides contact information to a specific instance of the user agent.

In SIP, a user may register from many different locations. The contact can provide in this situation more detailed information as to exactly which device for that user initiated the call.

2.3.16 callee_contact

Contact for the call recipient. It is a SIP Uniform Resource Identifier (URI) that provides contact information to a specific instance of the user agent of the user which accepted the call.

In SIP, a user may register from many different locations. The contact can provide in this situation more detailed information as to exactly which device for that user accepted the call or in some cases where a call is forked to multiple possible recipients, the actual recipient which accepted the call.

2.3.17 caller_internal

A boolean value (true,false) indicating whether the call originator is an internally recognized user (true) or not (false). This field can be used in combination with the callee_route to help categorize calls as Incoming, Outgoing, Tandem and Internal.

2.3.18 callee_route

The callee_route is a string containing a comma separated list of “calltags” that provides information as to the category of the destination (and how it was obtained). The majority of the tags are based on the dial plan rules currently enabled. The current list of possible call tags are:

Tag	*Description*
INT	Internal
LOCL	Local
VM	Voice Mail
VMR	Voice Mail Redirect
LD	Long Distance
TF	Toll Free
STS	Site to Site
CUST	Custom dialing rule
EMERG	Emergency
PAGE	Page
AA	Auto Attendant
AL	Alias
REST	Restricted

2.4 Special Users

A number of special users are defined in the SipXecs system and may appear in the CDR record under either the caller_aor or callee_aor fields. The current list of special users and their meaning are:

User Id	Role/Description
~~id~park	Park Server
~~id~media	Media Server
~~id~acd	Automatic Call Distribution (ACD) Server
~~id~config	Configuration Server
~~id~sipXrls	Resource List Server
~~id~registrar	Registrar Server
~~mh~	Music on Hold Server
~~id~sipXsaa	Shared Appearance Agent
~~id~xmpprlsclient	XMPP Server

When reporting on calls involving any of these special users, it is recommended to translates these special users to a more meaningful name.

3. Remote Database Access

Postgres database access on sipXecs is setup by default to only allow local connections. In order to provide off-board processing/reporting of CDR data, the following must be performed before any remote connections are attempted:

1. Login (as root) over SSH to the Master Server.
2. Enable Client Authentication
   Edit the PostgreSQL configuration file /var/lib/pgsql/data/pg_hba.conf using a text editor such as “vi”.
   - Append the following configuration line to give access to a network of IP addresses in the w.x.y.z range:

     host all all w.x.y.0/24 trust
   - Save and close the file.
3. Enable networking for Postgresql
   Edit the PostgreSQL configuration file /var/lib/pgsql/data/postgresql.conf using a text editor such as “vi”.
   - Find the configuration line that reads as follows:
     listen_addresses='localhost'
   - If the line is commented out (i.e. # in column one), uncomment it.
   - Set IP address(es) to listen on; you can use a comma-separated list of addresses or '*' indicates all ip addresses.
   - Save the file.
4. Restart the PostreSQL server
   /etc/init.d/postgresql restart

4. CDR Database Access

Connectivity to the CDR database whether local or remote will require the following information:

- Address of the Master Server
- Database: SIPXCDR
- Port: 5432 (default port).
- Username: postgres
- Password: <no password>

It is assumed that the developer will install and utilize the appropriate remote database access tools for the platform required.