

Call Detail Records Information

Database Information

In openUC / sipXecs 4.6 and prior all CDR information is stored in Postgres.

All the information regarding call events is aggregated in the call_state_events table

On every node where proxy runs, a postgres db server runs too and there are saved all call events processed by that node. There is another process named sipXcallresolver (ruby based) and is started by the CDR service that aggregates all call events from all proxy nodes and saves them in SIPXCDR database - call_state_events table on primary node.

The sipXcallresolver performs another aggregation based on data from call_state_events table and creates CDRs records for history, and saves them in the same database - SIPXCDR, in cdrs table

Additional Information

Configuring Call Detail Records: [Configuring Call Detail Recording \(CDR\)](#)

Web Services for CDR Records: [Call Detail Records Web Service](#)

Some more good information: [CDR Extras](#)

Postgres Row Breakdown

Here is a breakdown of the rows:

1. call_id This is a unique identifier for each call, should be used to filter out duplicates records from the csv file
2. from_tag A system identifier
3. to_tag A system identifier
4. caller_aor This is the call originator
5. callee_aor This is the terminating number
6. start_time This is the time the caller initiated the use of a system resource for a call
7. connect_time This is the actual time the caller and callee were connected
8. termination This is whether or not the call was completed (Calls with "F" can be ignored)
9. failure_status This is the error code for a failed call
10. failure_reason This is the reason the call failed
11. call_direction Direction of the call
12. reference Unknown
13. caller_contact Extension and IP address of the originator
14. callee_contact Extension and IP address of the called number
15. caller_internal Internal or external call
16. callee_route Route(s) the call used

CDRS Table

Here is the description of the cdrs table in the postgres data base:

Table "public.cdrs"		
Column	Type	Modifiers
id	bigint	not null default nextval('cdrs_id_seq'::regclass)

call_id	text	not null
from_tag	text	not null
to_tag	text	not null
caller_aor	text	not null
callee_aor	text	not null
start_time	timestamp without time zone	
connect_time	timestamp without time zone	
end_time	timestamp without time zone	
termination	character(1)	
failure_status	smallint	
failure_reason	text	
call_direction	character(1)	
reference	text	
caller_contact	text	
callee_contact	text	
caller_internal	boolean	
callee_route	text	

Example:

SIPXCDR=# select * from view_cdrcs;

id	caller_aor	callee_aor	start_time	connect_time	end_time	duration	termination	failure_status	failure_reason	call_direction
1	"Test User" <sip:2949@sipx.sipfoundry.org>	<sip:2900@sipx.sipfoundry.org;user=phone>	2011-05-03 19:22:14.154	2011-05-03 19:22:23.356	2011-05-03 19:22:25.342	00:00:01.986	C			
2	"R SHEMEL CT 256" <sip:3900@CTAUDCD1>	<sip:2900@10.10.10.1;user=phone>	2011-05-03 19:32:43.78	2011-05-03 19:32:56.515	2011-05-03 19:33:04.956	00:00:08.441	C			
3	"Test User" <sip:2949@sipx.sipfoundry.org>	<sip:919703106641@sipx.sipfoundry.org;user=phone>	2011-05-03 19:36:24.123	2011-05-03 19:36:24.202			F	404	Not Found	
4	"Test User" <sip:2949@sipx.sipfoundry.org>	<sip:93038347175@sipx.sipfoundry.org;user=phone>	2011-05-03 19:36:15.664	2011-05-03 19:36:15.709			F	404	Not Found	
5	"Test User" <sip:2949@sipx.sipfoundry.org>	<sip:919703106641@sipx.sipfoundry.org;user=phone>	2011-05-03 19:39:40.689	2011-05-03 19:39:40.919			F	403	Forbidden	
...										

Call State Events Table

Column event_type holds the event type sent

R - Call Request

T - Transfer call Event

S - Call Established Event

F - Call failed event

E - Normal End Call event

Below is an example with call_state_events table entries given from_url and to_url values. These applies when we try to place a call using a 3PCC

Example:

when from = user1, to = user1 status=R we know that user1 phone is ringing

when from = user1, to = user1 status=S we know that user1 picked up the phone

when from = user1, to = user2 status=R we know that user2 phone started ringing

when from = user1, to = user2 status=S we know that user2 answered the phone

Failure situations:

when from = user1, to = user1, status=F we know that user1 rejected the call

when from = user1, to = user2, status=E we know that user2 rejected the call (if there is no call established event (S) before, we know that user2 never answered the call and rejected it instead)

(If there is an (S) status event before, then user2 normally ended the call)

Call Direction

In the code (the cdr scripts) the call_direction field is never populated.

In the cdr scripts it is declared like this: "# call_direction char(1) Plugin feature see below ", but is never mentioned again.

In the sipXcom wiki only info on this field is : 11. call_direction Direction of the call

Two interesting discussions were found about call_direction are here:

- <http://comments.gmane.org/gmane.comp.voip.sipx.devel/6267>

- <https://www.sipfoundry.org/topic/build-3-7-cdr-call-direction-setting-not-working/>