

Configuring Call Detail Recording (CDR)

Introduction

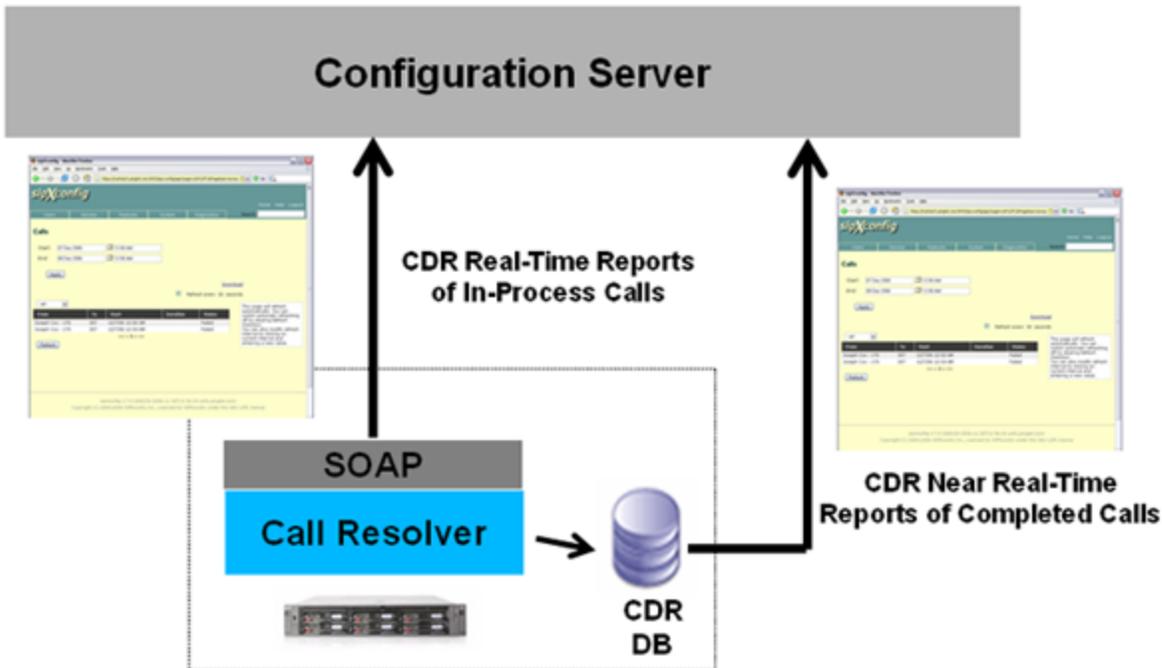
Call Detail Recording (CDR) is a feature which monitors call activity in the sipXecs system and constructs, in a condensed format, records that describe these calls. The records, which are stored in a database, are often used for call accounting and/or general system utilization monitoring purposes.

Although sipXecs does not provide a call accounting reporting mechanism (left for 3rd party application developers which specialize in this), it does provide a near real-time display of active and historic calls as well as a set of basic reports.

Architecture

Call Detail Recording is included as part of the Management Role bundle of sipXecs and is enabled by default. Roles, sometimes referred to as bundles, are pre-configured and cannot be modified. The Management role, by definition, can only operate on the master(primary) server.

The diagram below provides a high-level view of CDR (service is known internally as CallResolver) and its interactions with Configuration Management.



Configuration

Configuration requirements for CDR are very minimal and can be found in the Configuration management under the System/Servers/Services/CDR. General settings (i.e. those most likely to be modified) are normally displayed while Advanced settings are only shown when the 'Show Advanced Settings' action (in the upper right hand side of the settings page) is selected.

General settings:

- Enable Call Detail Records (default checked)
This setting determines whether or not to monitor call activity and record details in CDR records. This parameter is best left enabled(checked).
- Purge aged records daily (default checked)
Purging of records from the database is done to conserve disk space as well as to remove old records which are no longer required. This setting is to enable/disable whether or not to check on a daily basis for old records to be deleted.
- Purge age for CDRs (default 185 days which is approximately 6 months)
If purging of old records is selected, this parameter indicates the maximum age(in days) that the records will be kept. Records that are older than this value will be deleted from the database.
- Logging Level (default NOTICE)
_The logging level typically does not need to be changed. If a CDR issue occurs and is reproducible, it is often recommended to set the logging level to DEBUG, reproduce the issue and generate a snapshot to be submitted when a problem is to be reported.

Advanced settings:

- All General settings.

- Purge age for CSEs (Call State Events)
Configuration options page is found under *Calling / Configuration / Call Detail Records (CDRs)*. By default CSE / CDR recording is disabled. In order to enable it, check the first three checkboxes shown below. The first screenshot shows typical settings and the second screenshot adds advanced settings.

Navigating to CDR Service

The screenshot shows the SIPXecs web interface. The 'System' tab is selected. Under 'Configure', 'Servers' is expanded to show 'pdx.epiphon.com'. The 'Services' section is active, displaying a table of services. The 'CDR' service is highlighted with a red box. Below the table are buttons for 'Start', 'Stop', 'Restart', and 'Refresh'. A warning message on the right states: 'Warning: Restarting services causes service interruption. Do it only if a service requires restart or is not working properly. In such case you might want to take a snapshot and report an issue.' A note below the warning says: 'This page will refresh automatically. You can switch automatic refreshing off by clearing the Refresh checkbox. You can also modify the refresh interval by clicking on the current interval and then enter a new value.'

Name	Status	Role
Park	Running	Primary SIP Router
Statistics	Running	Management
Configuration	Running	Management
ACD	Running	Call Center
ACD Agent Status	Running	Call Center
ACD Statistics	Running	Call Center
Media Services	Running	Conferencing Primary SIP Router
Voicemail	Running	Voicemail
Voicemail MWI	Running	Voicemail
SIP Trunking	Running	SIP Trunking
Paging	Running	Primary SIP Router
Media Relay	Running	Primary SIP Router
SIP Registrar	Running	Primary SIP Router
CDR	Running	Management
SIP Proxy	Running	Primary SIP Router
Presence	Running	Primary SIP Router
Auto Attendant	Running	Primary SIP Router

CDR Service Settings

The screenshot shows the 'CDR' configuration page. The 'Advanced' section is expanded, showing several settings:

- Enable Call Detail Records: (Default: checked)
- Purge aged records daily: (Default: checked)
- Purge age for CDRs: 355 (Default: 35)
- Purge age for CSEs: 7 (Default: 7) - This field is highlighted with a red box.
- Logging Level: DEBUG (Default: NOTICE)

Buttons for 'OK', 'Apply', and 'Cancel' are at the bottom. A red 'Advanced' label is visible next to the 'Purge age for CSEs' field.

Note: Modifications to these parameters will require a restart of CDR and the Proxy Service. Service interruption is typically between 5 and 10 seconds and no in-process calls are affected.

Reporting

sipXecs Configuration Management provides the ability to examine information on CDR records in 3 ways:

- Near real-time reporting of currently active answered calls (via a CDR supplied SOAP interface)
- Viewing of historical calls based on a date range.
- Report generation

All 3 reporting mechanisms are found in the Configuration Management under Diagnostics/Call Detail Records.

Real-Time Active Calls

Real-time active calls is a display of all currently active answered calls in the system. Details of the call indicate the Originator and Destination of the call as well as the start time and duration of the call from the time it was answered. The information displayed for the calls is automatically refreshed every 30 seconds. The rate of the refresh can be changed by selecting the current value and then editing it with a new value.

The screenshot shows the SIPX ECS web interface. The top navigation bar includes 'Users', 'Devices', 'Features', 'System', and 'Diagnostics' (highlighted with a red box). On the left, a sidebar menu has 'Active' (highlighted with a red box), 'Historic', and 'Reports'. The main content area is titled 'Calls' and features a table with columns: From, To, Start, Duration, and Status. A 'Refresh' button is located below the table. To the right of the table, there is a checkbox for 'Refresh every 30 seconds' (highlighted with a red box) and a text box for '30 seconds'. A 'Download' button is also visible. On the far right, there is explanatory text about CDR database entries and a note about automatic refreshing. The footer contains copyright information for Pingtel Corp. and sipXconfig.

Historical CDR Records

The historical CDR record display shows those calls which are completed and stored in the database. A date range can be specified for display of the calls during the specified time period. Like the Real-time active calls display, the information can be refreshed automatically at a defined rate (30 seconds default).

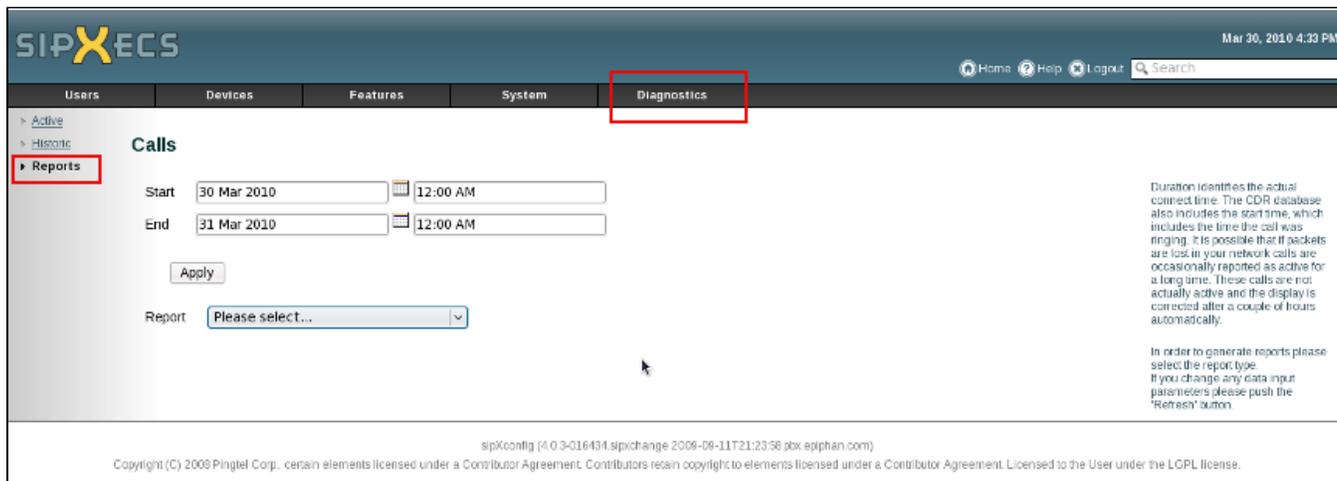
The screenshot shows the SIPX ECS web interface for historical CDR records. The top navigation bar is the same as in the previous screenshot, with 'Diagnostics' highlighted. The sidebar menu has 'Historic' (highlighted with a red box) selected. The main content area is titled 'Calls' and includes date range selection fields for 'Start' (01 Mar 2010 12:00 AM) and 'End' (31 Mar 2010 12:00 AM), with an 'Apply' button. Below these fields is a dropdown menu set to '- all -'. The table structure is identical to the real-time view, with columns for From, To, Start, Duration, and Status, and a 'Refresh' button. The 'Refresh every 30 seconds' checkbox and text box are also present, with the '30 seconds' value highlighted by a red box. A 'Download' button is located to the right of the table. Explanatory text and footer information are consistent with the previous screenshot.

Report Generation

Although the historical record display is informative, it is difficult to get the "big" picture of the call activity in the system. sipXecs 4.0 has added a basic reporting capability along with the following reports:

- CDR table report
List of all calls in the date range specified.
- CDR Report grouped by extension
List of all calls in the date range specified but grouped by the call originator.
- CDR Graph 10 most active callers
A bar chart depicting the 10 most active callers in the system.

- CDR Graph 10 most active receivers
A bar chart depicting the 10 most active extension recipients in the system.
- Minutes of outgoing calls
A bar chart depicting the amount of minutes used by the most active extensions.
- Call Status
A pie chart showing the percentage of calls which completed in particular states (i.e. Completed, Failed, ...)



Remote Database Access

The built-in reporting capabilities of sipXecs, although powerful, are sometimes not sufficient for a customer's needs. In order to fulfill the reporting/analysis requirements, it is often necessary to utilize a 3rd party reporting application or have the customer create one. These specialized applications must be run outside of sipXecs and as such would require remote access to the CDR database.

CDR records are stored in a Postgres database (Database SIPXCDR) on the sipXecs system. The default configuration for this database is to only allow local connections. In order to open up the ability for remote database access, two Postgres configuration files must be modified.

Please note that only an experienced Administrator should attempt this procedure

PostgreSQL pg_hba.conf

The pg_hba.conf configuration file is found under /var/lib/pgsql/data. The steps detailed below are an example of how to allow remote connections from IP address 192.168.5.0 .

1. Open up a terminal to the sipXecs system and login as root.
2. Change to the directory /var/lib/pgsql/data. (i.e. cd /var/lib/pgsql/data)
3. Edit the pg_hba.conf file. (i.e. vi pg_hba.conf)
4. Add the following line in the section commented as IPv4 remote connections.
host all all 192.168.5.0/24 trust
5. Save the file change.

PostgreSQL postgresql.conf

The postgresql.conf configuration file is found under /var/lib/pgsql/data. The steps detailed below are an example of how to tell PostgreSQL to listen for connection requests from any address.

NOTE: This configuration file contains a lot of information. Caution must be taken when editing.

1. Open up a terminal to the sipXecs system and login as root.
2. Change to the directory /var/lib/pgsql/data. (i.e. cd /var/lib/pgsql/data)
3. Edit the postgresql.conf file. (i.e. vi postgresql.conf)
4. Add the following line (or uncomment it if it already exists) in the section commented as CONNECTIONS AND AUTHENTICATION.
listen_addresses = ''*
5. Save the file change.

PostgreSQL configuration reload

Once all the configuration changes have been made to allow remote connectivity to the database, PostgreSQL must be told to reload its configuration in order to recognize the changes. Reload the configuration can be done in two methods.

1. Run pg_ctl as the user "postgres" to tell PostgreSQL to reload the configuration (i.e. >pg_ctl reload -D /var/lib/pgsql/data) **No service disruption**
2. Restart the PostgreSQL service (i.e. >service postgresql restart) **Service disruption**

Recorded call events

SIPXCDR database stores all call events occurred during the process of establishing a call. Every call event is saved in the **call_state_events** table

Analysis

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/>