

# CDR via ESL

## About

[Event Socket Library](#) allows one to retrieve CDRs in realtime. The CDR information are provided in the event **CHANNEL\_HANGUP\_COMPLETE**.

If you set channel variable `hangup_complete_with_xml=true`, then the body of that event will be a full XML CDR

See also `mod_cdr_*` modules (e.g., [mod\\_cdr\\_pg\\_csv](#), [mod\\_cdr\\_csv](#), [mod\\_cdr\\_mongodb](#)).

## Example in Perl on how to use ESL to display CDRs

```
#!/usr/bin/perl -wT
# handle_cdr.pl
# Connect to event socket, listen for CHANNEL_HANGUP_COMPLETE events
# Uses event data to create custom CDRs

use strict;
use warnings;

use lib '/usr/src/freeswitch.git/libs/esl/perl';

use ESL;
my $host = 'localhost';
my $port = '8021';
my $pass = 'ClueCon';
# Connect to event socket
my $con = ESL::ESLconnection->new($host, $port, $pass);
if ( ! $con ) {
    die "Unable to establish connection to FreeSWITCH.\n";
}
# Listen for events of type CHANNEL_HANGUP_COMPLETE only
$con->events('plain', 'CHANNEL_HANGUP_COMPLETE');

print "Connected to FreeSWITCH $host:$port and waiting for events...\n\n";
while ( $con->connected() ) {
    my $e = $con->recvEvent();
    my @raw_data = split /\n/, $e->serialize();
    my %cdr;
    foreach my $item ( @raw_data ) {
        #print "$item\n";
        my ($header, $value) = split /: /, $item;
        $header =~ s/^variable_//;
        $cdr{$header} = $value;
    }
    # %cdr contains a complete list of channel variables
    print "New CDR: ";
    print $cdr{uuid} . ', ' . $cdr{direction} . ', ';
    print $cdr{answer_epoch} . ', ' . $cdr{end_epoch} . ', ';
    print $cdr{hangup_cause} . "\n";
}
```