- NOTE: This is old! We keep it for posterity's sake but we highly recommend ESL FreeSWITCH::Client IS BROKEN in current releases so DO NOT use it, make sure to move to using Perl ESL if you run into any issues with it.

I am not a big perl guy, I dont know the language that well, but this is a functional example that should help some people get started. If you look in the FreeSWITCH repository under scripts/perl/ you will noticed a FreeSWITCH directory containing Client.pm. You must preserve the FreeSWITCH directory name and copy that and Client.pm to some place that is in your perl include path. Your current working directory is normally in that path.

fs.pl is a console app that uses readline and this library. Another example which monitors DTMF events from a conference and plays a file when it gets them follows. You need to bind DTMF to the 'event' item in Mod conference for it to be sent to event socket for consumption. For example, in <caller-controls> something like:

```
    <control action="event" digits="0"/>
    <control action="event" digits="1"/>
    ... repeat for any digit command you want to handle ...
```

```
#!/usr/bin/perl
use FreeSWITCH::Client;
use POSIX ':signal_h'; # used for alarm to ensure we get heartbeats
use Data::Dumper; # used to print out myhash debug info


# configure these
my $password  = "ClueCon";                    # the password for event socket
my $host      = "localhost";                  # the hostname to connect to
my $port      = 8021;                         # the port to connect to
my $timeout   = 30;                           # seconds to expect a heartbeat event or reconnect

# hash for music, default must be defined, if you define other entities then the conference name
my %soundfile = (
                 'default' => '/sounds/fpm-calm-river.wav',
                 '123'     => '/sounds/welcome.raw',
             );



##
# dont touch  these
##
my $fs;
my $lastheartbeat;



# this connects to the event socket
sub es_connect()
{
    print "Connecting to $host:$port\n";
    $fs = init FreeSWITCH::Client {-password => $password, -host => $host, -port => $port};
    if(defined $fs) {
        $fs->sendmsg({'command' => 'event plain heartbeat CUSTOM conference::maintenance'});
        $lastheartbeat = time;
    }
}


sigaction SIGALRM, new POSIX::SigAction sub {
```

```
    if ($lastheartbeat < (time - $timeout)) {
        print "Did not receive a heartbeat in the specified timeout\n";
        if (defined $fs) {
            $fs->disconnect();
            undef $fs;
        }
        es_connect();
    }

    # reset the alarm
    alarm $timeout;
} or die "Error setting SIGALRM handler: $!\n";



es_connect();
alarm $timeout;

while (1) {
    if(defined $fs) {
        my $reply = $fs->readhash(undef);
        if ($reply->{socketerror}) {
            es_connect();
        }

        if ($reply->{body}) {
            $myhash = $reply->{event};

            if ($myhash->{'event-name'} eq "HEARTBEAT") {   ## Deal with heartbeats
                $lastheartbeat = time;
                print "Got a heartbeat\n";

            } elsif ($myhash->{'event-subclass'} eq "conference::maintenance") {   ## deal with C

                if($myhash->{'action'} eq 'dtmf') {  ## DTMF event
                    print "conf: $myhash->{'conference-name'}\tmember: $myhash->{'member-id'}\tDT
                    if(defined $soundfile{$myhash->{'conference-name'}}) {
                        $fs->sendmsg({'command' => "api conference $myhash->{'conference-name'} p
                    } else {
                        $fs->sendmsg({'command' => "api conference $myhash->{'conference-name'} p
                    }

                } else {  ## Just print out all other events
                    print "conf: $myhash->{'conference-name'}\tmemid: $myhash->{'member-id'}\tact
                }
            } else {  ## Unknown event
                print Dumper $myhash;
#           print "$reply->{body}\n"; # print out what was sent, myhash is translated by Client.p
            }

        }
    }
}
```