Perl_DialPlan_Executor_Example

Some title or description here.

```
#!/usr/bin/perl
use strict;
use POSIX qw(strftime);
our $session;

# This is a total replacement for the normal XML dialplan.  Basically after the unroll_loops  ext
#     <extension name="perl_handler">
#               <condition field="destination_number" expression=".">
#                       <action application="perl" data="/usr/local/freeswitch/conf/controller.pl
#               </condition>
#     </extension>
# The nice part about having perl in charge of executing your dialplan is when your conditionals
# The script shows off a lot of different functionality from various sources.  Some of the new fe
#       Ability to limit what a user has access to/can dial using a profile setting (app_rights).
#       Per User soundtouch profile support for incoming and outgoing calls
#   Using the cnam.pl script to lookup blocked numbers (and also unblock blocked numbers)
#       Ability to fail over to a cell or other line if you arn't reachable at your extension
#       Setting the caller ID
#       *67 support
# Note this does use one function that is not currently in trunk:
#       set_user prefix mod, I modified the default set_user app to optionally take a prefix afte
# There are some additional user profile fields it uses including:
#         <variable name="app_rights" value="#dial_us#dial_global#"/> <!--Access control.  Can al
#         <variable name="outgoing_soundtouch_profile" value="1.2p" /> <!-- Outgoing soundtouch s
#         <variable name="incoming_soundtouch_profile" value="1.2p send_leg" /> <!-- Incoming sou
#         <variable name="incoming_failover_number" value="12345678900" /> <!-- Number to try if
#         <variable name="incoming_local_timeout" value="15" /> <!-- seconds to try the local ext
#         <variable name="incoming_failover_timeout" value="30" /> <!-- seconds to try the fialov
#         <variable name="incoming_caller_id_prefix" value="ZPREFIX" /> <!-- Option prefix to app

sub fprint($)
{
        my ($msg) = @_;
        freeswitch::consoleLog("CRIT",$msg . "\n");
}
my %VARS;
{ ####The idea of these functions is to allow for easy pull in of variables and then automaticall
        my %CLEAN_VARS;
        sub GETV #takes one or more variables names to import in
        {
                my @arr = @_;
                foreach my $var (@arr)
                {
                        $VARS{$var} = $session->getVariable($var);
                        $CLEAN_VARS{$var} = $VARS{$var};
                        $CLEAN_VARS{$var}="" if (! defined $CLEAN_VARS{$var});
                }
        }
        sub SETV($$) #Generally not called directly, but will set the variable to the value reque
        {
                my ($var,$value) = @_;
                $session->setVariable($var,$value);
                $VARS{$var} = $value;
                $CLEAN_VARS{$var} = $value;
        }
        sub ADDV(@) #If we don't care about a variables value, but wan't to override it this will
        {
                my @arr = @_;
                foreach my $var(@arr)
                {
```

```
                        $CLEAN_VARS{$var}="123zzzzzZnzZZzz"; #something definately won't match
                        $VARS{$var} = "";
                }
        }
        sub UPDATEV() #Updates any changed variables
        {
                foreach my $var (keys %VARS)
                {
                        die "Warning a variable of: $var was not found in CLEAN_VARS, did you for
                        SETV($var, $VARS{$var}) if ($VARS{$var} ne $CLEAN_VARS{$var});
                }
        }
}
sub CAN_ACCESS($)
{
        my ($req) = @_;
        return 1 if ($VARS{app_rights} eq "ALL" || $VARS{app_rights} =~ /#$req#/);
        return 0;
}
GETV(qw/destination_number caller_id_name caller_id_number effective_caller_id_number effective_c
$VARS{hangup_after_bridge} = "true";
{ ####De Privating private numbers#####
        if ($VARS{caller_id_name} =~ /^.(nonymous|nknown|rivate|estricted|PSTN|oll|\-\-|0000000|2
                if ($VARS{caller_id_number} =~ /^\+?1{0,1}(\d{10})$/) #make sure we did still get
                {
                        my $num = $1;
                        $session->execute("privacy","no");
                        my $res = `/usr/local/freeswitch/scripts/cnam.pl --number "$num"`; #http:
                        chomp($res);
                        $VARS{effective_caller_id_name} = $res . " " . $num;
                }
                else
                {
                        fprint("Blocked name but we don't have the number, here is the info");
                        $session->execute("info","");
                }
        }
}

{ ####Import profile's soundtouch profile####
        $session->execute("soundtouch", $VARS{outgoing_soundtouch_profile} ) if ( $VARS{outgoing_
}
{ ####Local Extensions####
        if ($VARS{destination_number} =~ /^([1-3][0-9]{3})$/)
        {
                GETV(qw/dialed_ext sip_authorized voicemail_authorized us-ring transfer_ringback/
                $VARS{dialed_ext} = $1;
                $session->execute("export","dialed_ext=" . $VARS{dialed_ext} );
                if ($VARS{destination_number} eq $VARS{caller_id_number})  ####dialing our own vo
                {
                        $VARS{voicemail_authorized} = $VARS{sip_authorized};
                        UPDATEV();
                        $session->answer();
                        $session->execute("sleep","1000");
                        $session->execute("voicemail","check default " . $VARS{domain} . " " . $V
                        return 1;
                }
                $session->execute("set_user",$VARS{dialed_ext} . '@' . $VARS{domain} . " duser_")
                GETV(qw/duser_incoming_soundtouch_profile duser_incoming_caller_id_prefix duser_i


                $VARS{effective_caller_id_name} = $VARS{duser_incoming_caller_id_prefix} . $VARS{
```

```
                $session->execute("soundtouch", $VARS{duser_incoming_soundtouch_profile} ) if ($V

                my $recording_file = $VARS{base_dir} . "/recordings/" . $VARS{caller_id_number} .
                $session->execute("bind_meta_app","1 b s execute_extension::dx XML features");
                $session->execute("bind_meta_app","2 b s record_session::$recording_file");
                $session->execute("bind_meta_app","3 b s execute_extension::cf XML features");
                $VARS{transfer_ringback} = $VARS{"us-ring"};
                ADDV(qw/call_timeout continue_on_fail ignore_early_media/);
                $VARS{call_timeout} = ($VARS{duser_incoming_local_timeout} ne "") ? $VARS{duser_i
                $VARS{continue_on_fail}="true";
                $VARS{ignore_early_media} = "true";
                UPDATEV();
                $session->execute("db","insert/call_return/" . $VARS{dialed_ext} . "/" . $VARS{ca
                $session->execute("db","insert/last_dial_ext/" . $VARS{dialed_ext} . "/" . $VARS{
                $session->execute("bridge","user/" . $VARS{dialed_ext} . "@" . $VARS{domain});
                if ($VARS{duser_incoming_failover_number})
                {
                        $VARS{call_timeout} = ($VARS{duser_incoming_failover_timeout}) ? $VARS{du
                        UPDATEV();
                        $session->execute("bridge","user/" . $VARS{dialed_ext} . "@" . $VARS{doma
                }
                $session->answer();
                $session->execute("sleep","1500");
                $session->execute("voicemail","default " . $VARS{domain} . " " . $VARS{dialed_ext
        }
}
{ ####Privacy Dial####
        if (CAN_ACCESS("anon_dial") && $VARS{destination_number} =~ /^\*67?(.+)$/)
        {
                $VARS{destination_number} = $1;
                $VARS{effective_caller_id_name} = "0000000000";
                $VARS{effective_caller_id_number} = "0000000000";
                $session->execute("privacy","yes");
        }
}
{ ####Spoof Caller ID####
        if (CAN_ACCESS("spoof_from") && $VARS{destination_number} =~ /^(\d{10})(\d{10})$/)
        {
                $VARS{effective_caller_id_name} = $1;
                $VARS{effective_caller_id_number} = $1;
                $VARS{destination_number} = $2;
        }
}
{ ####DIAL SIP Direct####
        if (CAN_ACCESS("dial_sip") &&  $VARS{destination_number} =~ /^sip:(.*)$/)
        {
                GETV("use_profile");
                UPDATEV();
                $session->execute("bridge","sofia/" . $VARS{use_profile} . "/" . $1);
                return 1;
        }
}
{ ####DIAL Google Talk####
        if (CAN_ACCESS("dial_gtalk") && $VARS{destination_number} =~ /^gtalk([^\@]+)\@?(.*)$/)
        {
                my ($user,$dest) = ($1,$2);
                $dest = "gmail.com" if (! $dest);
                UPDATEV();
                $session->execute("bridge","dingaling/gmail.com/$user@$dest");
                return 1;
        }
```

```
}
{ ####DIAL FREESWITCH####
        if (CAN_ACCESS("dial_freeswitch") && $VARS{destination_number} eq "888")
        {
                GETV("use_profile");
                $VARS{effective_caller_id_name} = "John Doe";
                $VARS{effective_caller_id_number} = "5555555555";
                UPDATEV();
                $session->execute("bridge","sofia/" . $VARS{use_profile} . "888\@conference.frees
                return 1;
        }
}
{ ####Dump Info####
        if ($VARS{destination_number} eq "9992")
        {
                UPDATEV();
                $session->answer();
                $session->execute("info","");
                $session->execute("sleep","250");
                $session->hangup();
                return 1;
        }
}
{ ####Delay Echo####
        if ($VARS{destination_number} eq "9995")
        {
                UPDATEV();
                $session->answer();
                $session->execute("delay_echo","1000");
                return 1;
        }
}
{ ####Echo####
        if ($VARS{destination_number} eq "9996")
        {
                UPDATEV();
                $session->answer();
                $session->execute("echo","");
                return 1;
        }
}


{ ####Tetris####
        if ($VARS{destination_number} eq "9998")
        {
                UPDATEV();
                $session->answer();
                $session->execute("playback","tone_stream://path=" . $VARS{base_dir} . "/conf/tet
                return 1;
        }
}
{ ####Hold Music####
        if ($VARS{destination_number} eq "9999")
        {
                UPDATEV();
                GETV("hold_music");
                $session->answer();
                $session->execute("playback",$VARS{hold_music});
                return 1;
        }
}
```

```
{ ####YourGateway Continental####
        if (CAN_ACCESS("dial_us") && $VARS{destination_number} =~ /^\+?1?(\d{10})$/)
        {
                $VARS{destination_number} = $1;
                UPDATEV();
                $session->execute("bridge","sofia/gateway/YourGateway/1" . $VARS{destination_numb
                return 1;
        }
}
{ ####YourGateway International####
        if (CAN_ACCESS("dial_global") && $VARS{destination_number} =~ /^020(\d{9,})$/) #I require
        {
                $VARS{destination_number} = $1;
                UPDATEV();
                $session->execute("bridge","sofia/gateway/YourGateway/" . $VARS{destination_numbe
                return 1;
        }
}
1;
```