This is an example of how to dial over Skype from a local extension.

Please keep in mind that this script does not provide any authentication. If you intend on using it as a DISA route, you should probably add PIN protection to the script or dialplan.

This example has two components:

1. The XML dialplan
   Used to launch the directory, and dial out.
2. The Perl script
   The actual directory 'meat'.

This script relies on your FreeSWITCH installation to have the following working modules:

- mod_perl
- mod_skypopen
- *mod_flite*
- *mod_dialplan_xml*

You may substitute *mod_flite* and *mod_dialplan_xml* and update your script where appropriate.

Dialplan

I have this in 'dialplan/local/skype.xml'

```
<extension name="skypeDir">
  <condition field="destination_number" expression="^1024$">
     <action application="perl" data="skype.pl"/>
  </condition>
</extension>

<extension name="skypeDirDial">
  <condition field="destination_number" expression="^skypeDirDial$">
    <action application="set" data="instant_ringback=true"/>
    <action application="set" data="ringback=${us-ring}"/>
    <action application="set" data="transfer_ringback=${us-ring}"/>
    <action application="set" data="call_timeout=30"/>
    <action application="set" data="hangup_after_bridge=true"/>
    <action application="bridge" data="skypopen/skype1/${destuser}"/>
  </condition>
</extension>
```

Script

I place this in 'scripts/skype.pl'

```
####
# skype.pl - skype directory example
####
# by daniel o'neill (http://wiki.freeswitch.org/wiki/Skypopen_Directory)
####
```

```
my $api = new freeswitch::API;
my $e = new freeswitch::EventConsumer("custom", "skypopen::incoming_raw");
my $c = 0; # incremented, counter used to correlate skypopen API responses to their requests

my $timeoutcheck = 30000; # time to wait for the user to dial an extension, in milliseconds.

my $interface = 'skype1'; # the skype interface to dial-out on.
                          # you could also use 'ANY', or read this from $session->getVariable() a

my $dp_context   = 'local';        # or, more commonly, 'default'
my $dp_extension = 'skypeDirDial'; # as located in the dialplan context
my $dp_format    = 'XML';          # or perhaps 'YAML'

# TTS engine to use.
# You could also use Cepstral, for example
$session->set_tts_parms("flite", "slt"); # awb, kal, rms or slt
# $session->set_tts_parms("cepstral", "Amy");

freeswitch::consoleLog( "info", "Entering Skype.pl directory.\n" );

$session->answer();

$friendList = getFriends( $e ); # this is actually pretty fast

# Gather our contacts
my $contacts;
my $d = 0; # 'online' or 'away' counter
foreach my $a(@{ $friendList }) {
        $contacts->{$a}->{'username'} = $a;
        $contacts->{$a}->{'status'} = getStatus( $e, $a ); # for less than 100, this is fairly qu
        if( $contacts->{$a}->{'status'} eq 'ONLINE'
         || $contacts->{$a}->{'status'} eq 'AWAY' ) {
                $contacts->{$a}->{'available'} = 1;
                $d++;
        }
}

# if < 10 contacts online, user may dial 1 through 9
my $len = 1;
if( $d > 999 ) {
        # if > 999 and < 10000 contacts online, user may dial 0001 through 9999
        $len = 4;
} elsif( $d > 99 ) {
        # if > 99 and < 1000 contacts online, user may dial 001 through 999
        $len = 3;
} elsif( $d > 9 ) {
        # if > 9 and < 100 contacts online, user may dial 01 through 99
        $len = 2;
}

# sort alphabetically by username
my @ckeys = sort { $a->{'username'} cmp $b->{'username'} } keys( %{$contacts} );
$d = 1; # entries begin at extension 1

# we'll speak this string later on.  if you want to say something before rambling off the directo
my $final = 'Skype directory. ';

# compile our spoken directory
foreach my $ukey( @ckeys ) {
        $a = $contacts->{$ukey};
        if( $a->{'available'} == 1 ) {
                my $user = $a->{'username'};
```

```
                my $num = sprintf('%0'.$len.'d', $d); # pad extension with zeros (eg, 001) so all
                $d = 1 + $d; # increment our extension
                $n = $num;
                $n =~ s/(\d)/\1,/gs; # makes 'three one one' instead of 'three hundred and eleven
                $final .= "For $user, dial $n. "; # this will be spoken
                $contacts->{$ukey}->{'number'} = $num; # assign the extension
        }
}

if( length($final) == 0 ) {
        $session->speak("No skype contacts are online.  Please try later.");
        return bail();
} else {
        # if you seriously have 9999 contacts, you may have to wait a while for your name to come
        $session->speak($final);

        # this method is simpler than a non-blocking solution, but requires the user to sit throu
        my $sel = $session->getDigits( $len, "#", $timeoutcheck );
        if( $sel && length("$sel") != 0 ) {
                my $user = getUser( $contacts, $sel );
                if( $user == -1 ) { return bail(); }

                # to see how this works, refer to the XML dialplan accompanying this script
                $session->setVariable( "destuser", $user );
                $session->transfer( $dp_extension, $dp_format, $dp_context );
        } else {
                $session->speak("Sorry, I didn't get a selection.  Goodbye.");
                return bail();
        }
}

sub getUser {
        my $users = shift;
        my $dtmfid = shift;
        foreach my $k( %{ $users } ) {
                if( $k->{'number'} eq $dtmfid ) {
                        return $k->{'username'};
                }
        }
        return -1;
}

sub bail {
        $session->hangup();
        return 1;
}

sub getStatus {
        my $e = shift;
        my $u = shift;

        if( !$session->ready() ) { return bail(); }
        $api->executeString( "skypopen $interface #$c GET USER $u ONLINESTATUS" );
        if( !$session->ready() ) { return bail(); }

        $ce = $e->pop(1);
        if( !$ce ) { return bail(); }

        my $body = $ce->getBody();
        my $status = 'unknown';
        if( $body =~ m/^#$c USER $u ONLINESTATUS (.*)$/ ) {
                $status = $1;
```

```
        } else {
                freeswitch::consoleLog( "info", "Skype.pl: Unexpected response: $body\n" );
        }

        $c=1+$c;

        return $status;
}

sub getFriends {
        my $e = shift;

        if( !$session->ready() ) { return bail(); }
        $api->executeString( "skypopen $interface #$c SEARCH FRIENDS" );
        if( !$session->ready() ) { return bail(); }

        $ce = $e->pop(1);
        if( !$ce ) { return bail(); }

        my $body = $ce->getBody();

        my @friends;
        if( $body =~ m/^#$c USERS (.*)$/ ) {
                @friends = split(/, /, $1);
        } else {
                freeswitch::consoleLog( "info", "Skype.pl: Unexpected response: $body\n" );
        }

        $c=1+$c;

        return \@friends;
}

return 1;
```

Once this is done, simply dial extension *1024* (unless you changed it) to be routed to the directory, and follow the voice prompts.

For fun, here's a Python version:

```
import os
import re
from operator import itemgetter, attrgetter
from freeswitch import *

requestCount    = 1 # incremented, counter used to correlate skypopen API responses to their requ
interface       = 'skype1'

def handler(session, args):
        consoleLog( 'info', "Entering Skype.py directory.\n" )

        timeoutcheck    = 30000                 # time to wait for the user to dial an extension,
        dp_context      = 'local'               # or, more commonly, 'default'
        dp_extension    = 'skypeDirDial'        # as located in the dialplan context
        dp_format       = 'XML'                 # or perhaps 'YAML'

        eCon = EventConsumer('custom', 'skypopen::incoming_raw')

        session.set_tts_parms("flite", "slt") # awb, kal, rms or slt

        session.answer()
```

```
        friendList = getFriends( eCon, session, {} )

        d = 0 # 'online' or 'away' counter

        contacts = {}
        for a in friendList:
                contacts[a] = {}
                contacts[a]['username'] = a
                contacts = getStatus( eCon, session, a, contacts )
                if ( contacts[a]['status'] == 'ONLINE' ) or ( contacts[a]['status'] == 'AWAY' ):
                        contacts[a]['available'] = 1
                        d = d + 1
                else:
                        contacts[a]['available'] = 0

        # if < 10 contacts online, user may dial 1 through 9
        len = 1;
        if( d > 999 ):
                # if > 999 and < 10000 contacts online, user may dial 0001 through 9999
                len = 4
        elif( d > 99 ):
                # if > 99 and < 1000 contacts online, user may dial 001 through 999
                len = 3
        elif( d > 9 ):
                # if > 9 and < 100 contacts online, user may dial 01 through 99
                len = 2

        final = ''
        d = 1 # entries begin at extension 1

###
# This doesn't work, and I don't know Python, but the idea here is to sort the entries alphabetic
###
#       # sort alphabetically by username
#       ckeys = sorted( contacts.iteritems(), key=itemgetter('username') )
#
#       # compile our spoken directory
#       for k in ckeys:
#               a = contacts[k]

        for username in contacts.keys():
                if( contacts[username]['available'] == 1 ):
                        num = "%d" % d
                        num = num.zfill(len) # pad extension with zeros (eg, 001) so all extensio
                        d = 1 + d # increment our extension
                        n = num
                        rx = re.compile(r'(\d)')
                        n = rx.sub( r'\1 ', num ) # say "three one three" instead of "three hundr
                        final += "For %s, dial %s. " % (username, n) # this will be spoken
                        contacts[username]['number'] = num # assign the extension
                else:
                        contacts[username]['number'] = ''

        if( final == '' ):
                session.speak("No skype contacts are online.  Please try later.")
                return bail(session)
        else:
                # if you seriously have 9999 contacts, you may have to wait a while for your name
                session.speak( 'Skype directory.  %s' % (final) )

                # this method is simpler than a non-blocking solution,
```

```
                # but requires the user to sit through each name before
                # allowing them to dial their extension.
                sel = session.getDigits( len, "#", timeoutcheck )
                if( sel and sel != '' ):
                        user = getUser( contacts, sel )
                        if( user == -1 ):
                                return bail(session)

                        # to see how this works, refer to the XML dialplan accompanying this scri
                        session.setVariable( "destuser", user )
                        session.transfer( dp_extension, dp_format, dp_context )
                else:
                        session.speak("Sorry, I didn't get a selection.  Goodbye.")
                        return bail(session)

def getUser(users, dtmfid):
        for k in users.keys():
                if( users[k]['number'] == dtmfid ):
                        return users[k]['username']
        return -1

def bail(session):
        session.hangup
        return 1

def getStatus(eCon, session, user, userstatus):
        global requestCount
        global interface

        userstatus[user]['status'] = 'pending'

        if not ( session.ready() ):
                return bail(session)
        API().executeString( "skypopen %s #%d GET USER %s ONLINESTATUS" % (interface, requestCoun
        if not ( session.ready() ):
                return bail(session)

        ce = eCon.pop(1)
        if not ( ce ):
                return bail(session)

        body = ce.getBody()

        rx = re.compile( r"^#(\d+) USER (.*?) ONLINESTATUS (.*)$" )
        m = rx.match(body)
        if (m):
                userstatus[ m.group(2) ]['status'] = m.group(3)
        else:
                consoleLog( "info", "Skype.py: Unexpected response: %s\n" % (body) )

        requestCount = 1 + requestCount

        return userstatus

def getFriends(eCon, session, friends):
        global requestCount
        global interface

        if not ( session.ready() ):
                return bail(session)
        API().executeString( "skypopen %s #%d SEARCH FRIENDS" % (interface, requestCount) )
        if not ( session.ready() ):
```

```
        return bail(session)

ce = eCon.pop(1)
if not ( ce ):
        return bail(session)

body = ce.getBody()

rx = re.compile( "^#(\d+) USERS (.*)$" )
m = rx.match(body)

friends = []
if (m):
        friends = re.split(', ', m.group(2))
else:
        consoleLog( "info", "Skype.py: Unexpected response: %s\n" % (body) )

requestCount = 1 + requestCount

return friends
```