Contents

- 1 Introduction
- 2 Script
- 3 Dialplan
- 4 See Also

Introduction

This script, when set as an extension in the dial plan, will allow an internal caller to enter a phone number. That phone number will then be called on the configured gateway and invited to accept the conference by pressing '1'. Once the callee accepts, both caller and callee will be transferred into the configured conference.

Script

Place the following script in your FreeSWITCH installation in the "scripts" folder and adjust the configuration options to suit your configuration. The configuration below expects a "conference" extension in the dialplan and a gateway called "vitelity-outbound". To use this script as-is, you will need a working TTS module installed.

```
/*
 * conference-dial.js
 *
 * Conference Dialer
 *
 * Allow an internal caller to invite an outside caller into a conference.
 *
 * This script, when set as an extension in the dial plan, will allow an
 * internal caller to enter a phone number.  That phone number will then be
 * called on the configured gateway and invited to accept the conference
 * by pressing 1.  Once the callee accepts, both caller and callee will be
 * transferred into the configured conference.
 *
 * 2009 Patrick W. Barnes < http://patrickwbarnes.com/ >
 *
 * Last Update: 2009-10-13
 *
 */

// == CONFIGURATION OPTIONS ==

var transfer_command = "conference XML default"; // Where to transfer the parties
var gateway = "vitelity-outbound"; // The gateway to use for the call

var timeoutcall = 30; // Length of time to ring target number, seconds
var timeoutphonenumber = 15000; // Length of time to wait for phone number input, ms
var timeoutaccept = 3000; // Length of time to wait for callee to accept, ms

var ttsengine = "flite"; // TTS engine
var ttsvoice = "slt"; // TTS voice
```

```
var ttstimer = "soft"; // TTS timer type

var originate_options = "ignore_early_media=true"; // session.originate flags

// == CONFIGURATION OPTIONS END ==

var digitmaxlength = 0; // Maximum length of input
var speakdigits = 1; // Whether or not to echo digits as dialed
var targetnumber = ""; // Target phone number

// Speak (TTS) to the specified session
function ttsSpeak (session, phrase) {
  session.speak(ttsengine, ttsvoice, phrase, ttstimer);
}

// Accept input digits
//  This callback function is called for each digit dialed
function dtmfRead (session, type, obj, arg) {
  try {
    if (type == "dtmf") {
      // # or * indicates that input is complete
      if (obj.digit == "#" || obj.digit == "*") {
        return(false);
      }
      // Add the dialed digit to the digit string
      dtmf.digits += obj.digit;
      if (speakdigits == 1) {
        // Echo the dialed digits to the user
        session.streamFile("digits/"+obj.digit+".wav");
      }
      // If the maximum length is met, stop taking input
      if (dtmf.digits.length >= digitmaxlength) {
        return(false);
      }
    }
  } catch (e) {
    console_log("err", e+"\n");
  }
  return(true);
}

// Get a phone number
function getPhone () {
  digitmaxlength = 11;
  dtmf.digits = "";
  session.collectInput( dtmfRead, dtmf, timeoutphonenumber );
  return(dtmf.digits);
}

// Execution begins here

var dtmf = new Object(); // Used to store dialed digits
dtmf.digits = "";

// session.ready() indicates when the call can be answered here
if (session.ready()) {
  success = 0; // Will be 1 when the call succeeds
  session.answer(); // Pick up the call

  ttsSpeak(session, "Conference Dialer"); // Greeting

  // Get number to dial
```

```
maxtries = 3; // User gets 3 attempts
tries = 0; // No attempts made so far
ttsSpeak(session, "Please enter the phone number to call, then press pound.");
// Repeat until we have a valid number or maxtries is exceeded
while (targetnumber.length != 7 && targetnumber.length != 10 && targetnumber.length != 11 && tr
  targetnumber = getPhone();
  tries++;
  if (targetnumber.length != 7 && targetnumber.length != 10 && targetnumber.length != 11) {
    ttsSpeak(session, "Invalid number.  Please enter the phone number to call, then press pound
    continue;
  }
  break;
}
if (targetnumber.length != 7 && targetnumber.length != 10 && targetnumber.length != 11) {
  // No valid number after maxtries
  ttsSpeak(session, "Goodbye");
  session.hangup("CALL_REJECTED");
}

ttsSpeak(session, "Initiating call.  Please stand by.");
console_log("info", "Conference Dialer: " + targetnumber + "\n");
if (targetnumber.length == 7) {
  // Add 1+areacode
  areacode = session.getVariable("default_areacode");
  targetnumber = "1"+areacode+targetnumber;
}
if (targetnumber.length == 10) {
  // Add 1
  targetnumber = "1"+targetnumber;
}
// Inherit caller ID info
cid_name = session.getVariable("outbound_caller_name");
cid_number = session.getVariable("outbound_caller_id");
originate_options = originate_options + ",origination_caller_id_name="+cid_name;
originate_options = originate_options + ",origination_caller_id_number="+cid_number;
originate_options = originate_options + ",originate_timeout="+timeoutcall;
// This is where we start the call
outSession = new Session("{"+originate_options+"}sofia/gateway/"+gateway+"/"+targetnumber);
// Once the call is answered, get the callee to accept the call by pressing 1
while (outSession.ready()) {
  // Keep the callee connected even when the caller hangs up and this script has ended
  outSession.setAutoHangup(false);
  digitmaxlength = 1;
  speakdigits = 0;
  dtmf.digits = "";
  maxtries = 3;
  tries = 0;
  ttsSpeak(outSession, "You have been called to join a conference.  Press 1 to accept.");
  while (tries < maxtries) {
    tries++;
    outSession.collectInput( dtmfRead, dtmf, timeoutaccept );
    if (dtmf.digits != "1") {
      ttsSpeak(outSession, "You have been called to join a conference.  Press 1 to accept.");
    }
  }
  if (dtmf.digits == "1") {
    // "transfer" is non-blocking, so we use it instead of "conference"
    session.execute("transfer", transfer_command);
    outSession.execute("transfer", transfer_command);
    success = 1;
  } else {
    // Call not accepted
```

```
      ttsSpeak(outSession, "Goodbye");
      outSession.hangup("NORMAL_CLEARING");
    }
  }
  if (success == 0) {
    ttsSpeak(session, "Call failed");
  }

}
```

Dialplan

With the above script in place, an extension must be added to the dialplan. Add something similar to the following to 'conf/dialplan/default.xml', replacing "502" with the desired extension number.

```
<extension name="conference_dialer">
  <condition field="destination_number" expression="^conference-dialer$|^502$">
    <action application="javascript" data="conference-dial.js" />
  </condition>
</extension>
```

See Also

- Javascript