

Create a Queue Manual Call Back script in Asterisk 1.4

Contributed by Andrew Yager
 Tuesday, 22 September 2009
 Last Updated Wednesday, 23 September 2009

One of the major focuses of my IT company is Voice over IP work, particularly with the open source PBX, Asterisk. Recently, we have had a few customers ask us about customising the asterisk call queue behaviour to do "more". Top on the list of requests is a way to allow customers to leave information, rather than wait on hold, and have this information emailed for a call back at a later time. This is actually more simpler than it sounds, although it does require some careful IVR design, and some poorly documented knowledge of the asterisk queues configuration. This article explains how we did it!

The solution essentially consists of an AGI script which obtains the information from the user, a context to call the AGI script, and some changes to the queues.conf file. (/etc/asterisk/queues.conf for most people)

```
[Sales]
periodic-announce-frequency=180 ;; ADD THIS LINE
strategy=ringall
announce-round-seconds=10
timeout=20
context=queue-breakout-sales ;; ADD THIS LINE
music=default
autofill=no
ringinuse=yes
maxlen=30
periodic-announce=/var/lib/asterisk/sounds/ogm/callback-instructions ;; ADD THIS LINE
reporholdtime=yes
joinempty=no
retry=30
announce-holdtime=no
announce-frequency=90
monitor-format=wav
service = 30
```

The lines in bold are the important ones to pay attention to. The first one indicates how often to play the announcement to your users indicating how to use this service. In our case, we wanted it to happen after three minutes. The second line is the context in the extensions dial plan that contains the breakout options (thanks syednetworks!). These options should be single digit numbers, * or # and reflect the options you want your queues to have. The third line is the audio to play for the announcement.

Next step is to configure your extension context (/etc/asterisk/extensions.conf for most people). Ours looks pretty simple

```
[queue-breakout-sales]
exten => 1,1,AGI(queue-callback.php)
```

The final step is to create an AGI script that has logic sufficient to obtain the information from your users. We chose to use the PHPAGI interface, because we were building an AGI that needed to integrate with some other LOB applications to verify user data (in particular their membership number). Our sample code is below - feel free to use it as a base for any applications you would wish to implement. It implements some pretty basic user input validation - you can easily hook this to work with your application. This file is placed in /var/lib/asterisk/agi-bin/ and is called "queue-callback.php". It does require the PHPAGI libraries to be in the same directory, or in the PHP include path.

```
#!/usr/bin/php -q
<?php
/**
 * Sample Callback AGI Script
 *
 * @author Andrew Yager
 * @version 0.1
 * @copyright Real World Technology Solutions Pty Ltd, 22 September, 2009
 **/
```

```

set_time_limit(0);
require('phpagi.php');

$agi = new AGI();

$agi->answer();
$cid = $agi->parse_callerid();
$agi->stream_file('ogm/callback-so-we-can-arrange-a-callback-we-need-some-more-details');

//member number
$confirmed=false;
$first=true;
$number="";
while(!$confirmed) {
    $agi->stream_file("ogm/callback-afer-the-tone-please-enter-your-member-number-using");
    $result = $agi->get_data('beep', 6000, 20);
    $number = $result['result'];
    if (checkMemberNumber($number)) {
        //valid member number
        $agi->stream_file('ogm/callback-thankyou-ill-just-confirm-that-with-you-i-have');
        $agi->say_digits($number);
        $result = $agi->get_data('ogm/callback-if-this-is-correct-please-press-1-or-press-2-to-enter-it-again', 3000, 1);
        $confirm = $result['result'];
        if ($confirm==1) {
            $confirmed=true;
        } else {
            $number="";
        }
    } else {
        //invalid member number
        $agi->stream_file("invalid");
        $confirmed=false;
        $number="";
    }
}

$memberNo = $number;

$confirmed=false;
$number=$cid['username'];
if (checkPhoneNumber($number)) {
    //valid caller id
    $agi->stream_file('ogm/callback-thankyou-i-have-your-return-phone-number-as');
    $agi->say_digits($number);
    $result = $agi->get_data('ogm/callback-if-this-is-correct-please-press-1-or-press-2-to-enter-it-again', 3000, 1);
    $confirm = $result['result'];
    if ($confirm==1) {
        $confirmed=true;
    } else {
        $number="";
        $confirmed=false;
    }
}

// $number contains a number - either entered by the user, or read from caller id

while(!$confirmed) {
    $agi->stream_file("ogm/callback-after-the-tone-please-enter-your-return-phone-number-including-area-code");
    $result = $agi->get_data('beep', 6000, 11);
    $number = $result['result'];

    if (checkPhoneNumber($number)) {
        $agi->stream_file('ogm/callback-thankyou-ill-just-confirm-that-with-you-i-have');
        $agi->say_digits($number);
    }
}

```

```

$result = $agi->get_data('ogm/callback-if-this-is-correct-please-press-1-or-press-2-to-enter-it-again', 3000, 1);
$confirm = $result['result'];
if ($confirm==1) {
    $confirmed=true;
} else {
    $number="";
}
} else {
    //phone number entered is invalid
    $agi->stream_file('ogm/callback-im-sorry-that-doesnt-seem-to-be-a-valid-phone-number');
}
}

```

```
$phoneNumber = $number;
```

```
$email = "Dear Helpdesk:
```

A member, whose member number is theoretically \$memberNo has requested a call back at ".date("d/m/Y h:i:s")." on \$phoneNumber.

```
Your friends in IT,
RWTS";
```

```
mail("to-address", "Member callback for $memberNo on $phoneNumber", $email, "From: callbacksystem@pbx.local");
```

```
$agi->stream_file("ogm/callback-one-of-our-helpdesk-staff-will-call-you-back-within-24-hours");
```

```
$agi->hangup();
```

```
function checkMemberNumber($number) {
    //only record if we have no number
    if ($number!="") {
        return true;
    }
    return false;
}

```

```
function checkPhoneNumber($number) {
    if (preg_match('/[01]d{9}/', $number)) {
        //valid phone number
        return true;
    } else {
        return false;
    }
}

```

```
}
```

This AGI file references a number of audio files. We have pre-recorded these; here are the scripts that we used:

callback-instructions:

We currently have a large number of calls and so it may take us a while to answer you. If you know your member number and would rather not wait, we can arrange for one of our help desk staff to call you back within 24 hours. If you would like to use this service, please press 1 now, otherwise continue to hold for the next available representative.

callback-so-we-can-arrange-a-callback-we-need-some-more-details:

So that we can arrange a call back, we need some details from you.

callback-afer-the-tone-please-enter-your-member-number-using:

After the tone, please enter your member number using the touch tone keys on your phone followed by the hash key.

callback-thankyou-ill-just-confirm-that-with-you-i-have:

Thank you. I'll just confirm that with you. I have:

callback-if-this-is-correct-please-press-1-or-press-2-to-enter-it-again:

If this is correct please press 1, or press 2 to enter it again.

callback-thankyou-i-have-your-return-phone-number-as:

Thank you. I have your return phone number as

callback-after-the-tone-please-enter-your-return-phone-number-including-area-code:

After the tone, please enter your return phone number, including area code.

callback-im-sorry-that-doesnt-seem-to-be-a-valid-phone-number:

I'm sorry – that doesn't seem to be a valid phone number. Please make sure you have included your area code.

callback-one-of-our-helpdesk-staff-will-call-you-back-within-24-hours:

Thank you. One of our helpdesk staff will call you back within 24 hours.