

How to place automatic test calls

Applies to VoipNow 3.X.X!

By placing automatic test calls, you can receive early alerts whenever your channel provider has issues routing calls to/from your server. This article shows how you can set Asterisk to place calls to a test number within a time interval and get an e-mail alert message whenever such calls fail.

However, this is not the only method you can apply. You can also use [UnifiedAPI](#).

The setup

1. Create a /tmp/autodial.call file.
2. Mind the ownership and permissions.

```
-rwxr-xr-x. 1 asterisk asterisk 106 Jun 16 13:40 /tmp/autodial.call
```

3. The autodial.call file should contain the following:

```
Channel: LOCAL/35555555@app-autodial
Application: Playback
Data: hello-world
CallerID: "autocall" <11111>
```

4. Replace 35555555 with a DID you have on your server and in a format accepted by your provider.
5. Then place the following lines in extensions_custom.conf.

```
[app-autodial]
exten => 35555555,1,Set(_SIPADDHEADER24=X-voipnow-channel: 2)
exten => 35555555,2,Dial(SIP/${EXTEN}@10.150.20.14)
```

6. Again, replace 35555555, the channel ID (in this example, channel ID is 2) and the IP of the channel with those you intend to use on your server.
7. Then run the following command to have the call placed towards the provider who will then route the call back to your server.

```
asterisk -rx "dialplan reload"
```

8. Register a phone terminal to the extension that receives this call.
9. Now make a test call like this:

```
cp /tmp/autodial.call /var/spool/asterisk/outgoing/
```

10. At this point, your phone should ring.

The script

If the call was successfully placed towards your server, we can now make a script to automate the process.

To create the script, follow the steps below:

1. Create a log activity to alert you when calls fail.
2. Make an ICR on the destination extension and set it to hang up the call. This way you won't be billed for the test calls.
3. Activate a Dial-IN Call Event on the extension and point it to http://<your_server_ip>/autodial.php
4. Create the autodial.php file as shown below. Mind the ownership and permissions.

```
-rwxr-xr-x. 1 voipnow voipnow 357 Jun 16 13:07 /usr/local/vipnow/admin/htdocs/autodial.php
```

5. This is the code that will be executed once your destination receives the autodialer call. It will save the contents of the Call Event request to autodial.log.

```
<?php
// set file to write
$file = 'autodial.log';
$somecontent = print_r($_GET, TRUE);
echo $somecontent;// see sample output below
// open file
$fp = fopen($file, 'a') or die('Could not open file!');
// write to file
fwrite($fp, "$somecontent") or die('Could not write to file');
// close file
fclose($fp);
?>
```

6. Now create the autodial.log file as shown below. Again, mind the file ownership and permissions.

```
-rwxr-xr-x. 1 voipnow voipnow 10780 Jun 16 13:47 /usr/local/voipnow/admin/htdocs/autodial.log
```

7. The calls will be placed by this script (replace with your email address):

```
#!/bin/bash
cp /tmp/autodial.call /var/spool/asterisk/outgoing/
CALLERID=`cat /tmp/autodial.call | grep CallerID | cut -d "<" -f2 | cut -d ">" -f1`
NEWCALLERID=$((CALLERID + 1))"
sed -i "s/${CALLERID}/${NEWCALLERID}/g" /tmp/autodial.call
sleep 10
CALLOK=`cat /usr/local/voipnow/admin/htdocs/autodial.log | grep $CALLERID`
if [ -z "$CALLOK" ]; then
    echo "Call with callerID $CALLERID failed." >> /usr/local/voipnow/admin/htdocs/autodial.log
    echo "Call with callerID $CALLERID failed." | /bin/mail -s "Failed autodial call"
    example@email.com
else
    echo "Call with callerID $CALLERID was successful." >> /usr/local/voipnow/admin/htdocs/autodial.
log
fi
```

The flow

Here is how it goes:

The script moves the autodial.call file to the "outgoing" directory.

Asterisk places a call towards the configured DID through the configured channel.

You increment the callerID value in the autodial.call file. Each call will have an incremental callerID.

The call comes back to the VoipNow server where it reaches the extension that hangs up and creates a Call Event request.

The autodial.php file inserts details about the call. Here is an example.

```
Array
(
    [CallID] => 1434455237.366
    [CallerIDNum] => 11156
    [CallerIDName] => autocal
    [CalledDID] => 2002
    [CalledExtension] => 0003*002
    [CallStatus] => CALLING
    [CallFlow] => in
    [CallerExtension] =>
    [CallAPIID] => d2zhpsga2y4wh7vvfg3
)
```

You wait for 10 seconds to make sure the call event log has time to be created and then look for the latest callerID in the log.

Based on the callerID, the script appends a message stating whether the call worked or not.

In case the call failed, you'll also receive an e-mail at the configured address.

If the flow is correct up to this point, you can set up the script in a cron to run every few minutes.

Related articles

- [Troubleshooting CallNow calls that drop](#)
- [How to transfer incoming calls to another extension voicemail](#)
- [Understanding the differences between a free and a paid channel](#)
- [Troubleshooting fax session failures caused by incorrect negotiation of T38FaxMaxDatagram parameter](#)
- [Troubleshooting calls and debug steps](#)