

Loway
r e s e a r c h

QueueMetrics
call center monitor

**Tracing outbound calls through QueueMetrics
On TrixBos**

Version 2.0

This version applies to QueueMetrics 1.4.5 and TrixBos 2.6.x

Index

Document contents	3
Revision history	3
Enabling QueueMetrics outbound on TrixBot.....	4
Enabling outgoing calls in TrixBot.....	5
Configuring QueueMetrics	9
Usage hints and tricks.....	11
For further help	12

Document contents

This document is a step-by-step guide on how to place outgoing calls that are to be traced in QueueMetrics on a TrixBox system.

Revision history

- May 22, 2007: First draft
- June 10, 2008: Complete revision, using dial-plan logic for outbound – also added Agent dialout.

Enabling QueueMetrics outbound on TrixBot

In order to make outgoing calls visible in QueueMetrics, we must set up a special procedure to do the logging. Luckily the hard part has already been written, so it is only a matter of integrating existing pieces into TrixBot.

The first thing that we have to consider is that just like we have both direct incoming calls not going through the ACD (and not seen by QueueMetrics) and inbound ACD calls (that are seen by QueueMetrics) we will have two kinds of outgoing calls: ordinary calls and calls that will be tracked. It is up to you to instruct your agents on which calls must be made for each case.

Just like incoming ACD calls are organized by queue, outgoing calls are flagged as being part of an outgoing campaign, so that later in QueueMetrics you can report on them knowing why they were made. As the outgoing campaign is a convention we rely upon, it needs not be defined anywhere in Asterisk and will be input in QueueMetrics as if it were a queue name.

So in order to place an outgoing call belonging to a campaign, our agent will have to key in:

- A special extension code that means the call is being made on an outgoing campaign (we will use the extension 8 for this example)
- A three digit code for the campaign (you can define them freely, just write down what it means), e.g. campaign 100: recalls, campaign 101: sales meetings, etc. We suggest you use a three digits code for the campaign so that they can be easily input using a numeric keyboard of a standard phone
- The other party's number that the agent is to call.

Enabling outgoing calls in TrixBox

First of all, find the file *extensions_queuemetrics.conf* in your QueueMetrics instance (you can find it under the system root as *WEB-INF/mysql-utils/extensions-examples* – to find the location of the system root, just look at the licence page in QM) and copy that to */etc/asterisk*.

Include it in your main dial-plan by editing the *extensions_custom.conf* file and adding the line:

```
#include extensions_queuemetrics.conf
```

Now edit the *extensions_queuemetrics.conf* file and look for a piece of dial-plan that looks like the following one:

```
[queuedial]
; this piece of dialplan is just a calling hook into the [qm-
queuedial] context that actually does the
; outbound dialing - replace as needed - just fill in the same
variables.
exten => _XXX.,1,Set(QDIALER_QUEUE=q-${EXTEN:0:3})
exten => _XXX.,n,Set(QDIALER_NUMBER=${EXTEN:3})
exten => _XXX.,n,Set(QDIALER_AGENT=Agent/${CALLERID(num)})
exten => _XXX.,n,Set(QDIALER_CHANNEL=Zap/g0/${QDIALER_NUMBER})
exten => _XXX.,n,Set(QueueName=${QDIALER_QUEUE})
exten => _XXX.,n,MixMonitor(Q-${QDIALER_QUEUE}-${UNIQUEID}.WAV|b|)
exten => _XXX.,n,Goto(qm-queuedial,s,1)
```

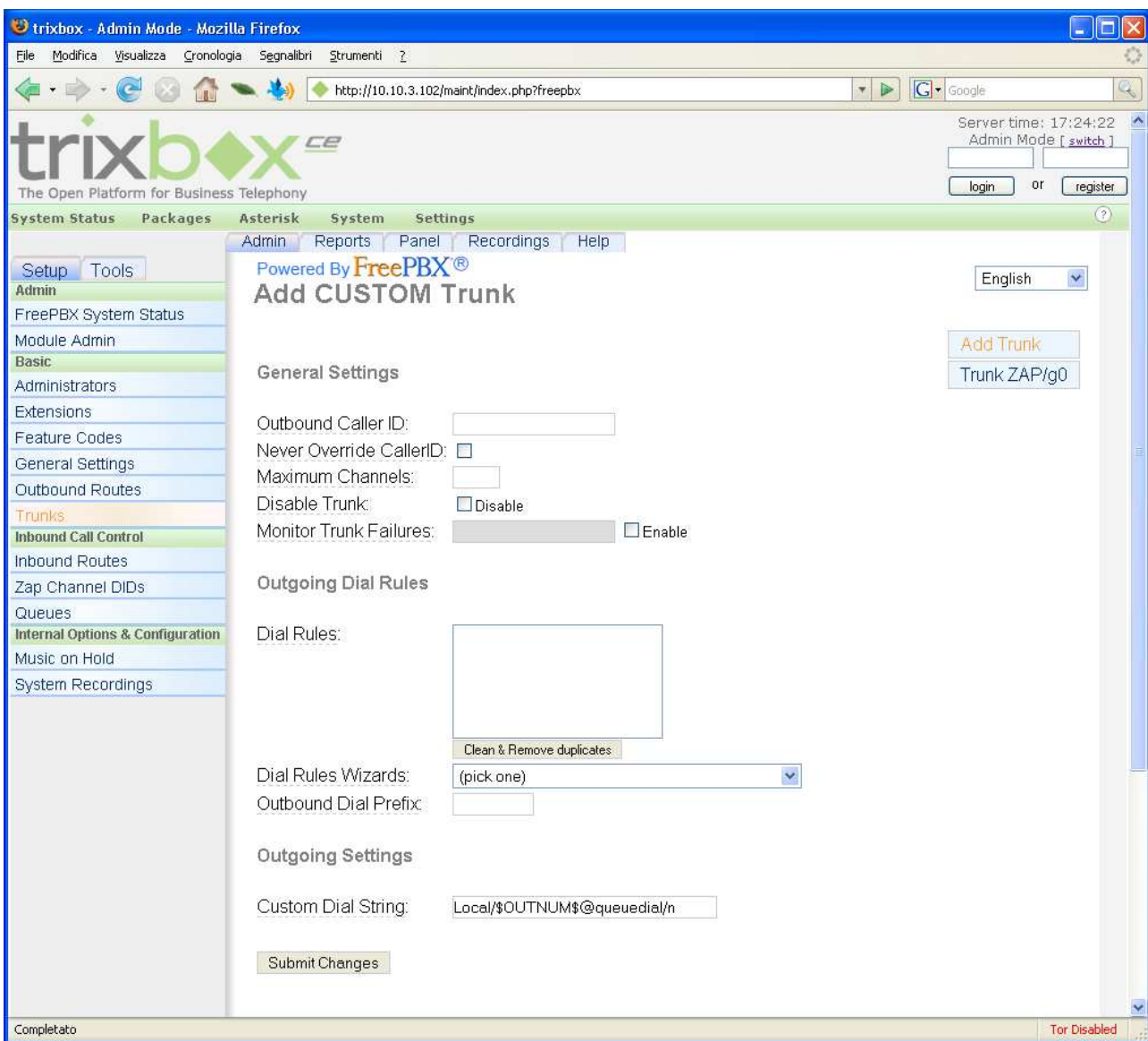
This basically says that when an outgoing call is placed...

- the first three digits are stripped and used as the Campaign number
- the QueueMetrics name for the campaign will be q-XXX
- the other digits are composed on channel "Zap/g0/" – you may need to change this to suit your trunks, what matters is that the content of the QDIALER_CHANNEL variable result in a valid channel.
- all activity is logged using *Agent/XXX* where XXX is the caller's extension number. This may or may not match logged in agents – see section *Tracking agents or extensions?* on page 11 to check if this is what you want or not.

The calls are also all recorded in a format that QueueMetrics is able to retrieve - if you do not want all calls to be recorded, just comment out the line with the MixMonitor command (just start the line with a semicolon).

Then we will have to create a trunk in FreePBX to send the calls to our custom code above.

Click on Trunks, select "Add Custom Trunk", then leave everything blank but the field "Custom dial string" where you will enter "Local/\$OUTNUM\$@queuedial/n". See the screen below for reference.



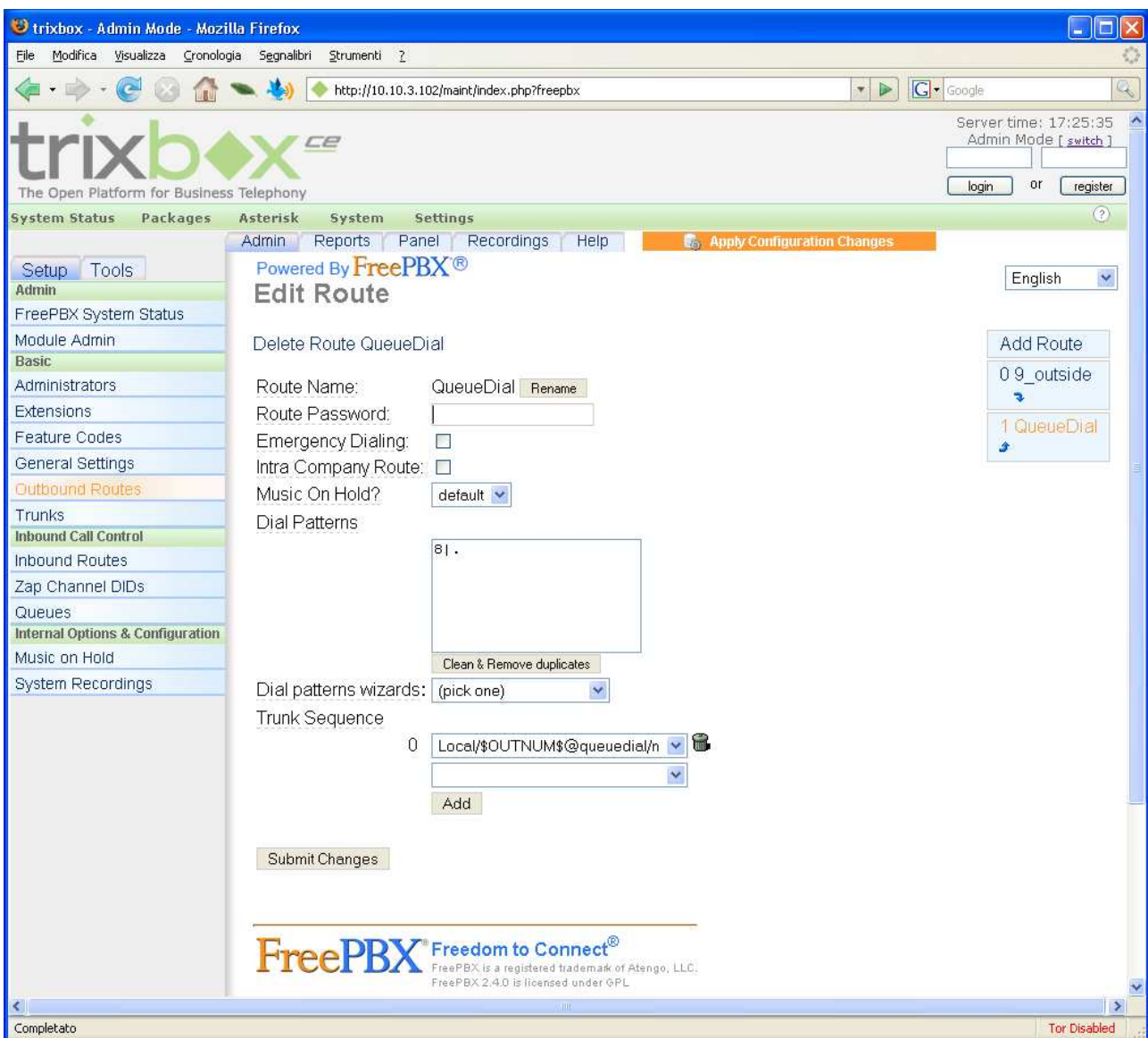
Now we will have to use that trunk as an outgoing route; to do this, click on "Outbound routes", then enter a new route like the following:

- Route name: QueueDial

- Dial patterns: 8|.
- Trunk sequence: select " Local/\$OUTNUM\$@queuedial/n"

This basically means that when your agents enter a number that starts with 8, the call is passed to the trunk we just defined, that in turn calls our AGI script.

Please pay attention to adding "/n" at the end of the Local string, otherwise it will not properly work. The following screenshot details how the result is expected to look like:



At this point apply the configuration changes in FreePBX by clicking on the top orange label and try dialing 8123456 from one of the extensions; this tries dialing the PSTN

number 456 on campaign 123. Even if you get an error (and this is very much likely, as 456 will not be a valid PSTN number) if you look at the end of the file */var/log/asterisk/queue_log* you will see that a few lines have been appended, e.g.:

```
1179399430 | 1179399430.13 | q-123 | NONE | ENTERQUEUE | - | 456  
1179399430 | 1179399430.13 | q-123 | NONE | ABANDON | 1 | 1 | 0
```

This means everything is in place and working. If now you run a successful call through it, the log will look something like:

```
1179822810 | 1179822810.22 | q-123 | NONE | ENTERQUEUE | - | 0332320550  
1179822813 | 1179822810.22 | q-123 | Agent/101 | CONNECT | 3 |  
1179822823 | 1179822810.22 | q-123 | Agent/101 | COMPLETEAGENT | 3 | 10
```

The double line with CONNECT is normal and is used to provide QueueMetrics real-time monitor a "hint" that a call is in progress.

Configuring QueueMetrics

We now just have to configure QueueMetrics to report on the new campaign "123". Log on to QueueMetrics as an administrator, click on "Edit queues" and create a new queue like the following:

- Alias "Outbound 123"
- Queues: "q-123" (that is, you prepend "q-" to the campaign code)
- Call flow: Outbound calls

Save the queue.

The result should look like the following screenshot:

The screenshot shows the 'Loway QueueMetrics version 1.3.5 - Mozilla Firefox' browser window. The address bar shows the URL 'http://10.10.3.100:8080/queuemetrics/qm_admin/configura_code.jsp'. The main content area displays a table of queues and a form to add a new queue.

Alias	Queue(s)	Wrap-up	Ann.	Key			
00 ALL	q1, q2	0 s.	0 s.		Edit	Agents	Delete
DPS	queue-dps	15 s.	0 s.		Edit	Agents	Delete
Queue 1	q1	0 s.	0 s.		Edit	Agents	Delete
Queue 2 OUT	q2	0 s.	0 s.		Edit	Agents	Delete
Sample Queue	sample	0 s.	0 s.		Edit	Agents	Delete

Legend: 📞 : inbound queue - 📞 : outbound queue

Add new queue

Alias:

Queue(s):

Wrap-up time (sec.):

Announcement (sec.):

Visibility key:

Call flow:

Attention levels

	Yellow alarm	Red alarm
Number of calls in queue:	<input type="text"/>	<input type="text"/>
Number of agents on call:	<input type="text"/>	<input type="text"/>
Number of agents waiting:	<input type="text"/>	<input type="text"/>
Number of agents paused:	<input type="text"/>	<input type="text"/>
Call wait duration:	<input type="text"/>	<input type="text"/>
Call talking duration:	<input type="text"/>	<input type="text"/>

Completato

Now if you go back to the home page, select "Outbound 123" from the drop down menu and click on "Today" , you should see at least one lost call - this means everything worked.

From now on, you can treat outbound campaigns as if they were incoming ACD queues in QueueMetrics. The only current limitation is that you will not see wait and call times correctly on the real-time screen, as such data is currently returned by Asterisk only upon call completion.

Tracking agents or extensions?

The simplest example provided here will track outgoing calls as *Agent/XXX*, where XXX is the number of the calling extension. This is very simple to implement and is okay for call centers that only do outbound.

If you use a setup where you do mixed inbound and outbound that is set up to use “true” Asterisk agents – as for example described in our basic TrixBox tutorial – you may want to make sure that:

- Only logged-on agents can place outbound calls (in order to make sure that call-center activity is done only by agents that have an explicit log-on)
- As Asterisk knows who the agent sitting at a given extension is, we want to log the call using the Agent code, not the extension – so your agents can work on shifts using the same extension, but still you know who made which call.

We include an additional dial plan script that is called *queuedial-loggedon* that does exactly this. If an agent that is not logged on tries to place an outbound call, they get the error message “Agent logoff” to remind them they need to log on first.

In order to implement this, you must do two things differently:

1. Edit the *queuedial-loggedon* context in order to set the dial-out channel *QDIALER_CHANNEL* right, as explained for the *queuedial* context
2. When you define the custom trunk, use *Local/\$OUTNUM\$@queuedial-loggedon/n* as the channel identifier.

There is nothing else to be done.

Usage hints and tricks

- As the telephone extension does not know that the number we are dialing will be any different from a standard extension, you can use a web interface to do the calling for you; this makes it pretty easy to build an unified, web-based calling interface and requires just a click from your agents
- If you are going to record a big number of calls, you should very likely use wav49 format (that is, Microsoft’s GSM) to do the recording and save 75% on disk space used.

For further help

The official QueueMetrics website is located at <http://queuemetrics.com> . On the downloads page, a guide to installing and integrating QueueMetrics for TrixBox systems is available.

You can get help on QueueMetrics user forums, located at <http://forum.queuemetrics.com>

A number of freely available Asterisk PBX recipes, mostly targeted to call centers, can be found on <http://astrecipes.net>