Loway

research

Turning TrixBox into a small call-center with QueueMetrics

AUTHOR:	LOWAY
VERSION:	2.0
DATE:	JULY 26, 2007

Loway di Lorenzo Emilitri
Via Ligornetto 17a - 6854 San Pietro - Switzerland

Index

Turning TrixBox into a call-center solution

If you are reading this, you probably already know that using TrixBox (TB) or Asterisk@Home (AAH) you can install a fully featured Asterisk-based PBX in a matter of minutes. This PBX is very good for most users as it is preconfigured to handle the most common scenarios one can find in a personal environment or in a small to medium sized office.

The typical usage patters of a small call center are a bit different from the ones you are likely to find in a classical PBX, because:

- Agents spend almost their whole working day available to answer the phone (as opposed to standard PBX users who use the phone occasionally while doing other work); handling and answering calls for them has to be made as easy as possible
- A call center is usually a high-density commercial enterprise; therefore it has to be run and monitored using tools that are able to quickly see how things are going, identify bottlenecks and address them.

In our opinion, running a successful call center is more a state of mind than a given set of telephone gear. What makes a difference is not the number of extensions, queues or agents you have; is a mindset where you consider that the customers calling in are *actually very important* and you do your best to serve them well within given budget limits.

Maybe you just run a small computer-repair shop and have a couple of lines coming in. What is the cost of having people wait or call multiple times because they cannot talk to anybody within a reasonable time? What will your clients think about you? On the contrary, what will your clients think of your customer service if it always answers on the very first ring? And how do you know if your technicians actually answer the phone when the calls come in or wait five minutes because they are doing other things? These are the questions you should ask. If you follow the guidelines in this document, you'll find an easy way to start answering to questions like these.

Running a call-center, therefore, is not a matter of having multiple PRIs or special hardware. You will not even need a separate box from your main PBX running TB. You will only need some software and a bit of configuration to set it up correctly.

Call centers 101: the very basics

Before we start building a small call center, we have to focus a bit on the terminology:

- A *campaign* is a set of calls that belong to the same scope, e.g. your technical support versus commercial support line are different campaigns, though they may be
- An *inbound* campaign is devoted to answering people calling in, while an *outbound* campaign is made up by agents dialing out. Call centers often mix inbound and outbound activities in order to optimize the use of personnel.

- A *queue* is the physical implementation of an inbound campaign. The queue receives calls and pipes them to the available agents according to a predefined logic (usually, FIFO for the calls and round-robin for the agents). In call center terminology, this functionality is often referred to as the ACD (Automated Call Distribution).
- An *agent* is a person working at a call center. The agent is different from a casual user as an agent logs in and out, in order to tell the system when he is available or not. In this way, the ACD searching logic minimizes agent searching time, as it almost never has to ring up an agent who is not available. An agent can be working on one or more queues: whenever he is available, all calls coming in to any queue he's working on will be piped to him.

In this tutorial, we will learn how to create an inbound queue and proper agents to handle it.

Prerequisites

To follow this guide, you will need an already-installed, reasonably modern TB instance. It may be your home or office PBX. You can follow this guide completely while having your PBX running, so there is no need for a prolonged downtime. You will need at least a couple of telephones to test your setup, and a land line you can use.

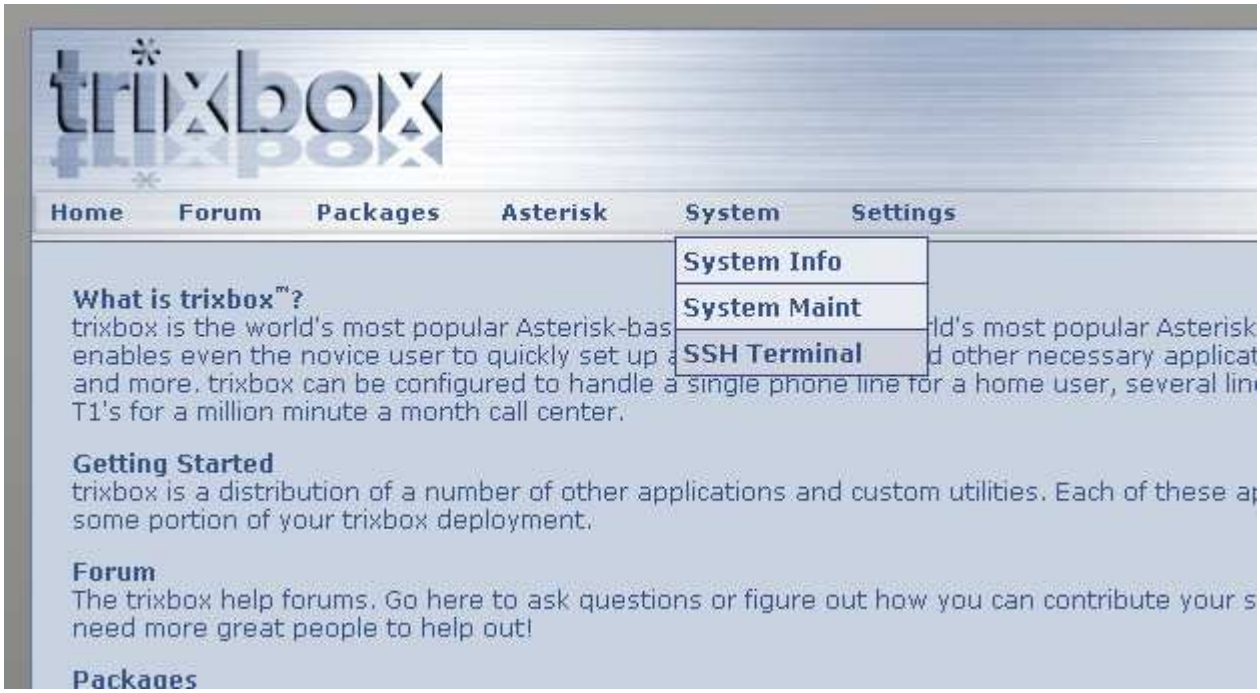
You should be basically familiar on how to use TB as a basic PBX: creating extensions, connecting to external lines and such things.

Installing QueueMetrics

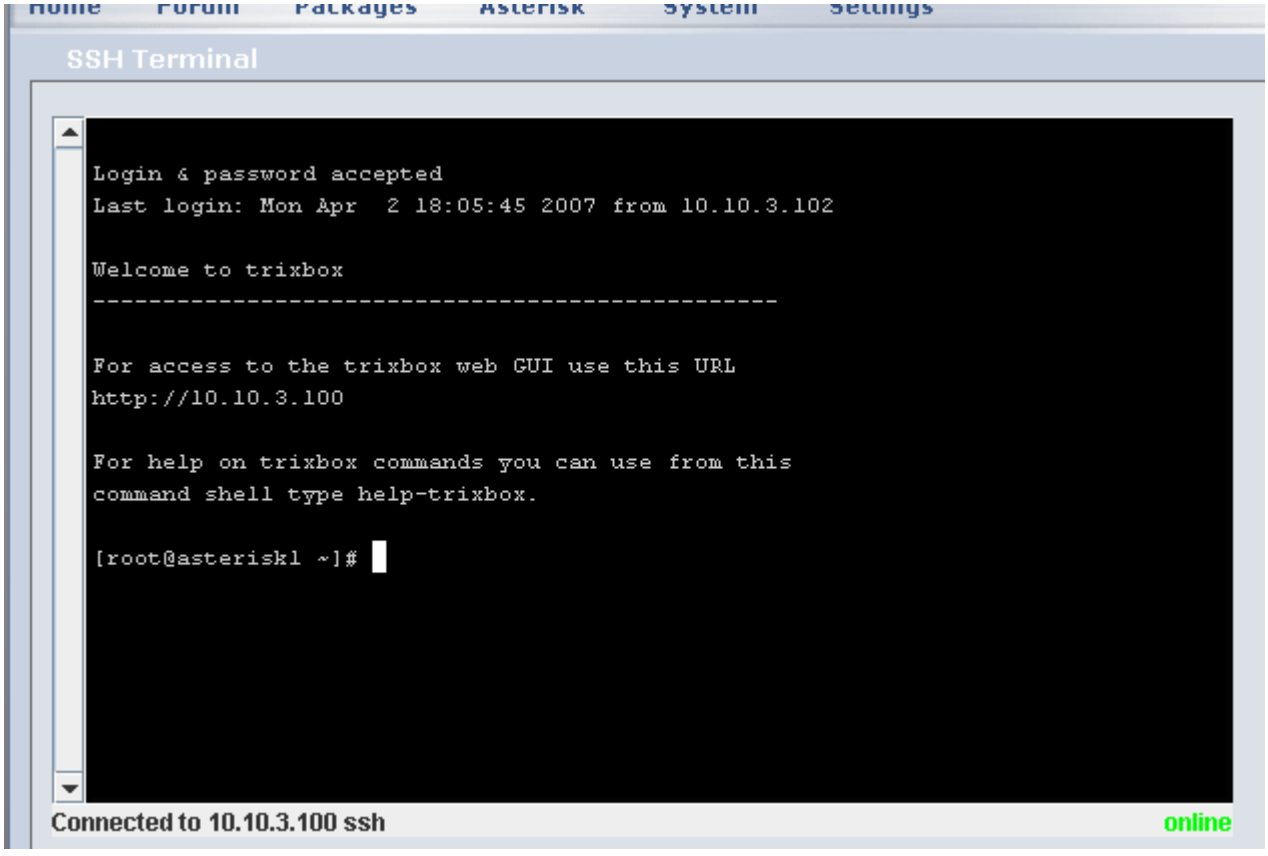
QueueMetrics is a full-fledged call center monitoring solution – see http://queuemetrics.com . It is an industry-proven, commercial product that is available free of charge to smaller call centers, home users and SOHO's and is used in hundreds of call center worldwide, including installations with hundreds of agents on-line.

Log in to your TB system using your web browser and the standard admin credentials (they are login: "maint" password: "password" if you have not changed them manually). You can do this by going to the URL: http://myserver/maint .

Select the label "SSH terminal" under the "System" flag, as shown in the following picture:



A terminal window will open (it may take a while waiting for the Java client to load). Log in as “root” with password “password”. You should see a terminal like in the following picture:



When you are logged in, type the following commands:

```
wget -P /etc/yum.repos.d http://yum.loway.ch/loway.repo

yum install queuemetrics
```

The *yum* command will download QueueMetrics and all of its dependences and install it on your system. This may take a while, depending on your internet connection speed. When asked to confirm the installation, type "y" to confirm.

After the installation is done, you have to install the sample MySQL database that will be used to initialize the system by executing the following commands:

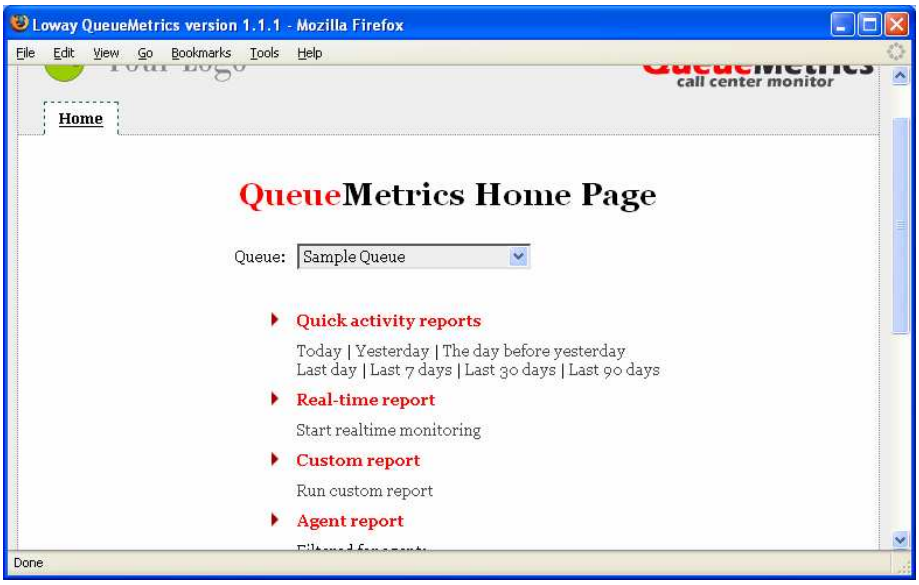
```
cd /usr/local/queuemetrics/webapps/queuemetrics-1.3.3/WEB-
INF/README

./installDb.sh
```

Note that the exact directory to use will depend on the QueueMetrics version being installed and is displayed on the last page of output that yum produces. Follow the on-screen instructions (it is a matter of typing in a couple of passwords as detailed by the *installDb* utility itself) and the database will be created. For your convenience, the default MySQL password for TB is "passw0rd" (yes that's a zero).

To test that everything is okay, you'll have to point your browser to the address http://myserver:8080/queuemetrics and you should see a screen like the following one.

If you see this screen, you know that QueueMetrics is working fine. As you'll be curious to check it out, you can login immediately with the credentials "demoadmin" password "demo".

You may notice that the first time you load up a new page, it may take a while to display it, depending on your TB box CPU speed; this is normal, as QueueMetrics uses Java Server Pages technology and such pages are internally compiled the first time they are accessed.



This is how the first page looks like. Feel free to look around a bit before logging off.

You may now also close the SSH window as we won't need it anymore for this tutorial.

Setting up TrixBox

Setting up TrixBox is quite easy – we will assume the following scenario during the rest of this tutorial:

- You have a few SIP telephones configured in the range 100-110
- You have one inbound queue as extension 200 called “Support”
- You have two agents that can work on this queue when they are logged on, and their logon codes are 300 (for Alice) and 301 (for Bob). Those codes need not be different from the numbers used for extensions, but we will assume so.

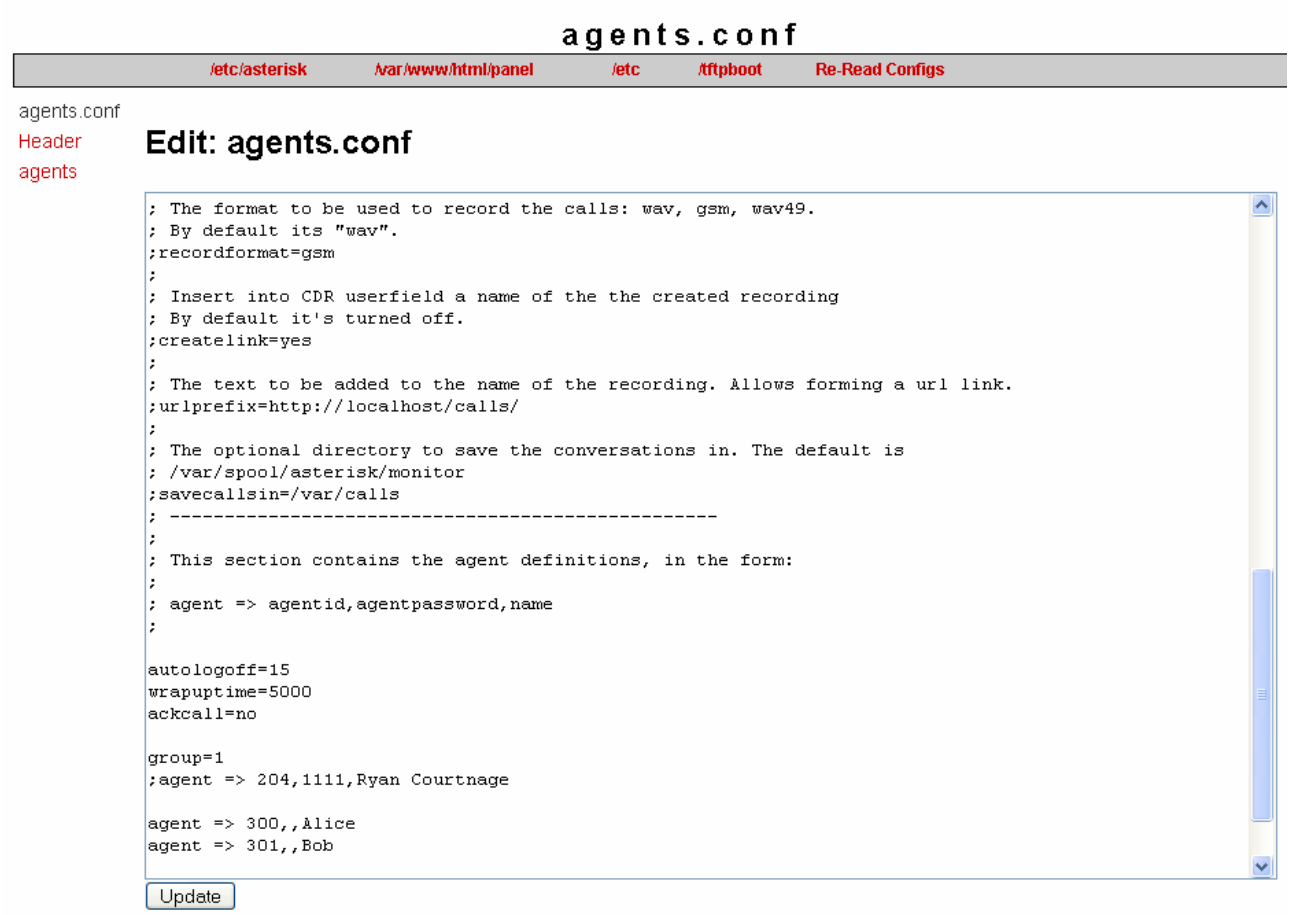
We will set up agents so that they can be used to login from either TrixBox (by calling a special extension) or right through QueueMetrics.

Creating agents

The creation of agents is the only thing that cannot be fully automated through the current version of FreePBX; on the other side this is very easy and is done only occasionally, while the association of agents to one or more queues can then be done through the GUI.

From the TrixBox main page, you select *Asterisk -> Config edit* as shown:

After this, you select the file Agents.conf and edit it as shown in the following page:



What you do is:

- You first comment out the default entry "*agent => 204,1111,Ryan Courtnage*" by pre-pending it with a semicolon
- You add an entry telling the system that we have an agent called Alice with code 300, by entering a line "agent => 300,,Alice"
- You add an entry telling the system that we have an agent called Bob with code 301, by entering a line "agent => 301,,Bob"

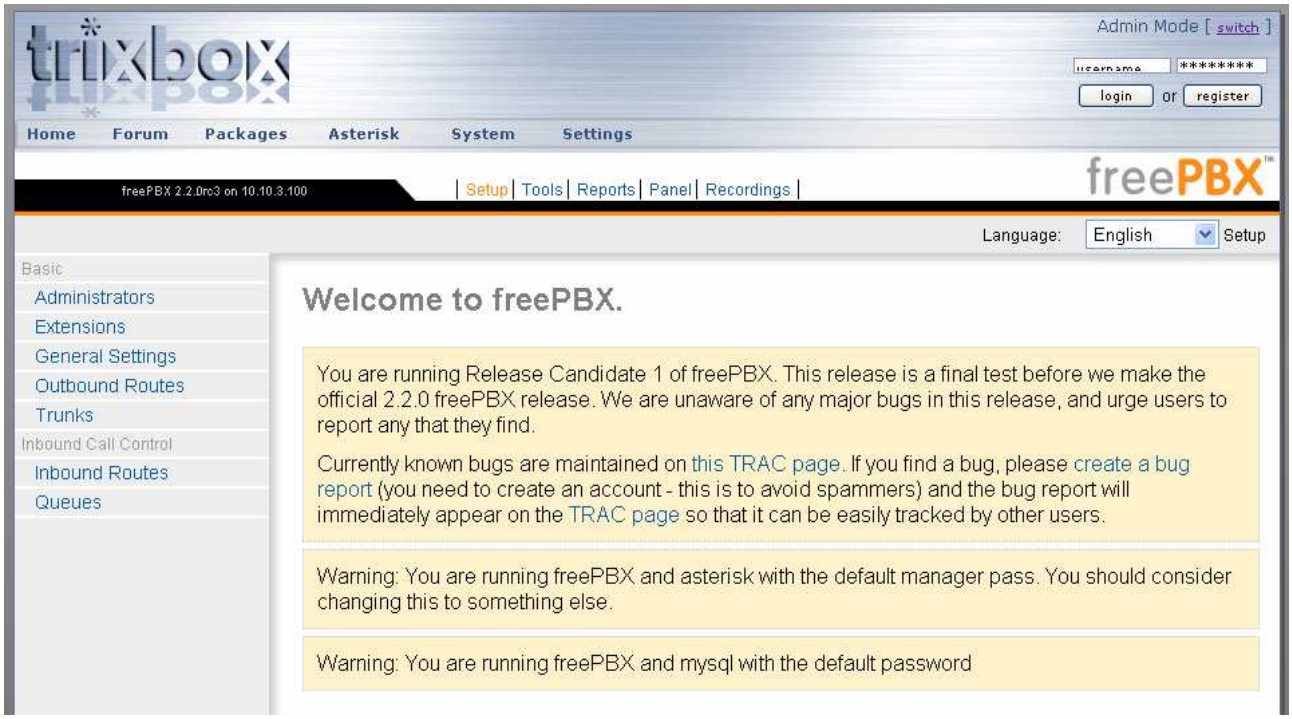
Once you're done, press Update to save.

Whenever you will need to add other agents to this system you will simply repeat this procedure. Do not forget to click on the "Re-Read configs" once you're done to tell Asterisk to load the new configuration.

Configuring a queue

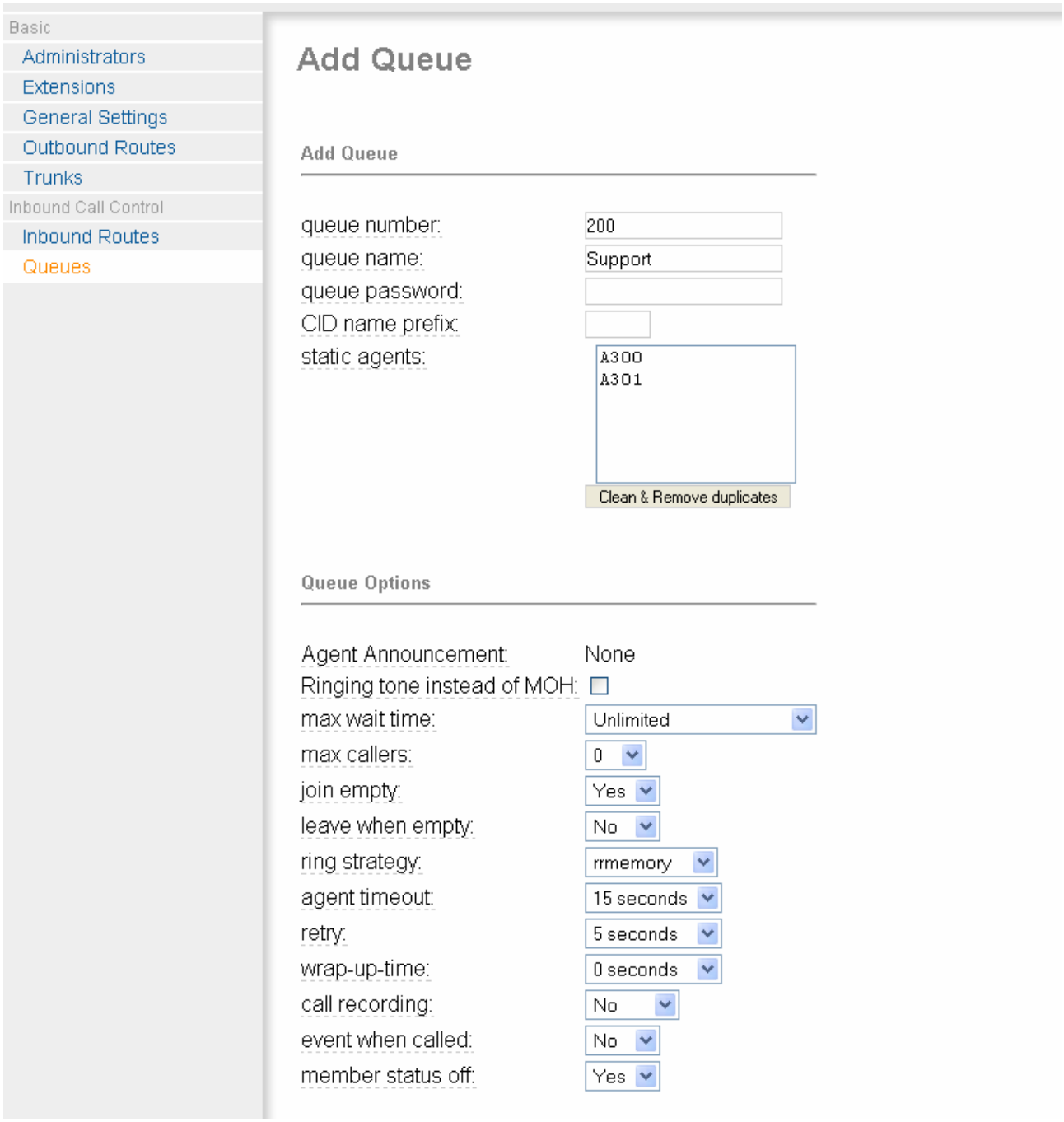
To add a queue to your system, go back to the main menu; select *Asterisk -> FreePBX* to enter the main configuration utility.

Once you're there, click on the "Setup" and the system should look like the following picture:



If you do not see the voice "Queues" under "Inbound call control", this means that the queue management module is not installed. To install it, you click on "Tools", then "Module Admin" from the right column, and select "Queues"; follow the install procedure and the Queues module will be installed in a few seconds.

To create a our queue now, we click on "Queues" and fill in the wizard as follows:



You will see that a queue is highly configurable; but for the moment we make ourselves content of entering the following data:

- Queue number: *200*
- Queue name: *Support*
- Static agents: *A300* and *A301*, each on a line by itself. (This means that Agent/300 and Agent/301 will be referenced right from the agents.conf file we just edited).

- Ring strategy: *rrmemory* (This is the basic call center setup where the ACD system tries one agent after the other until someone answers)

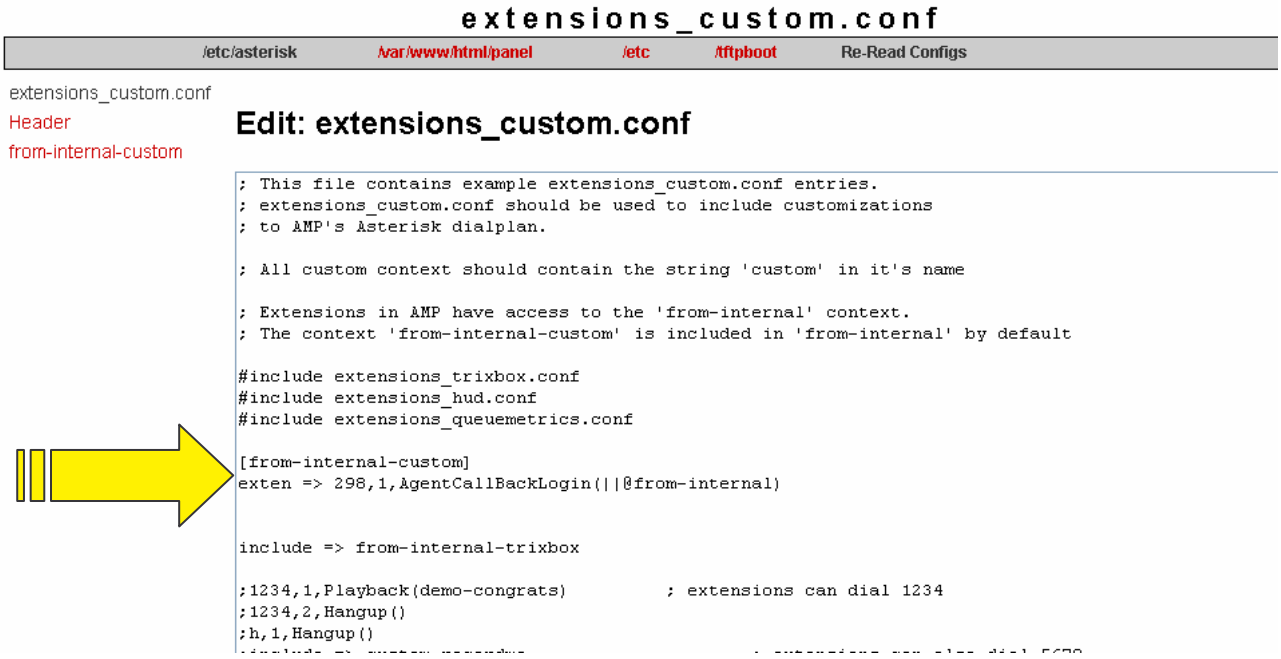
When you are done, click on the "Submit changes" button the red line and have FreePBX update the system.

Creating log-on point for your agents

You can have your agents log in (that is, telling Asterisk they are ready to take calls) either by calling a specified extension or by using QueueMetrics' own Agent Login page. You can even configure both and use them alternatively.

Logging on from an extension

To create an extension for your agents to log on, go to the extensions editor like you did for creating the agents and add the following line:



It must be exactly below the line [from-internal-custom] and it must say:

```
exten => 298,1,AgentCallBackLogin(||@from-internal)
```

After you're done, press "Update" to save your changes and then click on "Re-read configs".

This means that when somebody dials "298", they get the agent log on menu; in order to log on, you must enter your agent code, followed by pound, followed by the extension you're sitting at. In order to log off, you simply dial the same extension, enter the agent number and then pound twice. The voice will tell you if the agent is logged on or off at the end of this procedure.

Logging on from QueueMetrics

If you want to use QueueMetrics for your agents to log on, log off, go to pause, set call status and whatever else, you must install a custom piece of dialplan that comes with QueueMetrics itself and makes it possible to enable all those features.

To do this, you must log in to the system (just like you did to install QueueMetrics) and run the following command:

```
cp /usr/local/queuemetrics/webapps/queuemetrics-1.4.0/WEB-
INF/mysql-utils/extensions-examples/extensions_queuemetrics.conf
/etc/asterisk
```

This copies the piece of dialplan right from the current version of QueueMetrics to the Asterisk configuration directory.

As FreePBX uses the a default context for extensions called from-internal, we must edit the file we just copied to tell QueueMetrics to use that context; so edit the file extensions_queuemetrics.conf we just copied:

```
vi /etc/asterisk/extensions_queuemetrics.conf
```

In the lines under “exten => 20” replace the string @sip with @from-internal.

Then edit the file *configuration.properties* to tell QueueMetrics to access this Asterisk box through the manager interface:

```
vi /usr/local/queuemetrics/webapps/queuemetrics-1.4.0/WEB-
INF/configuration.properties
```

Find a line that says:

```
callfile.dir=/var/spool/asterisk/outgoing
```

and change it to

```
callfile.dir=tcp:admin:amp111@127.0.0.1
```

Find a line that says

```
cluster.servers=aleph|trix
```

and change it to

```
cluster.servers= trix
```

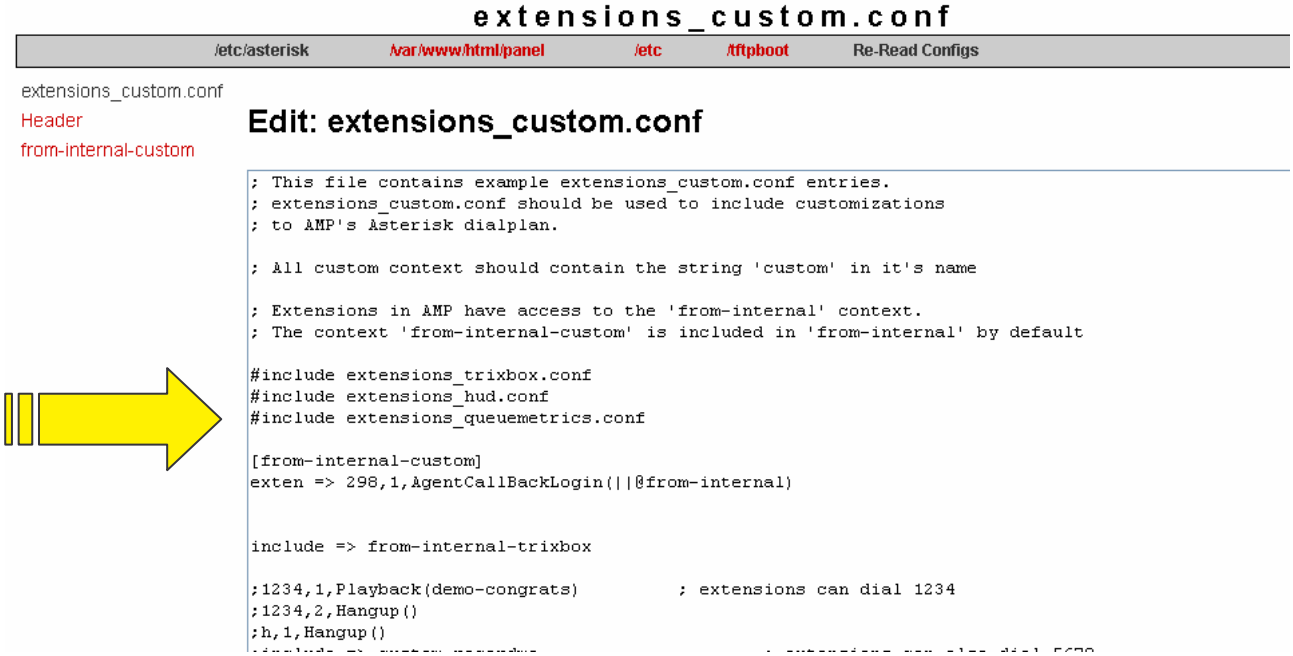
Find a line that says

```
cluster.trix.manager=tcp:admin:amp111@10.10.3.100
```

and change it to

```
cluster.trix.manager=tcp:admin:amp111@127.0.0.1
```

In order for the new configuration to be picked up, you then edit the extensions_custom file in order to include the following line:



Just below the last #include line, you add:

```
#include extensions_queuemetrics.conf
```

After you're done, press "Update" to save your changes and then click on "Re-read configs".

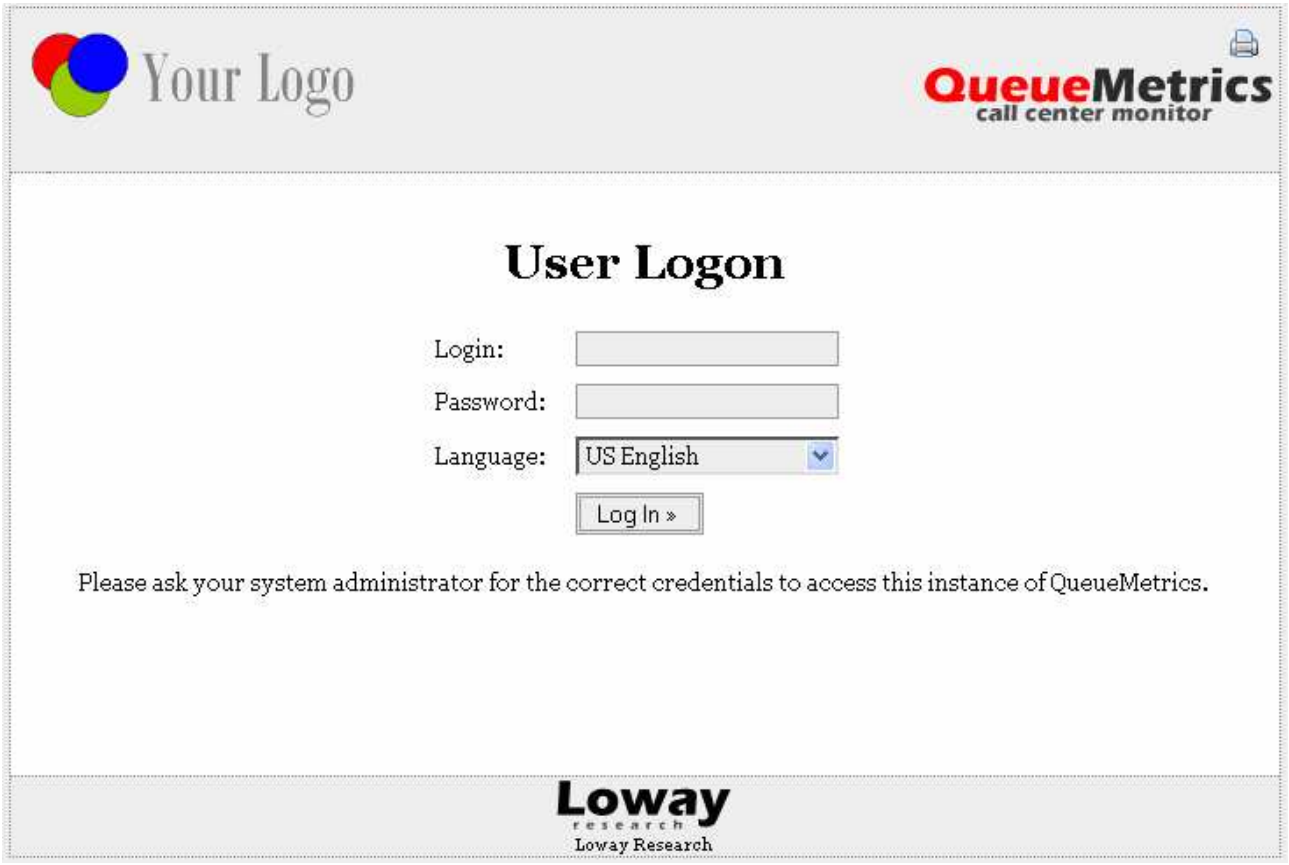
Testing your queue

Now you can test your queue. From one of the telephones, have an agent log in as agent/300, then call the number 200 from a different telephone. The first phone should ring and you can put a call through. Congratulations: you configured your first queue successfully!

Make sure you place a couple of calls, so that you can later find some data for QueueMetrics to analyze.

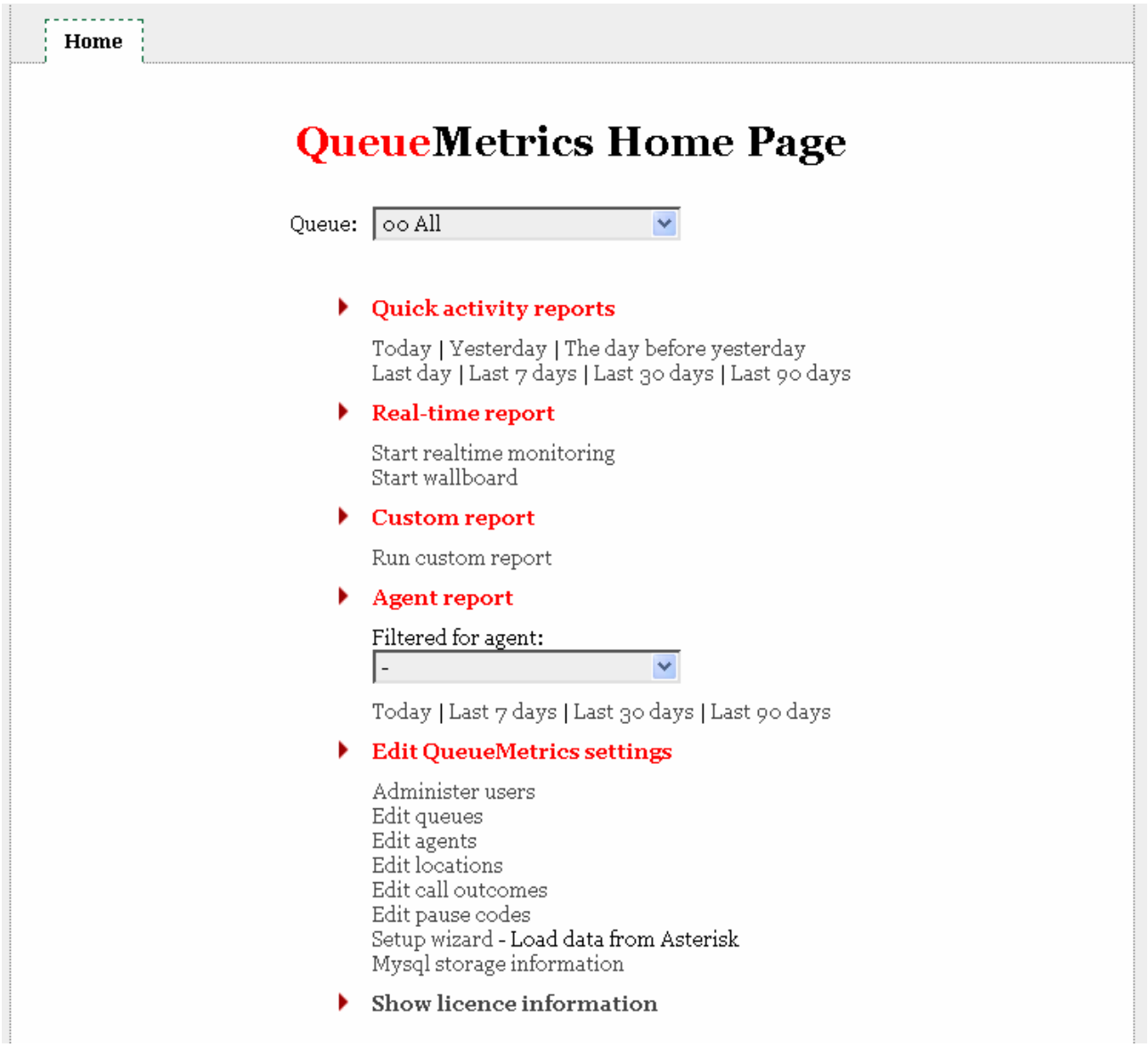
Configuring the agents and the queue in QueueMetrics

Go to QueueMetrics; it is usually installed at http://myserver:8080/queuemetrics

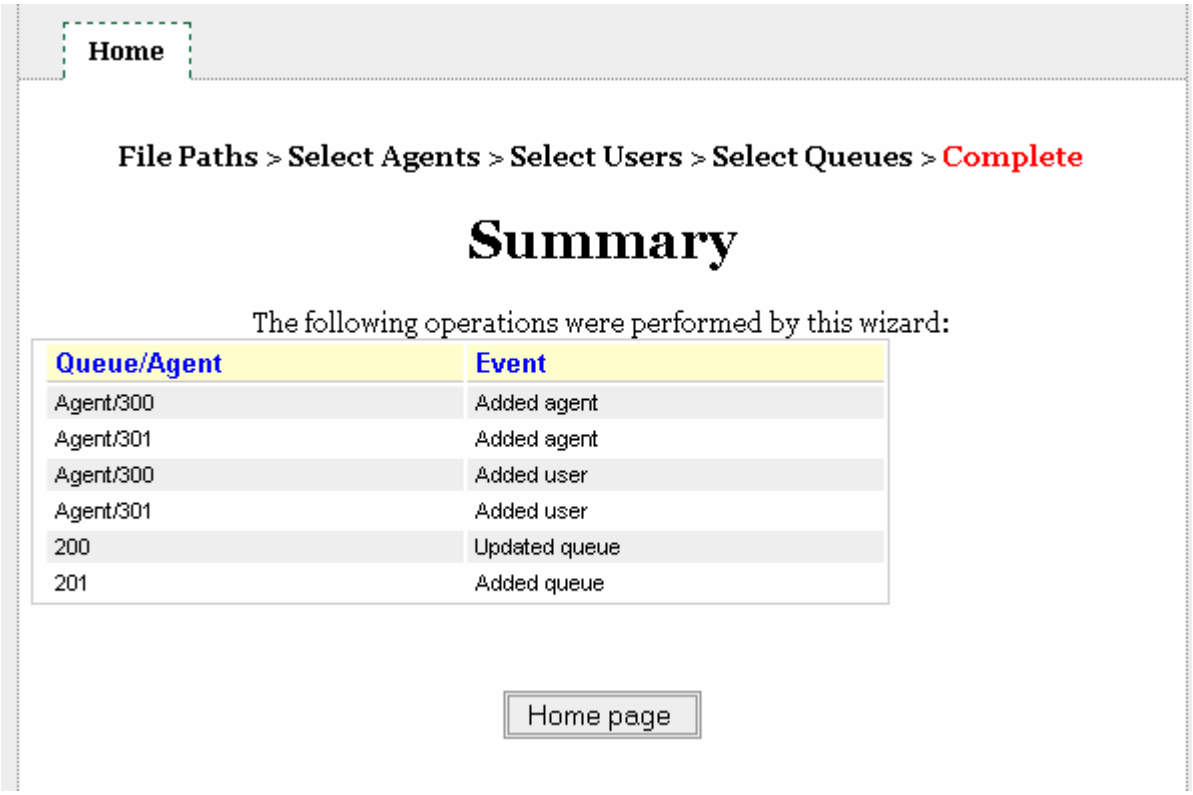


Login as “demoadmin” password “demo”.

You should see a screen like the following one:



You simply click on “Setup wizard”, then confirm all of the steps. All agents and queues that you created manually, or changed from the last time you run the wizard, will be imported automatically. At the end, you get a screen like the following one:



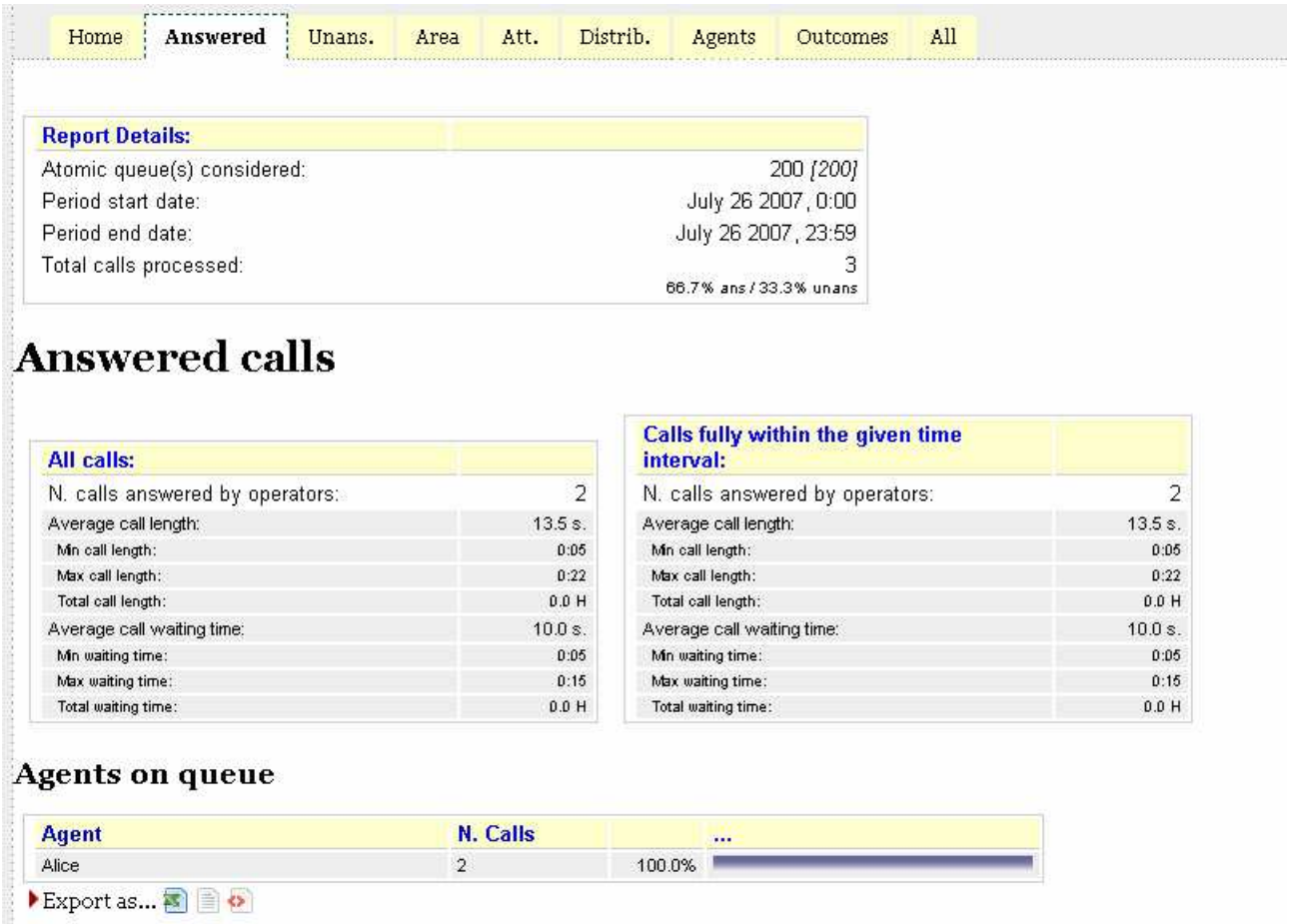
That's all there is to do. QueueMetrics picked up the configuration from Asterisk and updated it.

If you want your agents to log on to QueueMetrics, you must set a password for each of them. You can do this by clicking on "Administer users", then select the user you want to enable and enter a password, as shown in the following figure:

Running your first report

If you tried placing a call to test your system, then you can run your first report by selecting the "Home" tab, then selecting queue 200 from the drop-down queue selection box and clicking on "Today".

If everything went well, you should see a report like the following one:



As you can see, QueueMetrics computes literally hundreds of values; while it's usually pretty easy to understand the ones you're interested in, you should really have a look at the QueueMetrics user manual for a detailed description of the results.

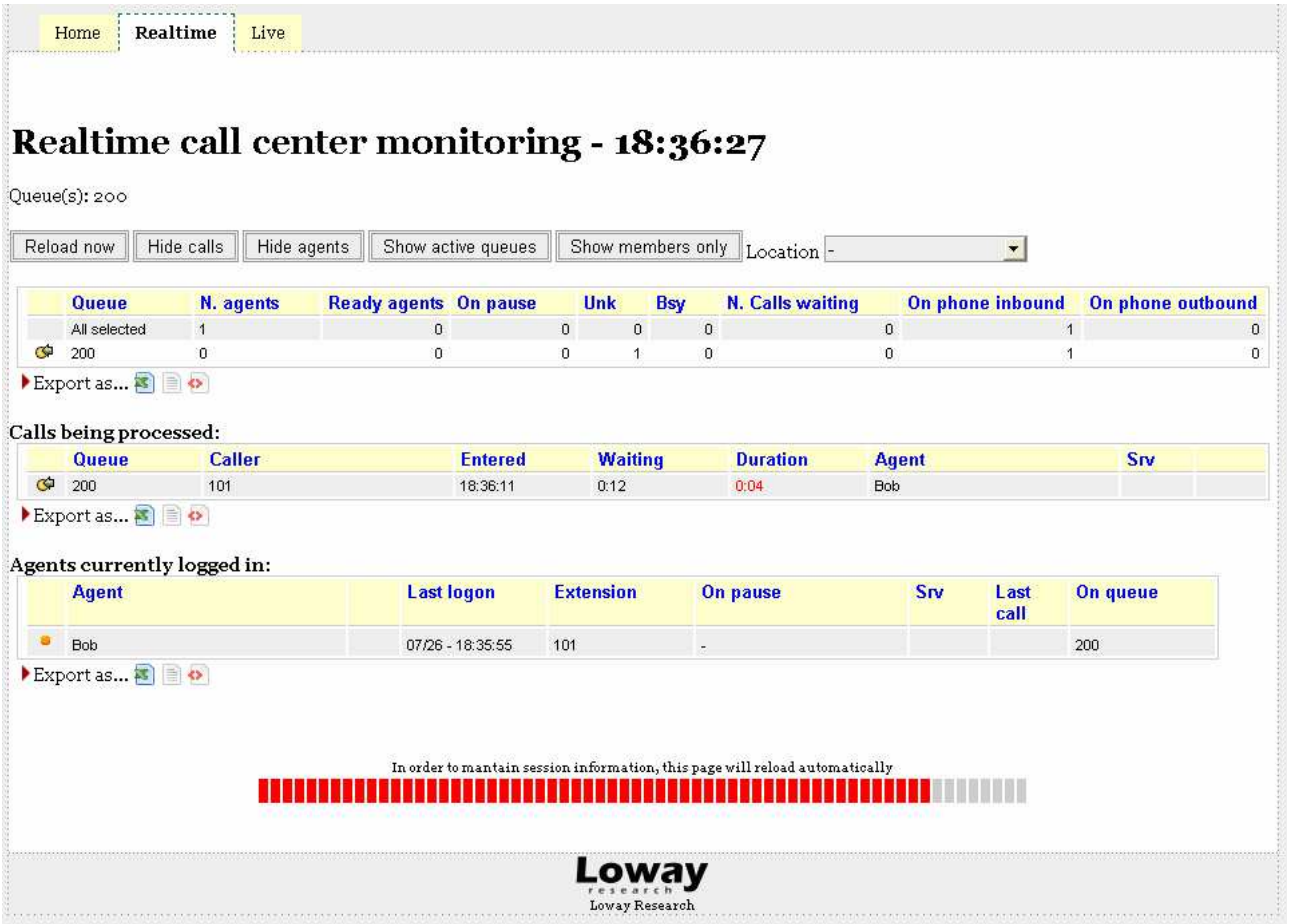
As a rough sketch, consider that

- The "Answered" tab gives statistics about the calls that were answered correctly: how any were handled, who handled them, average duration and average wait times
- The "Detail" button on the bottom of the "Answered" page gives a call-by-call activity detail
- The "Unanswered" tab gives information on calls that were lost, i.e. the user disconnected before talking to a living person

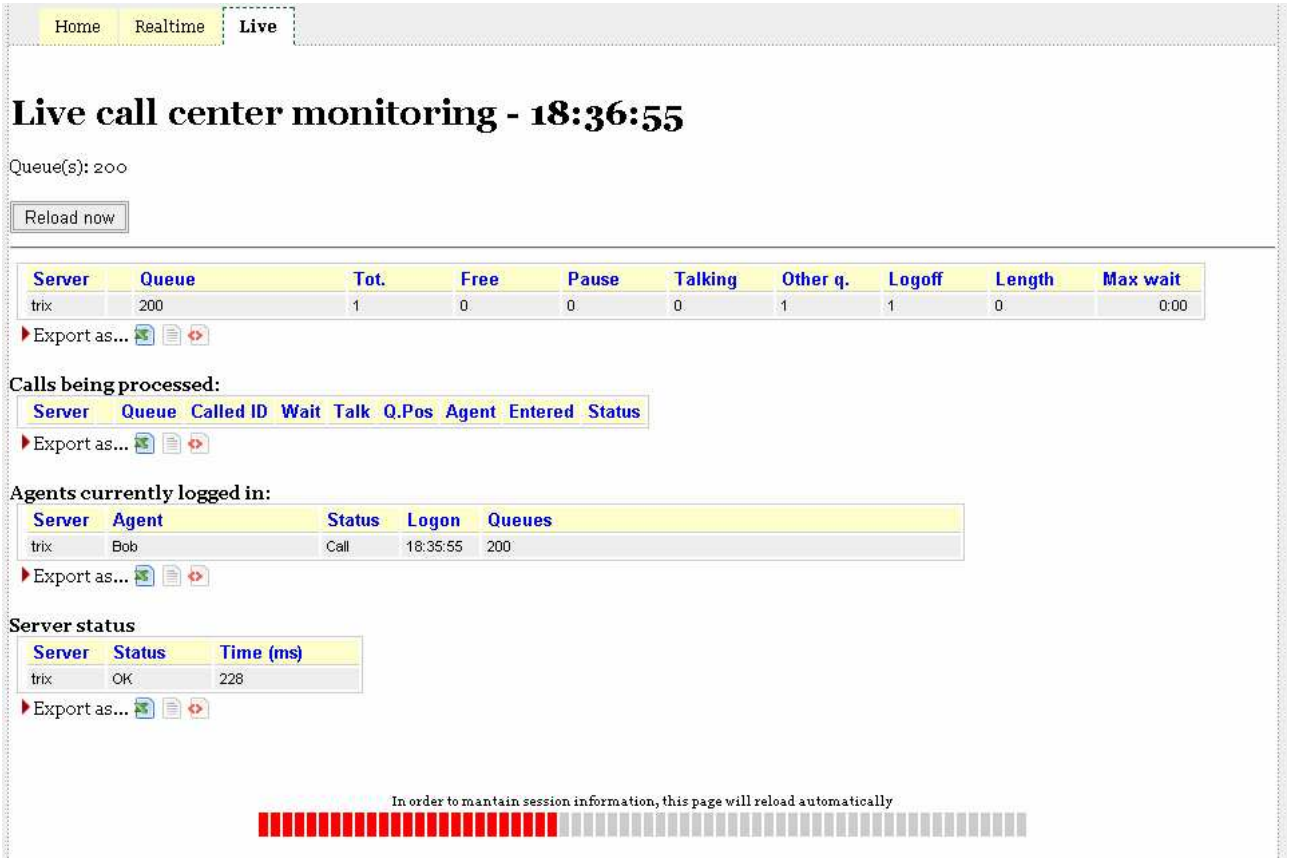
- The “Detail” button on the bottom of the “Unanswered” page gives a call-by-call detail of lost calls
- The “Distrib” tab gives a daily, hourly and day-of-week breakdown of taken and lost calls
- The “Agents” tab gives detail of agent sessions and calls handled during those sessions
- The “Detail” button on the bottom of the “Agents” page gives a session-by-session detail of agent activity
- The “All” tab is a convenient way to have all the stats on a single page so that it can be saved or printed for future reference.

Running live monitoring

A supervisor can easily run live monitoring in order to see in real time what is going on with the call-center. To do this, click on the “Home” folder, then select the queue you are interested in and click on “Start real-time monitoring”. You will see the current call center situation of each and every queue, agents available and unavailable, calls being taken and waiting on the queue, with a screenshot similar to the following picture:



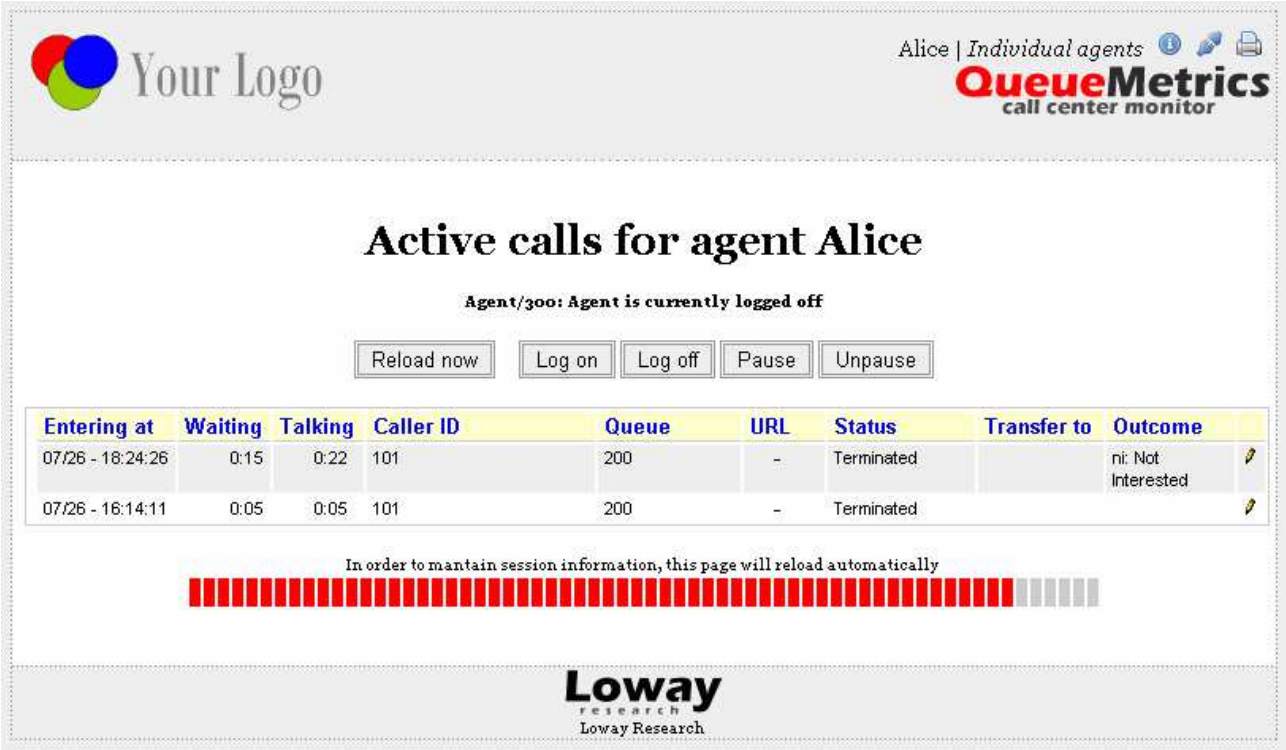
If you then click on the “Live” tab, you will see a real-time report computed directly from querying the Asterisk manager interface, as opposed to computing it from the recorded queue events.



Using the agent's page

If you previously enabled agents for log-on by setting a password, you can have them log in by entering “agent/XXX” as their user name (where XXX is the agent code as defined in *agents.conf*) and the password you set.

They will be lead to a page that shows the latest calls they have handled and a few buttons that let them log on, log off, go to pause and stop the current pause.



They can also set pause status codes and call outcomes out of a user-defined list provided by the QueueMetrics administrators.

Common problems and solutions

Avoiding queue_log file rotation

With a standard TB/AAH install, the *queue_log* file is rotated daily or weekly along with the other Asterisk logs found in */var/log/asterisk*. The queue_log file contains essential information on how the call-center is going that is being used by QueueMetrics to report on the well-being and the actual work being performed by your call-center, and you surely want to keep that data in a safe place for cross-period analysis.

Disabling log rotation

Disabling log rotation is actually quite easy: go to /etc/logrotate.d and look for a file named asterisk. If you run TrixBox, you'll find something like:

```
/var/log/asterisk/*log {
   missingok
   rotate 5
   weekly
   create 0640 asterisk asterisk
   postrotate
       /usr/sbin/asterisk -rx 'logger reload' > /dev/null 2>
/dev/null
   endscript
}
```

There may as well be other sections where other files are rotated. You just delete the section above and the queue_log file will not be rotated anymore.

What if my queue_log has already been rotated?

If your queue_log has already been rotated, you'll want to join the remaining pieces together. That's very easy:

- First, stop Asterisk.
- Make a backup of all *queue_log** files
- Rename the existing *queue_log* to *queue_log.now*.
- Execute the following commands (this example if for 5 leftover pieces, you may find a different number on your system):

```
cat queue_log.5 >> queue_log
cat queue_log.4 >> queue_log
cat queue_log.3 >> queue_log
cat queue_log.2 >> queue_log
```

```
cat queue_log.1 >> queue_log
cat queue_log.now >> queue_log
```

- Remove all files but queue_log itself
- Restart Asterisk.

This should be it. Now your QueueMetrics will work just fine.

Stopping and starting QueueMetrics

You can stop and restart the QueueMetrics application by issuing the commands

```
/etc/init.d/queuemetrics stop
/etc/init.d/queuemetrics start
/etc/init.d/queuemetrics restart
```

Setting QueueMetrics memory limits

QueueMetrics is a complex application and it is made to be used by multiple parallel users. This means that if you have a large data set and many users running queries on it, it is possible that you start getting "Out of memory" errors. To fine-tune the amount of memory used by your system, you can edit the file */etc/init.d/queuemetrics* and modify the option:

```
JAVA_OPTS="-Xms128M -Xmx128M"
```

The Xms parameter is the amount of memory that Java uses on startup for its object heap; and the Xmx is its maximum allowed size. For best speed, keep both to the same value unless you have experience in tuning Java memory requirements.

Using MySQL storage

It is possible to set up QueueMetrics in order to read queue event data right from a MySQL database instead of parsing the queue_log file. This leads to a set of advantages:

- Faster access times for large call centers with many queues or large data sets
- You can run QueueMetrics on a different machine from the one that is running Asterisk
- You can monitor multiple Asterisk instances at once, as if they were a large, clustered single logical Asterisk instance.

In order to do this, a script called *qloaderd* must be installed to upload data from each Asterisk server to the central database. *Qloaderd* can be found with the QueueMetrics distribution under the *WEB-INF/mysql-utils* directory.

We recommend running MySQL storage if your server has over 20 agents on it.

For more information, see the QueueMetrics User Manual and the *qloaderd* help files.

Getting help

If you still are having problems installing or running QueueMetrics on TB/AAH, we suggest you check out the following resources:

- The QueueMetrics FAQ at http://queuemetrics.com/faq.jsp are a collection of common solved problems that many people experienced with QueueMetrics. If you are struck by an error message, this is the first place to look at.
- The QueueMetrics forums at http://forum.queuemetrics.com will help you in pinpointing your problems and getting community support. They will also be helpful in seeing what other people are doing with QueueMetrics.
- AstRecipes is a wiki collecting Asterisk "recipes", aimed mostly at call-center users – see http://astrecipes.net
- You may want to contact Loway if your problems are still unsolved – see http://queuemetrics.com/contact.jsp for all relevant contact information.