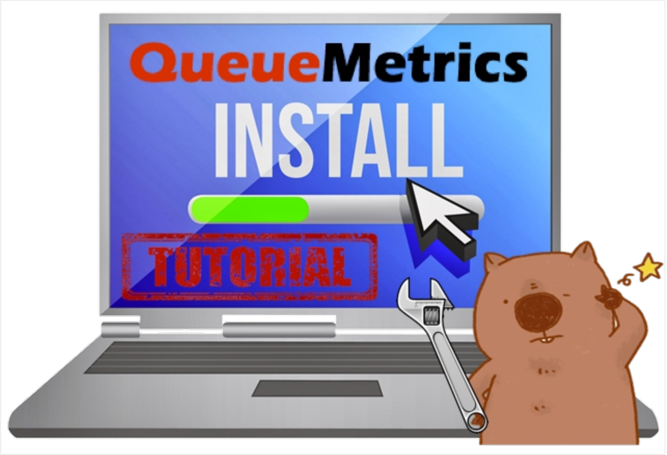
Loway presents

QueueMetrics Installation under Systemd (Debian/Ubuntu)



Installation Tutorial of QueueMetrics' Uniloader on a Debian/Ubuntu system

Under **Systemd** operating systems like **Debian** or **Ubuntu** you hove to install **Uniloader** manually. It might sound a bit daunting however by following this simple steps you will be guided through the process.

What is QueueMetrics?

QueueMetrics is a highly scalable monitoring and reporting suite that addresses the needs of thousands of contact centers worldwide and offers a broad range of integrated benefits like agent productivity monitoring, target measurement, conversion rates tracking, realtime campaign statistics analysis and an easy to use interface. It's available on premise or as a cloud hosted solution service.

What is Uniloader?

Uniloader is a program that is installed on your Asterisk PBX. It uploads data to a local or remote QueueMetrics instance and receives actions to be performed on the local PBX.

Uniloader is deployed as a single binary file that has to be installed on the PBX itself. It is designed as a very lightweight application so it can work unobtrusively even on low-end hardware; and it is meant to be very safe, so data will not be lost even in cases where the remote QueueMetrics server becomes unavailable.

When it runs, it uploads data using either HTTP/S or the MySQL protocol (depending on the back-end you specify). HTTP/S is meant to run with remote QueueMetrics instances, especially QueueMetrics Live (see http://queuemetrics-live.com for more information) , while MySQL is meant for local systems; either case works if QueueMetrics is hosted on the same machine.

When running over HTTP, if the QueueMetrics server has no direct connection to the PBX, Uniloader is able to act as a proxy and will receive actions to be performed on the Asterisk server via AMI (Asterisk Manager Interface). This way you can run QueueMetrics remotely and still take advantage of the ability to log agents on and off, pause them, listen to calls via ChanSpy, etc.

Uniloader is also used to perform other administrative/complementary tasks that perform useful functions on an Asterisk system connected to QueueMetrics; for example, it can generate music-on-hold events on queues, and can help diagnosing issues.

Manual installation under Systemd (CentOS, Debian/Ubuntu)

The Uniloader can be downloaded from: https://www.queuemetrics-live.com/uniloader.jsp

First log in as root, then download the package as TGZ, uncompress it under `/opt` and create a link so the command can be called from anywhere.

```
 cd /opt

wget http://downloads.loway.ch/qm/uniloader-0.4.1.tar.gz

tar zxvf uniloader-0.4.1.tar.gz


cd /usr/bin/

ln -s /opt/uniloader-0.4.1/bin/uniloader_amd64 uniloader
```

Now if we run Uniloader from the shell, it will print its default message.

Tip:

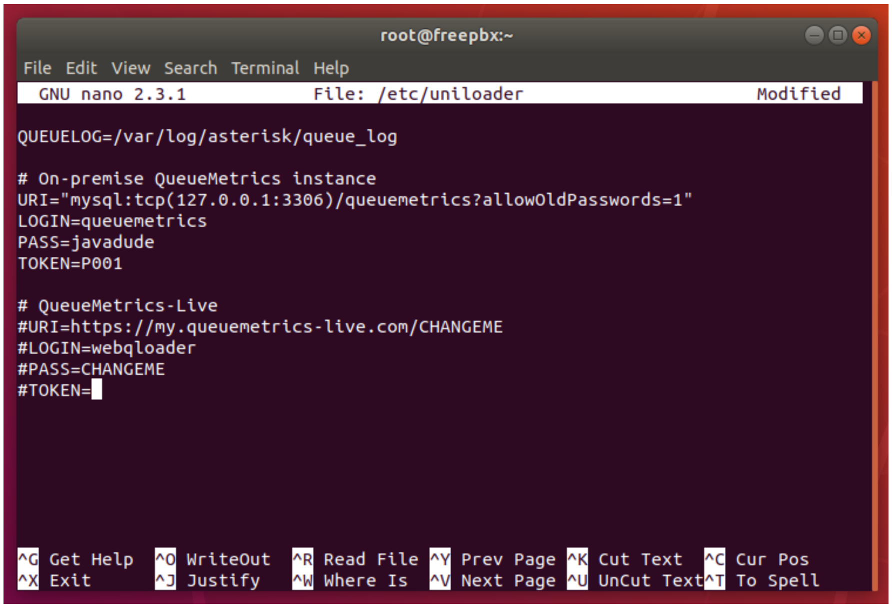
> *All files in the examples below are available within the Uniloader package, under* `init-scripts/systemd` *, so you can just copy them to the right location.* ——————————————————————————————

Uniloader

Create a configuration file as `/etc/uniloader` and set its values (edit is as to suit your system):

```
QUEUELOG=/var/log/asterisk/queue_log


# On-premise QueueMetrics instance

URI="mysql:tcp(127.0.0.1:3306)/queuemetrics?allowOldPasswords=1"

LOGIN=queuemetrics

PASS=javadude

TOKEN=P001


# QueueMetrics-Live

#URI=https://my.queuemetrics-live.com/CHANGEME
```

```
#LOGIN=webqloader

#PASS=CHANGEME

#TOKEN=
```

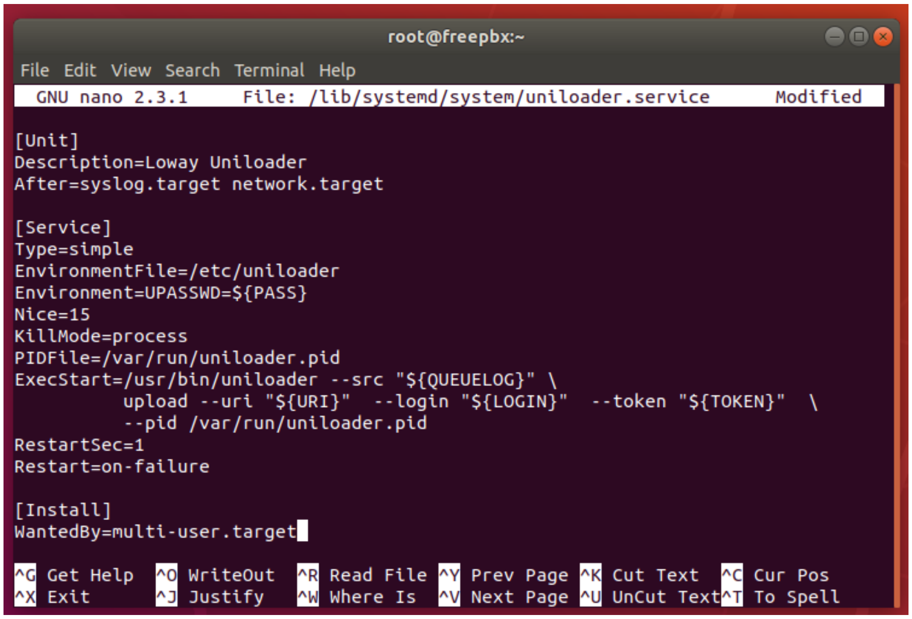


```
  GNU nano 2.3.1              File: /etc/uniloader                      Modified

QUEUELOG=/var/log/asterisk/queue_log

# On-premise QueueMetrics instance
URI="mysql:tcp(127.0.0.1:3306)/queuemetrics?allowOldPasswords=1"
LOGIN=queuemetrics
PASS=javadude
TOKEN=P001

# QueueMetrics-Live
#URI=https://my.queuemetrics-live.com/CHANGEME
#LOGIN=webqloader
#PASS=CHANGEME
#TOKEN=

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

Now create a unit file as `/lib/systemd/system/uniloader.service` :

```
[Unit]

Description=Loway Uniloader

After=syslog.target network.target


[Service]

Type=simple

EnvironmentFile=/etc/uniloader

Environment=UPASSWD=${PASS}

Nice=15

KillMode=process

PIDFile=/var/run/uniloader.pid

ExecStart=/usr/bin/uniloader --src "${QUEUELOG}" \

        upload --uri "${URI}"  --login "${LOGIN}"  --token "${TOKEN}"  \

        --pid /var/run/uniloader.pid
```

```
RestartSec=1
Restart=on-failure


[Install]
WantedBy=multi-user.target
```



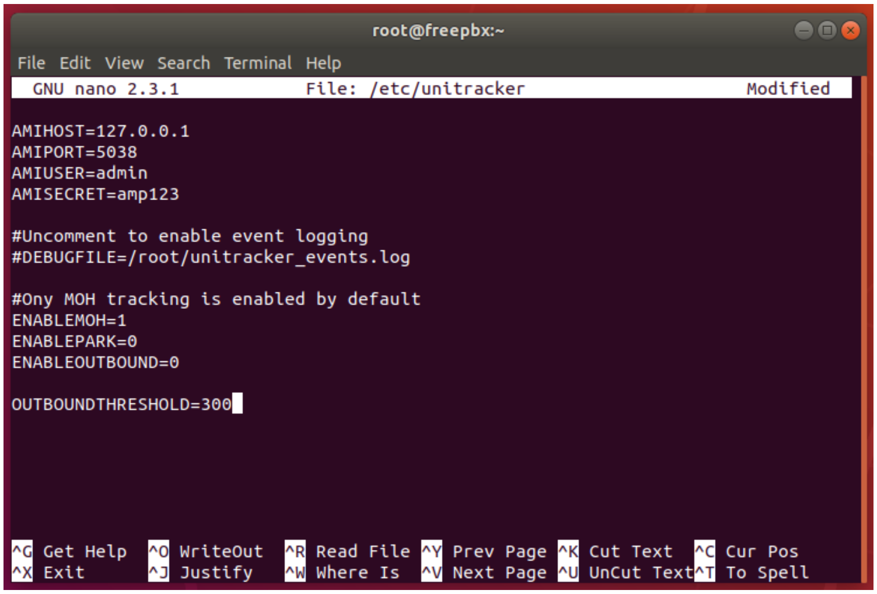
```
  GNU nano 2.3.1      File: /lib/systemd/system/uniloader.service       Modified

[Unit]
Description=Loway Uniloader
After=syslog.target network.target

[Service]
Type=simple
EnvironmentFile=/etc/uniloader
Environment=UPASSWD=${PASS}
Nice=15
KillMode=process
PIDFile=/var/run/uniloader.pid
ExecStart=/usr/bin/uniloader --src "${QUEUELOG}" \
          upload --uri "${URI}"  --login "${LOGIN}"  --token "${TOKEN}"  \
          --pid /var/run/uniloader.pid
RestartSec=1
Restart=on-failure

[Install]
WantedBy=multi-user.target

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

Unitracker

If you also need the tracking service, create a configuration file as `/etc/unitracker` and set its values (edit is as to suit your system):

```
AMIHOST=127.0.0.1
AMIPORT=5038
AMIUSER=admin
AMISECRET=amp123


#Uncomment to enable event logging
#DEBUGFILE=/root/unitracker_events.log


#Ony MOH tracking is enabled by default
ENABLEMOH=1
ENABLEPARK=0
```

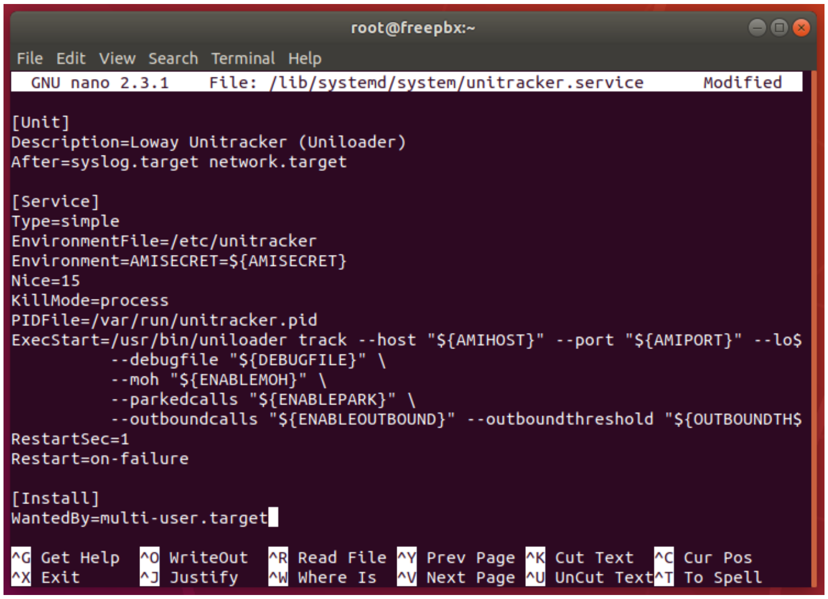
```
ENABLEOUTBOUND=0


OUTBOUNDTHRESHOLD=300
```



```
  GNU nano 2.3.1              File: /etc/unitracker                    Modified

AMIHOST=127.0.0.1
AMIPORT=5038
AMIUSER=admin
AMISECRET=amp123

#Uncomment to enable event logging
#DEBUGFILE=/root/unitracker_events.log

#Ony MOH tracking is enabled by default
ENABLEMOH=1
ENABLEPARK=0
ENABLEOUTBOUND=0

OUTBOUNDTHRESHOLD=300




^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

Now create a unit file as `/lib/systemd/system/unitracker.service` :

```
[Unit]

Description=Loway Unitracker (Uniloader)

After=syslog.target network.target


[Service]

Type=simple

EnvironmentFile=/etc/unitracker

Environment=AMISECRET=${AMISECRET}

Nice=15

KillMode=process

PIDFile=/var/run/unitracker.pid

ExecStart=/usr/bin/uniloader track --host "${AMIHOST}" --port "${AMIPORT}" --login "${AMIUSER}" \

        --debugfile "${DEBUGFILE}" \

        --moh "${ENABLEMOH}" \

        --parkedcalls "${ENABLEPARK}" \
        --outboundcalls "${ENABLEOUTBOUND}" --outboundthreshold "${OUTBOUNDTHRESHOLD}"

RestartSec=1

Restart=on-failure



[Install]

WantedBy=multi-user.target
```



```
[Unit]
Description=Loway Unitracker (Uniloader)
After=syslog.target network.target

[Service]
Type=simple
EnvironmentFile=/etc/unitracker
Environment=AMISECRET=${AMISECRET}
Nice=15
KillMode=process
PIDFile=/var/run/unitracker.pid
ExecStart=/usr/bin/uniloader track --host "${AMIHOST}" --port "${AMIPORT}" --lo$
          --debugfile "${DEBUGFILE}" \
          --moh "${ENABLEMOH}" \
          --parkedcalls "${ENABLEPARK}" \
          --outboundcalls "${ENABLEOUTBOUND}" --outboundthreshold "${OUTBOUNDTH$
RestartSec=1
Restart=on-failure

[Install]
WantedBy=multi-user.target
```

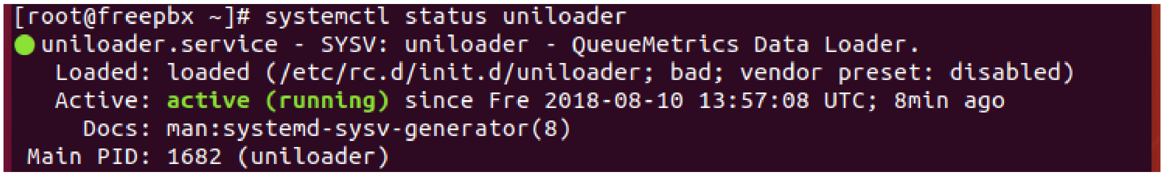
Starting and enabling the services

Under Systemd, you need to notify the daemon that there are new init files, tell it that you want them riun on boot, and start them. Run the following commands for Uniloader:

```
systemctl daemon reload


systemctl enable uniloader


systemctl start uniloader
```

To check that the loader is running:

```
systemctl status uniloader
```



```
[root@freepbx ~]# systemctl status uniloader
● uniloader.service - SYSV: uniloader - QueueMetrics Data Loader.
   Loaded: loaded (/etc/rc.d/init.d/uniloader; bad; vendor preset: disabled)
   Active: active (running) since Fre 2018-08-10 13:57:08 UTC; 8min ago
     Docs: man:systemd-sysv-generator(8)
 Main PID: 1682 (uniloader)
```

And to restart the service after you make some changes to the configuration file:

```
systemctl restart uniloader
```

All logs will be sent to the system journal, and are visible as:

```
journalctl -u uniloader
```

The same procedure must be followed to enable unitracker.

QueueMetrics References

For more technical information about QueueMetrics call center solution please refer to the User Manual.

Visit www.queuemetrics.com for a 30 days full featured trial.

Attend our Free Webinars for a live demonstration of QueueMetrics.