



Abiquo Events Notifier

The Abiquo Events Notifier is a Python application that accesses the Abiquo Database and sends emails to notify users of events in Abiquo. You can configure the users, as well as the types of events that will trigger notification emails. By default, the Email Notifier accesses the Abiquo Database every five seconds but you should tune this value for your installation.

You can run the Abiquo Events Notifier on the Abiquo Server. Download the Abiquo Email Notifier script from the Abiquo Enterprise Support website.

See the User Guide information on Events view about [Abiquo Actions](#) and [Severity Levels](#).

1. Installation and Configuration

1.1. Install the MySQL Python package

```
# yum install -y MySQL-python.x86_64
```

1.2. On the Abiquo database server enter the following commands to grant the necessary access to the Abiquo database.

IP is the IP address of the event notifier server. USER is MySQL user with a secure password. The default USER is "event-notifier". To use a different username enter it in the command below.

```
# mysql kinton
> grant all privileges on kinton.metering TO
'<USER>'@'<IP>' identified by 'PASSWORD';
> grant all privileges on kinton.user TO ' <USER>'@'<IP>'
identified by 'PASSWORD';
> flush privileges;
```

**Example for event-notifier with password event-password at
10.10.10.10**

```
# mysql kinton
> grant all privileges on kinton.metering TO
'event-notifier'@'10.10.10.10' identified by
'event-password';
> grant all privileges on kinton.user TO TO
'event-notifier'@'10.10.10.10' identified by
'event-password';
> flush privileges;
```

1.3. Modify the notifier.cfg file and enter the appropriate values.

Set the polling interval, which is the amount of time between database checks. The default is five seconds and the recommended value is between 5 and 10 seconds, but you should tune the polling value for your installation.

Polling Interval configuration in notifier.cfg file

```
[main]
polling_interval = 5 //in secs
```

MySQL configuration in notifier.cfg file

```
[mysql]
ip = 10.10.10.10
user = event-notifier
pwd = event-password
```

Field	Description
ip	IP address of the Abiquo server
user	Username of the notifier database user (same as USER in step 2 above)
password	Password is the same as the PASSWORD in step 2 above

Email configuration in notifier.cfg file

```
[email]
from = events@abiquoserver
subject = New events have occurred in the Abiquo
environment
body = The new events are:
smtp_ip = localhost
smtp_port = 25
```

Field	Description
from	Sender email address
subject	Subject
smtp_ip	Outgoing mail server IP address. The default is localhost
smtp_port	The default outgoing mail server port is 25

1.4. Initialize the rules database

```
# ./init_rules_db.py
```

1.5. Run the events notifier utility in the background

```
# ./main.py &
```

Alternatively, the events notifier can be run as “./main.py”. Events will then display on the screen.

1.6. Add rules for the events you wish to monitor

Rules can be added while the notifier is running. It will load rules every time it analyzes events, depending on the polling interval.

1.6.1. Examples of Rule Creation

In the following examples, USERA and USERB must be Abiquo users with valid email addresses.

1. Monitor VAPP_CREATE and VAPP_DELETE actions performed by USERA and notify USERB.

```
# ./add_rule.py --user USERB --action
VAPP_CREATE,VAPP_DELETE --owner USERA
```

2. Notify all users when their deploys/undeploys have finished.

```
# ./add_rule.py -u all -a VAPP_POWERON,VAPP_POWEROFF
```

3. Notify USERA when their deploys/undeploys have finished.

```
# ./add_rule.py -u USERA -a VAPP_POWERON,VAPP_POWEROFF
```

4. Notify USERB on all major events.

```
# ./add_rule.py -u USERB -s MAJOR
```

5. Notify the cloud admin user on all critical events.

```
# ./add_rule.py -u admin -s CRITICAL
```

2. List the active rules

```
# ./list_rules.py
```