

Installation

Table of Contents

<u>Chapter 1. Installation</u>	1
<u>Interactive step-by-step installation of PyKota with pksetup</u>	1
<u>Manual installation</u>	2
<u>Database server installation</u>	2
<u>Print Server Installation</u>	8
<u>Notes</u>	12

Chapter 1. Installation

Table of Contents

[Interactive step-by-step installation of PyKota with pksetup](#)

[Manual installation](#)

Last modified on \$Date: 2007-07-09 21:53:04 +0200 (lun 09 de jul de 2007) \$

Before being able to use PyKota, you have of course to install it first. But before installing, you must carefully plan your installation.

First you have to determine which machine will be the PyKota database server. The database server is the host responsible for keeping a centralized database of print usage for all your printers, users and groups.

Then you have to list all the *Print Servers* for which you plan to use print quota facilities.

With most database backends, several print servers can share a single database, however as we'll see later this is not possible if you choose to use SQLite as your print quota database backend.

Finally you have to download PyKota's latest version or buy an official package, from <http://www.pykota.com/software/pykota>. If you've just bought an official package, then as soon as you've receive it you have to decompress and visit its archive, to do so just type the following commands :

```
jerome@nordine:~$ tar -zxf pykota-1.26_official.tar.gz
jerome@nordine:~$ cd pykota-1.26_official
jerome@nordine:~/pykota-1.26_official$
```

You can see many files in this directory, the first ones to read are README, then COPYING and LICENSE. They will give you basic installation instructions and explain the licensing terms under which PyKota is distributed. Of course they are also mostly boring to read ! Detailed installation and operating instructions are defined in the `./docs` directory, in the form of SGML documentation in the [DocBook](#) format. You have to compile these files into readable documentation like the HTML or PDF formats, or buy an official PyKota package which already contains these compiled forms of the documentation. Of course you already know this because that's what you are currently reading !

Interactive step-by-step installation of PyKota with pksetup

pksetup is a command line tool with which you'll be able to install PyKota and all its dependencies in a completely interactive way. At the end of the installation, a shell script is created which allows you to replicate the very same installation in an automated way. This can be useful if you've got several servers to install identically.

Currently, **pksetup** is experimental, and only works with [Debian](#) and [Ubuntu](#) distributions. In addition, the database backend which will be installed with this command is [PostgreSQL](#) and you have no choice for another backend. If you want another database backend, or use a different distribution, or want to do the installation manually, then read and follow the instructions in the next section.

To launch the installation procedure, just type `pksetup` followed with the name of your distribution, like :

Installation

```
jerome@nordine:~/pykota-1.26_official$ ./bin/pksetup debian
```

and then follow the instructions and answer to the several questions you'll be asked.

Manual installation

To do a manual installation, we will see what has to be done on each of the servers we are planning to use.

Note

Of course, depending on the size of your network, you may very well use the same machine as both a Print Server and a database server. This is especially the case if you've got only one server.

Database server installation

Depending on PyKota's version number, different types of storage backends may be supported, so we will see for each one of them how to configure it.

PostgreSQL

PostgreSQL is an *Object Relational DataBase Management System* distributed under a *Free Software* license from the <http://www.postgresql.org> web site. It certainly is the free RDBMS which has the most advanced features, and is widely used all over the world.

To configure your database, you must have PostgreSQL already working. The complete installation of PostgreSQL is not covered by the present manual, please refer to your system's documentation or to <http://www.postgresql.org> for details.

One thing you have to check, though, is that every Print Server on which you want to install the print quota mechanism, must be able to connect to the PostgreSQL server. In the default installation of PostgreSQL this may not be the case for security reasons, except if both servers are in fact the same machine. In any case, it is recommended that you check the `/etc/postgresql/pg_hba.conf` file and modify it if needed. This file is self documented and its modification is straightforward. You also have to make sure that PostgreSQL accepts TCP/IP connections. To do so you either have to launch it with the `-i` option or modify the `/etc/postgresql/postgresql.conf` file, which is self documented and easy to modify too. Allowing TCP/IP connections is not necessary though if your print quota database server and your Print Server are the very same host.

Here's an excerpt from a `pg_hba.conf` file. This one rejects all connections to PyKota's database excepted when made from the same host by PostgreSQL users `pykotauser` or `pykotaadmin` with the correct password.

```
local all postgres ident sameuser
local all all reject
host pykota pykotauser 127.0.0.1 255.255.255.255 crypt
host pykota pykotaadmin 127.0.0.1 255.255.255.255 crypt
host pykota all 127.0.0.1 255.255.255.255 reject
```

Of course if your print server and your database servers have different IP addresses, you have to replace the `127.0.0.1` address above with your print server's IP address. As an alternative, you could still keep these lines and add similar lines with other IP addresses if you have several print servers for which you want a

Installation

single centralized database.

Tip

Don't forget to restart PostgreSQL if you modify any of its configuration files, in order for the changes to take effect.

Be careful, you may be unable to connect from a Print Server to the PostgreSQL server even if the configuration is correct. Sometimes your connections may be blocked by one or more network firewalls along the route from one machine to the other. If this is the case, then the best thing you can do is to ask your *Network Administrator* to not filter the IP port used by PostgreSQL, which is usually port 5432/tcp.

Note

The TCP/IP network port used by PostgreSQL may be different. When in doubt, ask your *System Administrator* for the correct value.

Now that your PostgreSQL server is up and running, and is waiting for your connections, you have to create the print quota database. To do so, you'll have to feed PostgreSQL with the `pykota-1.26_official/initscripts/postgresql/pykota-postgresql.sql` file. This file will create a print quota database administrator in the PostgreSQL system, then create an empty print quota database and set some permissions on it. The print quota database administrator is the PostgreSQL's user used to manage the quota database. The print quota database Administrator is not present in the quota database itself, he is only defined in PostgreSQL and don't have to exist on any system, nor in the print quota database. His default name is `pykotaadmin`. A print quota database read-only user is also created under the name of `pykotauser`. This read-only user is used by PyKota to connect to the print quota database when an user who is not a PyKota administrator [1] launches a `pykota` command. This prevents normal users from being able to modify their own, or other users', quota information. The database which will be created will be named `pykota` by default. The `pykotaadmin` and `pykotauser` users by default respectively have `readwritepw` and `readonlypw` as their passwords.

Note

You can choose other names and passwords if you want by modifying the `initscripts/postgresql/pykota-postgresql.sql` file accordingly, and report your changes into PyKota's configuration files.

To run this script, you can use the **psql** frontend to PostgreSQL, but your privileges must be sufficient to be allowed to create users and databases. You can launch **psql** as the `postgres` user which is PostgreSQL's default administrator, and connect to the default database named `template1`. From a command line interpreter (i.e. shell), type the following commands :

```
jerome@nordine:~$ cd pykota-1.26_official/initscripts/postgresql
jerome@nordine:~/pykota-1.26_official/initscripts$ psql -h localhost -U postgres template1
Welcome to psql, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help on internal slash commands
       \g or terminate with semicolon to execute query
       \q to quit

template1=# \i pykota-postgresql.sql
          ... a lot of output lines
pykota=#
```

Note

If you use RPM or DEB packages, usually the `pykota-postgresql.sql` file gets installed into the `/usr/share/pykota/postgresql` directory, along with a README file.

If you want to you can change passwords later in PostgreSQL for the `pykotaadmin` and `pykotauser` users. To do so, just type the following lines while still being at the **psql** prompt (replace the password values by your own :

```
pykota=# ALTER USER pykotaadmin PASSWORD 'somepassword';
ALTER USER
pykota=# ALTER USER pykotauser PASSWORD 'anotherpassword';
pykota=# \q
jerome@nordine:~/pykota-1.26_official/initialscripts/postgresql$
```

The `\q` command above will quit the **psql** program and return you to the shell's command line prompt.

To improve security further, you could encrypt your database connections, or take any other step as needed. Please refer to PostgreSQL's documentation for details.

Warning

Defining passwords may not be sufficient if your database access rule is set to `trust` in the `/etc/postgresql/pg_hba.conf`. Again, please refer to PostgreSQL's documentation for details. Also, passwords will fly unencrypted over the network by default, so be sure to take any necessary step to secure your database server from unauthorized use. This has nothing to do with PyKota though, it is just a general rule to keep in mind.

For more details, please see `initialscripts/mysql/README.postgresql`.

If no error occurred, then your print quota database is ready to be used. Now you can let the print quota database server alone, the remaining work will have to be done on each one of the print servers which will use this particular print quota database server.

Tip

If an error occurred, maybe your PostgreSQL version is too old, or an unexpected problem (like a bug) happened. Please contact us via email so that we can try to fix the problem. Thanks in advance.

LDAP

Any LDAP server, and particularly OpenLDAP, can be used as a print quota database backend. Some other LDAP servers can be used, but this is currently untested in production.

OpenLDAP is a Lightweight Directory Access Protocol server implementation published as Free Software. You can download it from <http://www.openldap.org>.

To use OpenLDAP as your print quota database backend, you have to copy the `pykota/initialscripts/ldap/pykota.schema` into OpenLDAP's schemas directory. Under Debian GNU/Linux, this is something like :

```
$ cp pykota.schema /etc/ldap/schema
```

Installation

Note

If you use RPM or DEB packages, the `pykota.schema` file is usually installed into the `/usr/share/pykota/ldap` directory, along with a README file, and may also be installed automatically in your LDAP server's schemas directory.

Then edit `/etc/ldap/slapd.conf` and add a line to include the PyKota schema. You should have something like :

```
# Schema and objectClass definitions
include      /etc/ldap/schema/core.schema
include      /etc/ldap/schema/cosine.schema
include      /etc/ldap/schema/nis.schema
include      /etc/ldap/schema/inetorgperson.schema
include      /etc/ldap/schema/pykota.schema
```

While this is not mandatory, it is recommended that you setup some indexes for some often accessed PyKota attributes. Here are the minimal indexes lines you may want to put in `slapd.conf` :

```
# Indexes for PyKota
index pykotaUserName pres,eq,sub
index pykotaGroupName pres,eq,sub
index pykotaPrinterName pres,eq,sub
index pykotaBillingCode pres,eq,sub
index pykotaLastJobIdent eq
```

Now you must ensure that the DN's you'll use to bind to your OpenLDAP server don't have search queries size limits, which gives for example (OpenLDAP 2.1.x or above) :

```
# No Limits for PyKota's administrator and read-only user
limits dn="cn=pykotaadmin,dc=example,dc=com" size.soft=-1 size.hard=soft
limits dn="cn=pykotauser,dc=example,dc=com" size.soft=-1 size.hard=soft
```

Where `pykotaadmin` and `pykotauser` are the usernames used to bind to your OpenLDAP server within PyKota, respectively in ReadWrite mode (as set in `pykotadmin.conf`) and in ReadOnly mode (as set in `pykota.conf`).

Finally, stop the OpenLDAP server, generate the index files, and restart OpenLDAP

```
$ /etc/init.d/slapd stop
$ slapindex
$ /etc/init.d/slapd start
```

With an LDAP backend, PyKota will need some branches in your LDAP directory to put its own datas. You can configure PyKota to either attach its datas to your existing users and groups, or to put them in their own `ou`. But some `ous` dedicated to PyKota are needed in any case, so the best bet may be to put all PyKota's datas below an `ou=PyKota` branch. While this will separate these datas from your existing users and groups entries, this may ease the maintainance.

PyKota needs at least an `ou` for printers, for users quotas, for groups quotas, for print jobs, for billing codes, and for pointers to the last job of each printer. In the future, this last `ou` may disappear as its content will probably be attached to each printer.

Actually PyKota doesn't create these `ous` for you, because it's difficult to guess what is the best configuration

Installation

for you. So you have to create them by yourself, either directly with a text editor and the **ldapadd** command, or with some specialized tool like **gq**. You can look at the `initscripts/ldap/pykota-sample.ldif` file to see which minimal branches are necessary.

Note

If you use RPM or DEB packages, usually the `pykota-sample.ldif` file is installed into the `/usr/share/pykota/ldap` directory, along with a README file.

If no error occurred, then your print quota database is ready to be used. Now you can let the print quota database server alone, the remaining work will have to be done on each one of the print servers which will use this particular print quota database server.

Tip

If an error occurred, maybe your OpenLDAP version is too old, or an unexpected problem (like a bug) happened. Please contact us via email so that we can try to fix the problem. Thanks in advance.

MySQL

MySQL is a simple Relational DataBase Management System distributed under a *Free Software* license from the <http://www.mysql.org> web site.

To configure your database, you must have MySQL version 4.1 or higher already working. We recommend that you use MySQL 5.0 or higher though. The complete installation of MySQL is not covered by the present manual, please refer to your system's documentation or to <http://www.mysql.org> for details.

One thing you have to check, though, is that every Print Server on which you want to install the print quota mechanism, must be able to connect to the MySQL server. In the default installation of MySQL this may not be the case for security reasons, except if both servers are in fact the same machine. In any case, it is recommended that you check the `/etc/mysql/my.cnf` file and modify it if needed.

Tip

Don't forget to restart MySQL if you modify any of its configuration files, in order for the changes to take effect.

Be careful, you may be unable to connect from a Print Server to the MySQL server even if the configuration is correct. Sometimes your connections may be blocked by one or more network firewalls along the route from one machine to the other. If this is the case, then the best thing you can do is to ask your *Network Administrator* to not filter the IP port used by MySQL, which is usually port 3306/tcp.

Note

The TCP/IP network port used by MySQL may be different. When in doubt, ask your *System Administrator* for the correct value.

Now that your MySQL server is up and running, and is waiting for your connections, you have to create the print quota database. To do so, you'll have to feed MySQL with the `pykota-1.26_official/initscripts/mysql/pykota-mysql.sql` file. This file will create an empty print quota database and set some permissions on it. The database which will be created will be named `pykota` by default. Two database users will be defined to have access in readonly and read+write modes under the respective names `pykotauser` and `pykotaadmin`. The `pykotaadmin` and `pykotauser`

Installation

users by default respectively have `readwritepw` and `readonlypw` as their passwords.

Note

You can choose other names and passwords if you want by modifying the `initscripts/mysql/pykota-mysql.sql` file accordingly, and report your changes into PyKota's configuration files.

To run this script, you can use the **mysql** frontend to MySQL, but your privileges must be sufficient to be allowed to create databases. You can launch **mysql** as the `root` user for example. From a command line interpreter (i.e. shell), type the following commands :

```
jerome@nordine:~$ cd pykota-1.26_official/initscripts/mysql
jerome@nordine:~/pykota-1.26_official/initscripts$ mysql pykota-mysql.sql
```

Note

If you use RPM or DEB packages, usually the `pykota-mysql.sql` file gets installed into the `/usr/share/pykota/mysql` directory, along with a **README** file.

To improve security further, you could encrypt your database connections, or take any other step as needed. Please refer to MySQL's documentation for details.

For more details, please see `initscripts/mysql/README.mysql`.

If no error occurred, then your print quota database is ready to be used. Now you can let the print quota database server alone, the remaining work will have to be done on each one of the print servers which will use this particular print quota database server.

Tip

If an error occurred, maybe your MySQL version is too old, or an unexpected problem (like a bug) happened. Please contact us via email so that we can try to fix the problem. Thanks in advance.

SQLite

SQLite is an embeddable Relational DataBase distributed under a Free Software license from the <http://www.sqlite.org> web site. It is very easy to configure and use, offers a very small memory footprint, is very fast, but can only be used on the print server because it doesn't include a server daemon : the database is directly embedded in the application.

To configure your database, you must have SQLite already working. The complete installation of SQLite is not covered by the present manual, please refer to your system's documentation or to <http://www.sqlite.org> for details.

Once SQLite is installed, you have to decide where you'll put your database. A good idea is to store it into the `pykota` user's home directory. Then to create the database, just type :

```
# sqlite3 ~pykota/pykota.db pykota/initscripts/sqlite/pykota.sqlite
# chown pykota.pykota ~pykota/pykota.db
# chmod 660 ~pykota/pykota.db
# chown pykota.pykota ~pykota
```

Installation

If user `pykota` doesn't exist yet, then please follow the instructions a bit below which explain how to install PyKota on the print server.

Once this is done, you'll want to set in `~pykota/pykota.conf` the following lines in the `[global]` section :

```
storagebackend : sqlitestorage
storagename : /etc/pykota/pykota.db
```

Of course you'll want to replace the path on the `storagename` line with the full path to the newly created SQLite database.

If no error occurred, then your print quota database is ready to be used. In case you need them, additional instructions are available in `pykota/itscripts/sqlite/README.sqlite`

Tip

If an error occurred, maybe your SQLite version is too old, or an unexpected problem (like a bug) happened. Please contact us via email so that we can try to fix the problem. Thanks in advance.

Berkeley DB

A Berkeley DB backend is planned, but it actually doesn't exist. It seems that remote storage won't be possible with such a backend, so in other terms this means that you will have a different quota database on each print server. This may still prove to be useful for small configurations.

Print Server Installation

For each Print Server on which you plan to implement the print quota mechanism, you have, of course, to have an already working printing environment. Currently PyKota works with CUPS but older releases also supported LPRng. LPRng support might be re-added in the future.

Here's the list of software you have to install on each Print Server, version numbers are given as an indication of which was successfully tested, but older versions may work too.

- CUPS version 1.1.14 or higher, version 1.2.4 or higher is recommended. You can download it from <http://www.cups.org>
- Python version 2.3 or higher. You can download it from <http://www.python.org>. While PyKota itself will try to preserve compatibility with Python version 2.3 for the near future, some Python modules which are needed by PyKota may require a more recent version of this language.
- print quota database client libraries, depending on your print quota database backend :

- ◆ PostgreSQL backend :

- ◇ PostgreSQL client libraries. They must match the PostgreSQL version used on your print quota database server.

- ◇ The Pygresql python module. Pygresql is normally included in PostgreSQL, but you may want to download it from <http://www.pygresql.org>

- ◆ OpenLDAP backend :

Installation

- ◊ OpenLDAP client libraries. They must match the OpenLDAP version used on your print quota database server.
- ◊ The Python-LDAP python module. You may download this module from <http://python-ldap.sourceforge.net>
- ◆ MySQL backend :
 - ◊ MySQL client libraries. They must match the MySQL version used on your database server.
 - ◊ The Python-MySQL python module, version 1.2.x or higher. You can download it from <http://sourceforge.net/projects/mysql-python>
- ◆ SQLite backend : SQLite is not a database server, but an embeddable database, so if you want to use it you MUST install SQLite on your print server. With PostgreSQL, MySQL or OpenLDAP you can store your datas on a different machine than the print server, but this is not possible with SQLite.
 - ◊ SQLite version 3.2.1 or higher and its library. You can download it from <http://www.sqlite.org>
 - ◊ The Python-SQLite python module version 2.0.5 or higher. You can download it from <http://www.pysqlite.org>
- ◆ Berkeley DB backend : Not supported yet.
- ucd-snmp or net-snmp tools, version 4.2.5 or above. You only need the **snmpget** command. You can download this software from <http://www.sourceforge.net/projects/net-snmp/>. You only need this if PyKota's internal SNMP accounting code doesn't work for your SNMP-aware printers.
- netatalk version 1.6.1 or above. You only need the **pap** command. You can download this software from <http://netatalk.sourceforge.net/>. You only need this if you plan to query your printers for their internal page counter via AppleTalk.
- eGenix' mxDateTime Python module version 2.0.3 or above. It must match your default Python version. You can download it from <http://www.egenix.com>.
- The Python accelerator Psyco. It must match your default Python version. You can download it from <http://psyco.sourceforge.net>. You only need this if you run on the x86 architecture because Psyco doesn't yet exist on other architectures.
- The pysnmp Python module version 3.4.2, or higher, version 4 is recommended. You can download it from <http://pysnmp.sourceforge.net>.
- The JAXML Python module. You can download it from <http://www.librelogiciel.com/software/>.
- The ReportLab Toolkit Python module. You can download it from <http://www.reportlab.org>.
- The Python Imaging Library - PIL module. You can download it from <http://www.pythonware.com>.
- The PyOSD Python module. You can download it from <http://repose.cx/pyosd/>.
- The pkpgcounter Generic Page Description Language parser. You can download it from <http://www.pykota.com/software/pkpgcounter>.
- The PyPAM Python interface to PAM. You'll need this if you plan to ask users to authenticate when printing through **pknotify** and **pykoticon**. You don't need this module otherwise. If needed, you can download it from <http://www.pangalactic.org/PyPAM/>.
- The PkIPPLib Python IPP library. You can download it from <http://www.pykota.com/software/pkipplib>.

Instead of downloading all these programs' sources and compiling them, which really is a boring task considering that many software are needed, you may prefer to look into the packages included with your GNU/Linux distribution of choice (if you use this operating system of course). Most, if not all, GNU/Linux distributions include all the software mentioned above, in the form of packages which are easier to install than sources tarballs. This is probably the same for the many *BSD distributions.

Installation

You can check that all needed software is installed by launching the **checkdeps.py** command :

```
$ python checkdeps.py
```

Once all these software are installed, installing PyKota itself is a breeze. PyKota being written entirely in the Python language, which is interpreted, there's no need to compile anything. You just have to execute the installation script :

```
$ python setup.py install
```

The setup script will automatically create the `/usr/share/pykota/conf` directory and put the sample configuration files `conf/pykota.conf.sample` and `conf/pykotadmin.conf.sample` there, along with a README file explaining their purpose.

Now you have to create a `pykota` system user and group. The PyKota software will automatically search its configuration files in user `pykota`'s home directory. For example we could create the user and group, and set `/etc/pykota` as the home directory, but any other home directory will do :

```
adduser --system --group --home /etc/pykota --gecos PyKota pykota
```

You now have to copy the sample configuration files into the `~pykota` directory, under the respective names `pykota.conf` and `pykotadmin.conf`. Once copied there, you just have to modify these files to adapt them to your own setup. These files are heavily commented, so you should have no problem. Also their format is quite common, because it's the one used by Samba for example, or by `.ini` files under MS-Windows, so you may already be familiar with this syntax. In a future release, this documentation will include the complete reference for all configuration fields available. Keep in mind that PyKota can be really heavily customized, and can delegate some work to any external command of your choice.

Please create a backup copy of the `~pykota` directory before modifying a working installation.

PyKota features some interesting possibilities which allow you to define options either globally so that they apply to all printers, or on a per printer basis. Please see the sample configuration files to see what I mean. In the simplest form, only a `[global]` section is needed. In more complex configurations, you will have to create one section per printer. Each section in the configuration files begins with a name between square brackets `[]`. The name to use to define a particular printer section is the name of the print queue you want to manage with PyKota.

After you have modified PyKota's configuration files, you have to double check their permissions, otherwise your installation may be insecure or may not work at all. The main configuration file `~pykota/pykota.conf` doesn't contain much sensitive information, so it can be made readable by anyone. If normal users read this file, at best they will learn the username and optional password of the read-only database user. This means that beside being allowed to read all the contents of the quota database, they won't be allowed to modify or delete it. On the other hand, the `~pykota/pykotadmin.conf` file contains the read-write user's identity and password. You must then ensure that no normal user can read this file. It should only be readable by the `root` user, which is always the case, and by PyKota administrators. In addition, users for which CUPS doesn't run as user `root` will have to ensure that the user their printing system is run as can read both of these files. An easy way to do so is to put the `lp` user (for example) into the `pykota` system group, then to give the correct permissions to PyKota's configuration files :

```
$ chown -R pykota.pykota ~pykota/  
$ chmod 750 ~pykota/  
$ chmod 644 ~pykota/pykota.conf
```

Installation

```
$ chmod 640 ~pykota/pykotadmin.conf
```

Warning

All the users allowed to read the `~pykota/pykotadmin.conf` are considered to be PyKota administrators. So be careful with these files permissions.

On some systems, you may be able to strenghten permissions like this :

```
$ chown -R pykota.pykota ~pykota/  
$ chmod 750 ~pykota/  
$ chmod 640 ~pykota/pykota.conf  
$ chmod 600 ~pykota/pykotadmin.conf
```

And on other ones, you may need to relax them, and change the files' owner :

```
$ chown pykota.pykota ~pykota/  
$ chmod 755 ~pykota/  
$ chown lp.pykota ~pykota/pykota.conf  
$ chmod 640 ~pykota/pykota.conf  
$ chown lp.pykota ~pykota/pykotadmin.conf  
$ chmod 640 ~pykota/pykotadmin.conf
```

This all depends on the printing system you are using, and the user the printing system is usually running as. You need to remember three things :

- The user your printing system runs as **MUST** be allowed to read both PyKota's configuration files.
- Any user who can read `pykotadmin.conf` is a PyKota administrator, and can do whatever he wants to the print quota database.
- If `cupsd.conf` contains `RunAsUser`, then you won't be able to authenticate users with **pknotify** and **pykotion**. Also in this case you may have to make PyKota's configuration files owned by the user CUPS runs as.

Don't forget to restart your print server software if you changed group membership for the user it runs as, otherwise your change wouldn't be taken into account.

Now depending on your printing system, the configuration to do is particular. We will now see how to plug PyKota into CUPS since LPRng is not supported anymore.

With CUPS

From version 1.16alpha7 on, configuring PyKota to integrate within CUPS is more than easy.

You just have to create a symbolic link to the **cupspykota** command in CUPS' backend directory :

```
$ cd /usr/lib/cups/backend  
$ ln -s /usr/share/pykota/cupspykota cupspykota
```

If you use CUPS v1.2 or higher, you must also type the following command to allow the **cupspykota** backend to correctly support other backends which must be run as the root user (e.g. the **lpd** backend) :

```
$ chmod 700 /usr/share/pykota/cupspykota
```

Installation

You have to restart CUPS for this modification to take effect :

```
$ /etc/init.d/cupsys restart
```

Now point your web browser to CUPS configuration page, usually at <http://localhost:631> on your print server.

Then when creating new printers or reconfiguring existing ones, just choose devices which are PyKota managed [\[2\]](#) instead of normal devices. You've got one PyKota managed device for each regular device available from CUPS, so just choose the appropriate one.

Repeat the above procedure for each print queue on which you want to use PyKota. That's all !

Troubleshooting

In case of problem, the simplest way to solve it is currently to ask on PyKota's mailing list, describing the symptoms, as well as the hardware and software you use.

A searchable FAQ is now available at <http://otrs.librelogiciel.com/public.pl>. A FAQ entry explaining in great details how to diagnose a problem correctly is available at <http://otrs.librelogiciel.com/public.pl?ID=2>.

You can also ask questions on IRC :

```
/server irc.freenode.net  
/join #pykota
```

Notes

- [\[1\]](#) a PyKota administrator is an user who can read the `~pykota/pykotadmin.conf` file.
- [\[2\]](#) Debian 3.0 Woody is known to have problems : CUPS 1.1.14 doesn't automatically detect PyKota managed devices. So you have to manually modify CUPS' `printers.conf` file as explained in PyKota's toplevel README file.