

MySQL-related Hints

Optimizing queries

See [db/MySQL/Optimizations](#)

After you receive your new temporary password

...please change it as soon as possible as follows:

Set the `MYSQL_HISTFILE` environment variable to `/dev/null` (By doing it you will prevent the command with your new password from being saved in clear-text history file.).

Then:

```
mysql -h lsst10.ncsa.illinois.edu -u<userName> -p
Password: <type current mysql password>
```

then

```
mysql> set password = password('theNewPassword');
```

You should then create a `~/.lsst` directory and a `~/.lsst/db-auth.paf` file containing the following (the database server has several IP address aliases; it is safest to list all of them):

```
database: {
  authInfo: {
    host: lsst-db.ncsa.illinois.edu
    port: 3306
    user: <your mysql user name>
    password: <your new mysql password>
  }
  # older addresses to support older code
  authInfo: {
    host: lsst10.ncsa.illinois.edu
    port: 3306
    user: <your mysql user name>
    password: <your new mysql password>
  }
  authInfo: {
    host: lsst10.ncsa.uiuc.edu
    port: 3306
    user: <your mysql user name>
    password: <your new mysql password>
  }
}
```

The directory should have 700 permissions and the file should have 600 permissions (go-rwx in both cases).

Compiling against MySQL

If you are using the LSST stack and eups:

- source /lsst/DC3/stacks/default/loadLSST.csh
- lsstpkg install mysqlclient
- setup mysqlclient
- then use \$MYSQLCLIENT_DIR/{include,lib}

If not, you will need to install MySQL (download from mysql.com and install or use rpm/yum/apt-get/....)

Accessing Database from Python

In many cases we need to access database from python. Examples include scripts for initializing database, or purging of expired databases. In DC2 we accessed mysql through command line calls, for example:

```
cmd = 'mysql -h lsst10.ncsa.uiuc.edu -u admin -pAdminPwd -e "CREATE DATABASE x"'
subprocess.call(cmd.split())
```

This is neither efficient, nor secure.

Proposal: use the Python Database API (DBAPI) compliant module (see *Python in a Nutshell*, pages 292-301). This would require importing a DBAPI-compliant module for MySQL (available from <http://sourceforge.net/projects/mysql-python>). The DBAPI is RDBMS agnostic, if we ever chose to go with a different DBMS vendor, we would simply need to import a different module.

Main advantages of this approach: portability, security and convenience.