pgloader

Author :	Jean-Paul Argudo <jean-paul.argudo@dalibo.com>
Version :	doc_pgloader.rest,v 1.1 2005/11/21 16 :05 :50 jpargudo Exp
Type :	User manual
Comment :	pgLoader v.1.x documentation (install, usage and example)
Licence :	BSD

1 About

pgloader (http ://pgfoundry.org/projects/pgloader/) is a new project allowing you to import data in a PostgreSQL database.

You have to launch pgloader as many times you have tables. pgloader handles just one table at a time.

All bad records are put together in a file, with a logfile explaining origins of errors.

2 Installation

Under Debian, the current installation is a bit tricky (as per 200510xx) :

```
wget http ://debian.wow-
vision.com.sg/debian/pool/main/p/postgresql libpgtcl_7.4.7-6sarge1_i386.deb
dpgk -i libpgtcl_7.4.7-6sarge1_i386.deb
apt-get install tcllib
wget http ://pgfoundry.org/frs/download.php/233/pgloader-1.0.tar.gz
tar zxvf pgloader-1.0.tar.gz
```

Then you can eventually put the binary into /usr/local/bin to facilitate comandlines :

```
$ cp pgloader-1.0/pgloader /usr/local/bin
```

3 Principle

You must fill two files per table :

- a parameter file, let's call it <table>.conf
- a datafile, let's call it <table>.data

You need also all necessary parameters to the db connexion you want to use :

Common ones are the following :

- host : name of the server where your PostgreSQL db lives (localhost ?)
- user : username (you ?)
- password : username's password (mybigsecret)
- dbname : name of the PostgreSQL db

This parameters are put together in a double-quoted string :

"host=localhost user=me password=mybigsecret dbname=mydatabase"

This string as the same type that PQconnectdb awaits for in the libpq. Its complete documentation can be read at : http ://www.postgresql.org/docs/current/static/libpq.html#LIBPQ-CONNECT

You can for sure add much more parameters, depending your db configuration.

4 Example

We want to insert records in "foo" table :

```
test=> \d foo
        Table «public.foo»
 Colonne |  Type   | Modificateurs
---------+---------+---------------
 a       | integer | not null
 b       | date    |
 c       | text    |
Index :
    «foo_pkey» PRIMARY KEY, btree (a)
```

4.1 The datafile

Our datafile "foo.data" as following records :

```
1;1987-12-04;"This is a test of data file"
2;2005-03-02;"diziz'another test with som'o'lil'quotes"
42;;"No need to date this"
67;1999-01-02;Oops I didn't escape this string?!
```

Please note that :

- fields are separated with a semicolon
- you can handle presence of empty data : the empty field is represented with two semicolons following
- we have a record per line
- theres is no other line separator excepted n
- dates are in ISO format : YYYY-MM-DD (a fix is coming to handle “set datestyle to” in the conf file)
- you can escape strings, optionnaly, double quoting them

4.2 Configuration file

The corresponding file “foo.conf” for the above datafile is the following :

```
# ----
# Conversion parameter file for pgloader
#
#      Possible file formats :
#              COPY            native PostgreSQL COPY format (default)
#              CSV                     Comma separated variables
#              MSCSV           Comma separated variables alternate format
#
#      The COPY command is constructed from the table_name, the
#      table_columns and the eventual nulls string definition.
#
#      The default column separator character is comma.
# ----

table_name     = foo
table_columns  = a,b,c
file_format    = CSV
group_size     = 1000
file_sepchar   = ;
#nulls         = NULL
quote          = "
file_is_utf8   = 0
```

Note that separation character is set to “ ;” and that quoting is specifyied with the character double-quote : “

Inserts will be commited each 1000, per blocks of 1000 rows at a time.

The datafile nor the database is in utf-8, so the parameter *file_is_utf8* is set to 0. Set it to 1 otherwise : when both database and datafile are in utf-8.

Since `pgctl` internals run in utf-8, the data must be converted *on the fly* to utf-8 when reading the datafile, thats why pgloader needs to know how is the datafile like, utf-8 or not.

4.3 pgloader execution

The execution is quite simple :

```
$ pgloader foo.conf foo.data "host=localhost user=me password=mybigsecret \
  dbname=mydatabase"

4 row(s) loaded
0 row(s) rejected
```

A simple verification of what has been inserted :

```
test=> select * from foo ;
 a  |     b      |                    c
----+------------+-----------------------------------------
  1 | 1987-12-04 | This is a test of data file
  2 | 2005-03-02 | diziz'another test with som'o'lil'quotes
 42 |            | No need to date this
 67 | 1999-01-02 | Oops I didn't escape this string?!
(4 lines)
```

Note : You will find this example in the doc/example/ directory.

4.4 when errors occurs

Check the following :

- if your configuration file is not okay, pgloader will tell you whats wrong
- if you have a problem with the data you try to import, you'll find in the .rej file data that have bee rejected. In the .rejlog file given problems will be explicited : a group of error messages per rejected row.

Then you'll have to correct errors in .rej file and import *that* file like all the others : don't reimport anything else, all the good data is already in the box :)