Installer PostgreSQL 8.0.0 et PostGIS 1.0.0 RC6 sous Windows

TECHER David

Installer PostgreSQL 8.0.0 et PostGIS 1.0.0 RC6 sous Windows

TECHER David

Ce document est un recueil de notes pour une compilation et une installation personnelle de PostGIS/PostgreSQL sous Windows. Les divers installations possibles concernent:

- PostgreSQL 8.0.0
- PostGIS 1.0.0 RC6
- Geos 2.1.1
- Proj 4.4.9

Table des matières

1. Pré-requis 1
    1. MinGW 1
    2. Msys 1
    3. Suppléments: Bizon, Flex et Zlib et MsysTDK 1
    4. Installation de Perl 2
2. Compilations et Installations 3
    1. Création des répertoires des sources et du répertoire de destination (facultatif) 3
    2. PostgreSQL 4
        2.1. Téléchargement 4
        2.2. Compilation et Installation 4
    3. Geos et Proj 5
        3.1. Téléchargement 5
        3.2. Compilations et Installations 5
        3.3. Création des DLL 5
    4. PostGIS 6
        4.1. Téléchargement 6
        4.2. Compilation et Installation 6
3. Paramétrer PostgreSQL 7
    1. Les variables d'environnement de PostgreSQL 7
    2. Le super-utilisateur de PostgreSQL 8
        2.1. Le super-utilisateur? 8
        2.2. Création du super-utilisateur 9
        2.3. Pouvoir accéder à la session du super-utilisateur sous DOS sans changer la session en cours 9
    3. Initialisation de PostgreSQL 10
    4. Démarrer/Arrêter le serveur et autoriser les connexions TCP/IP 10
        4.1. Démarrage/Arrêt manuel 10
        4.2. Démarrage automatique 10
            4.2.1. Installation du service 10
            4.2.2. Lancer le service 10
    5. Devenir soi-même super-utilisateur de PostgreSQL 11
4. PostGIS 12
    1. Créer une base avec PostGIS 12

Chapitre 1. Pré-requis

Avant de procéder à une compilation des outils, il est nécessaire de disposer d'un environnement permettant leurs compilation sous Windows: MinGW/Msys et Perl.

`MinGW` ainsi que `Msys` sont nécessaires pour pouvoir compiler PostgreSQL et les autres. MinGw "propose" l'utilisation de `make` ainsi que du compilateur `gcc` sous Windows. Msys servira surtout pour pouvoir utiliser la commande `configure.`

1. MinGW

Télécharger le fichier suivant MinGW-3.1.0-1.exe en vous rendant à l'URL suivante:

- `MinGW`http://prdownloads.sf.net/mingw/MinGW-3.1.0-1.exe?download

Il s'agit d'une installation standarde. Précisez juste `c:\MinGW` comme répertoire d'installation. Ajoutez au début de la variable d'environnement `PATH` de votre machine `C:\MinGW\bin;C:\MinGW\lib`

2. Msys

Télécharger le fichier suivant MSYS-1.0.9.exe en vous rendant à l'URL suivante:

- `Msys` http://prdownloads.sf.net/mingw/MSYS-1.0.9.exe?download

Lors de l'installation de Msys, choisissez comme proposé par l'installeur: `c:\msys\1.0` comme chemin d'installation. Une fenêtre DOS s'ouvrira. Trois questions vous seront poser. Répondez par

- question 1: y
- question 2: y
- question 3: c:/MinGW

3. Suppléments: Bizon, Flex et Zlib et MsysTDK

Ces utilitaires/librairies sont aussi écessaires pour les compilations de PostgreSQL/PostGIS. Ils complèteront notre environnement de travail de MinGW/Msys. Ces quatres outils se trouvent repectivement aux URL suivantes

- `bison` http://www.mingw.org/download.shtml (Fichier .exe)
- `msysDTK` http://www.mingw.org/download.shtml (Fichier .exe)
- `zlib` http://gnuwin32.sourceforge.net/packages.html (Lien SetUp)
- `Flex` http://gnuwin32.sourceforge.net/packages.html (Lien:SetUp)

Lors de leur installation, précisez comme répertoire d'installation `c:\MinGW`

4. Installation de Perl

Perl sera nécessaire pour la compilation de PostGIS notamment pour la mise à jour du fichier `lwpost-gis.sql` (par exemple). Il suffit d'installer au moins la version Active Perl 5.6. Téléchargez le fichier

`Perl` http://www.activestate.com/Products/Download/Download.plex?id=ActivePerl

Prendre l'installeur en MSI. C'est une installation standard, indiquez comme chemin d'instalaltion **c:\Perl**. Lors de l'installation, vérifiez de valider l'option de modification de la variable d'environnement PATH: il faut que le répertoire **c:\Perl\bin** soit à la fin de l'installation répertorié dans votre variable PATH. Sinon ajoutez-le à la main.

Chapitre 2. Compilations et Installations

Nous allons ici procéder aux diverses compilations et installations. Les diverses étapes doivent avoir lieu dans l'ordre chronologique des sections successives de ce chapitre. Nous commencerons par définir une hiérarchie des répertoires des sources pour pouvoir mieux nous retrouver pour la suite et accueillier les sources que nous aurons téléchargés.

L'installation de nos binaires auront lieu vers le répertoire `C:\PostgreSQL\8.0.0`.

Nous allons lancer un terminal MinGW. Pour celà, double-cliquez sur **C:\msys\1.0\mys.bat**
*Pour la suite, dans les lignes de commandes à saisir dans MinGW ou dans les répertoires de télécharchement des sources (voir les sous-sections "Téléchargement"), **XXX** désigne le répertoire **C:\msys\1.0\home\XXX** que Msys vous aura créer. Regardez pour celà le nom du répertoire contenu dans **c:\msys\1.0\home** . Notons aussi que, dans les lignes de commandes de ce chapitre, la caractère ~ correspondra à ce répertoire. De plus, dans les lignes de commandes, le caractère \ indique que la ligne de commande en cours continue sur la ligne suivante (pour des raisons d'affichage...) Pour la compilation de PostgreSQL et de Geos, veuillez à préparer le café ou de la lecture de magazines :-)*

1. Création des répertoires des sources et du répertoire de destination (facultatif)

Une fois la fenêtre de MinGW lancée, tapez les commandes suivantes

```
cd ~
mkdir sources
mkdir sources/PostgreSQL
mkdir sources/PostGIS
mkdir sources/Geos
mkdir sources/Proj
mkdir /c/PostgreSQL
```

Avec ces commandes, nous mettrons donc nos sources dans **c:\msys\1.0\home\XXX\sources** et notre répertoire de destination sera **c:\PostgreSQL**. Ces deux répertoires correspondent au répertoires racines des sources et de destination.

Figure 2.1. Commandes de création des répertoires

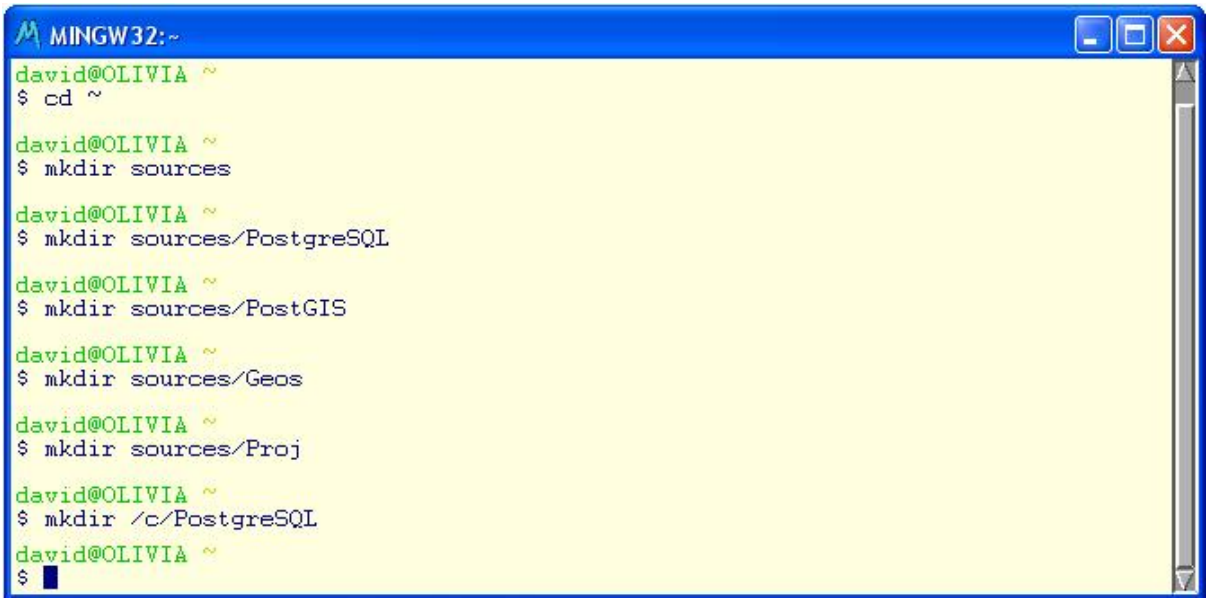
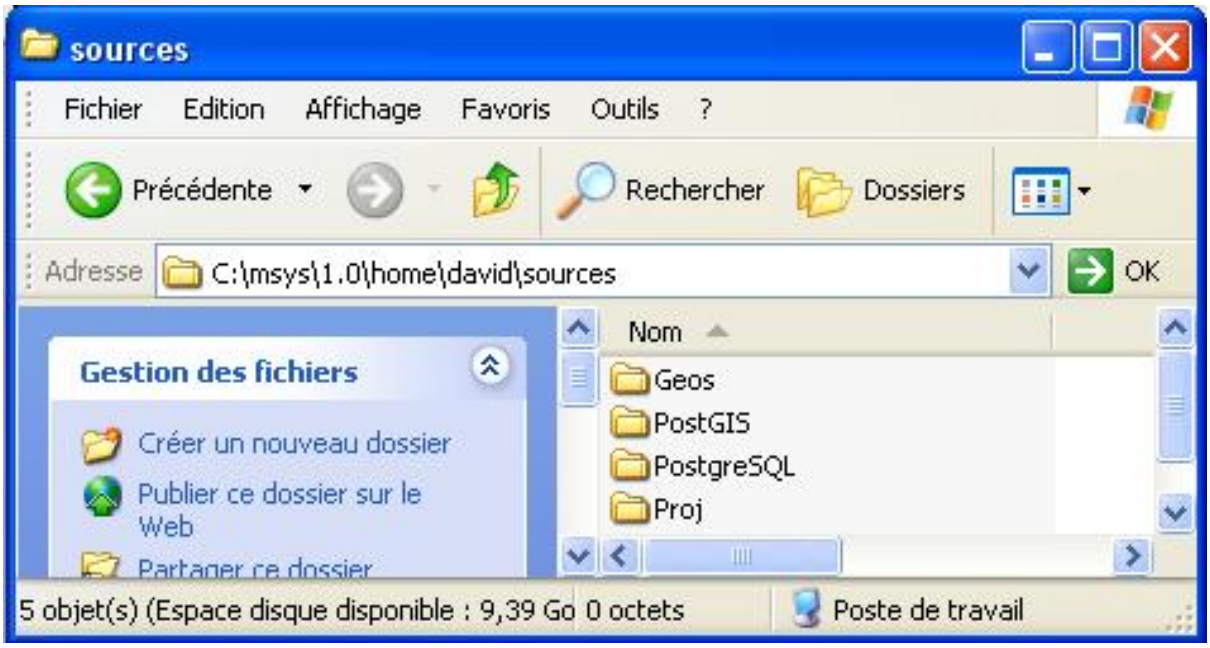


Figure 2.2. Les répertoires des sources (créés par MinGW)



2. PostgreSQL

2.1. Téléchargement

Téléchargez les sources de PostgreSQL version 8.0.0 à cette URL

ftp://ftp2.fr.postgresql.org/pub/unix/postgresql/source/v8.0.0beta/postgresql-8.0.0.tar.gz

et copiez ce fichier vers **`c:\msys\1.0\home\XXX\sources\PostgreSQL`**

2.2. Compilation et Installation

Depuis MinGW, tapez les commandes suivantes. Arrivé à la commande make, allez-vous préparer un café!!! Y'en a un pour au moins 20 à 40 minutes de compilation selon votre machine.

```
cd ~/sources/PostgreSQL

tar xvzf postgresql-8.0.0.tar.gz

cd postgresql-8.0.0

configure --prefix=/c/PostgreSQL/8.0.0

make

make install
```

*La commande **cd** - tout le monde sait - sert à se rendre dans un répertoire. La commande **tar** sert à décompresser des archives compressés se terminant par .tar, .tar.bz2 ou d'autres formats. La commande **configure** permet de vérifier les dépendances envers/avec les autres outils/librairies présentes dans l'environnement. et de les valider. L 'option **--prefix** permet de savoir où sera installée la distribution du matériel. Ici ce sera / c/PostgreSQL/8.0.0 qui en chemin DOS revient à C:\PostgreSQL\8.0.0. La commande **make** sert à compiler les sources présentes dans les divers sous-répertoires du matériel à installer. La commande **make install,** une fois la compilation réussie va copier les divers binaires résultants (fichiers .dll et .exe etc...) dans les divers répertoires prévues à cet effet.. Dans le cas de PostgreSQL, vous devriez donc vous retrouverez avec les répertoires bin, lib, include, doc, share dans C:\PostgreSQL\8.0.0 correspdants à ces fameux répertoires prévus à cet effet. L e répertoire 8.0.0 est automatiquement créé par **make install**.*

3. Geos et Proj

Nous allons maintenant nous intéresser à l'installation de Geos et de Proj

3.1. Téléchargement

Téléchargez les sources respectives de Geos et de Proj grâce à leurs URL respectives

http://geos.refractions.net/geos-2.1.1.tar.bz2

ftp://ftp.remotesensing.org/pub/proj/proj-4.4.9.tar.gz

et copiez-les respectivement vers **`c:\msys\1.0\home\XXX\sources\Geos`** et **`c:\msys\1.0\home\XXX\sources\Proj`**

3.2. Compilations et Installations

Depuis MinGW, tapez les commandes suivantes. Pour la compilation de Geos, un deuxième café ne serait pas de trop!!!

```
cd ~/sources/Geos

tar xvjf geos-2.1.1.tar.bz2

cd geos-2.1.1

configure --prefix=/c/PostgreSQL/8.0.0 && make && make install

cd ~/sources/Proj

tar xvzf proj-4.4.9.tar.gz

cd proj-4.4.9

configure --prefix=/c/PostgreSQL/8.0.0 && make && make install
```

3.3. Création des DLL

Il faut pour notre distribution créer deux librairies:`libproj.dll` et `libgeos.dll` dans le répertoire **c:\PostgreSQL\8.0.0\lib**

```
cd /c/PostgreSQL/8.0.0/lib

g++ -shared -o libgeos.dll -Wl,--out-implib=libgeos.dll.a -Wl,--export-all-symbols -Wl,\
--enable-auto-import -Wl,--whole-archive libgeos.a -Wl,--no-whole-archive /c/mingw/lib/libmingw32.a

cd c/PostgreSQL/8.0.0/lib

gcc -shared -o libproj.dll -Wl,--out-implib=libproj.dll.a -Wl,--export-all-symbols -Wl,\
--enable-auto-import -Wl,--whole-archive libproj.a -Wl,--no-whole-archive /c/mingw/lib/libmingw32.a
```

4. PostGIS

Nous allons maintenant nous intéresser à l'installation de PostGIS

4.1. Téléchargement

Téléchargez les sources de PostGIS version 1.0.0 RC6 à cette URL

http://postgis.refractions.net/postgis-1.0.0-rc6.tar.gz

et copiez ce fichier vers **`c:\msys\1.0\home\XXX\sources\PostGIS`**

4.2. Compilation et Installation

Depuis MinGW, commençons par décompresser les sources:

```
cd ~/sources/PostGIS

tar xvzf postgis-1.0.0-rc6.tar.gz
```

*Ouvrez ensuite le fichier **c:\msys\1.0\home\XXX\sources\PostGIS\postgis-1.0.0-rc6\loader\Makefile** et à la ligne 9 (environ) du fichier remplacer 'EXEC = ' par 'EXEC = .exe ' de manière à pouvoir compiler/installer les fichiers shp2pgsql.exe et pgsql2shp.exe.*

Depuis MinGW, faites

```
cd ~/sources/PostGIS/postgis-1.0.0-rc6

configure --enable-autoconf --with-geos=/c/PostgreSQL/8.0.0 --with-proj=/c/PostgreSQL/8.0.0 \
--with-pgsql-src=/home/XXX/sources/PostgreSQL/postgresql-8.0.0 --prefix=/c/PostgreSQL/8.0.0
```

Dans la ligne précédente, il faudra remplacer dans l'option **--wiht-pgsql-src**, le terme **XXX** par le nom de répertoire convenable. Regardez pour celà le nom du répertoire contenu dans c:\msys\1.0\home. Il ne reste plus qu'à compiler et installer PostGIS:

```
cd ~/sources/PostGIS/postgis-1.0.0-rc6

make && make install
```

Ca y est, les outils sont installés!!! Il ne reste plus qu'à définir l'environnement de travail de PostgreSQL. Ce qui fera l'objet du prochain chapitre.

Chapitre 3. Paramétrer PostgreSQL

Dans le chapitre précédent, nous avons effectué les diverses installations nécesssaires de nos outils. Afin de tirer profit de notre installation, il est maintenant temps de passer à la finalisation qui va concerner le serveur PostgreSQL. Afin que ce dernier soit fonctionnel, il est nécessaire de définir diverses variables d'environnement de PostgreSQL, de définir le super -utilisateur de PostgreSQL qui fera dans un premier temps son initialisation, d'installer le service etc...Nous verrons ces divers points en détail dans ce chapitre.

1. Les variables d'environnement de PostgreSQL

Que vous soyez sous Windows XP ou Windows 2000, créez les variables d'environnement suivantes - propres à PostgreSQL -:

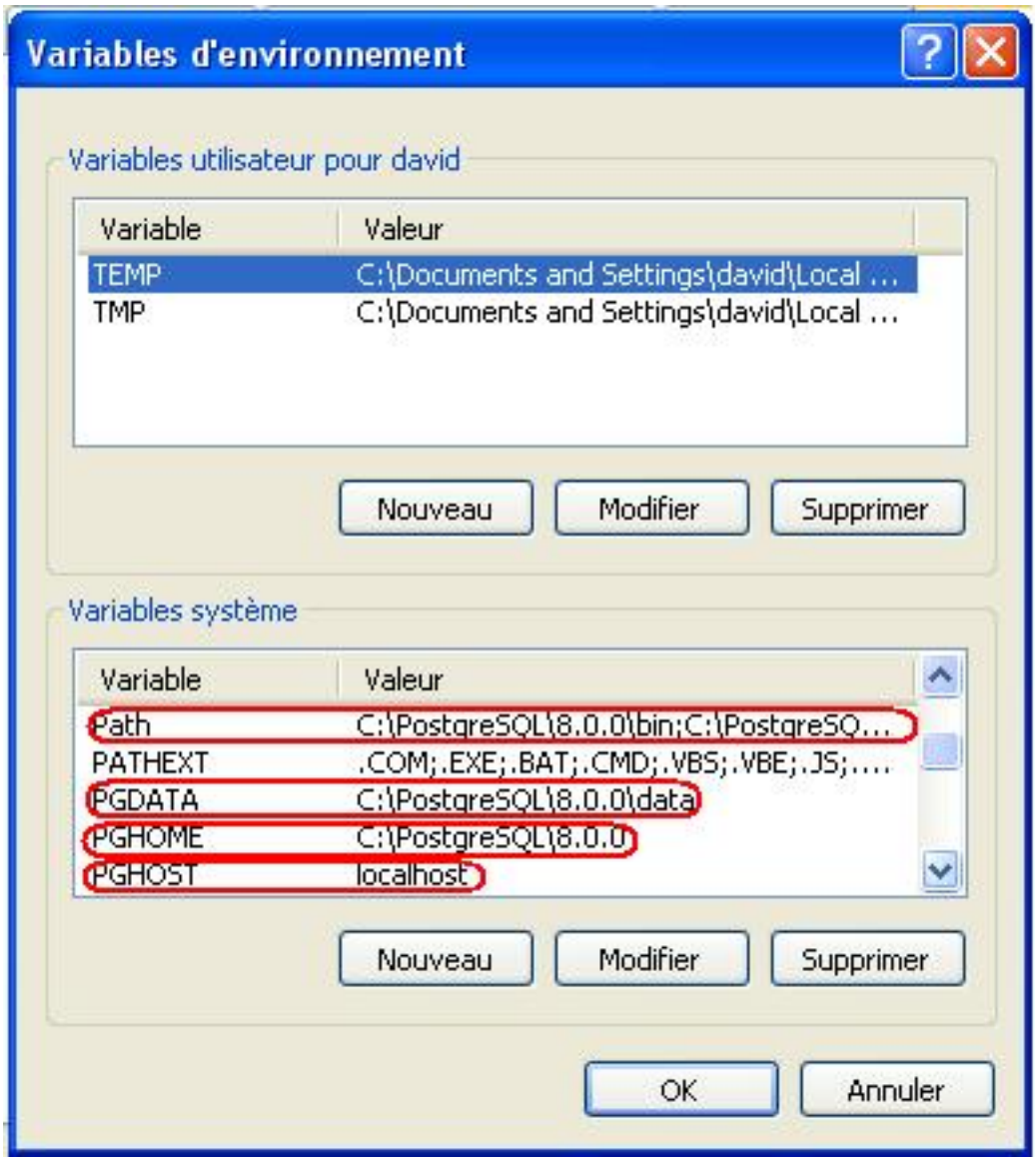
- **PGHOME = C:\PostgreSQL\8.0.0**
- **PGDATA = C:\PostgreSQL\8.0.0\data**
- **PGHOST = localhost**
- **PGPORT = 5432**

`PGHOME` permettra à PostgreSQL de savoir où est installé la distribution. `PGDATA` lui permettra de connaître le répertoire-racine des données des diverses bases [...]. Etant donné que PostgreSQL est un "serveur", `PGHOST` lui permettra de savoir à quel hôte se connecter et sur quel port - défini par `PGPORT` -, il doit écouter. L'intérêt de ces variables sera vu plus tard.

Ajoutez aussi au début de votre variable **PATH** les chemins d'accès **C:\PostgreSQL\8.0.0\bin;C:\PostgreSQL\8.0.0\lib;C:\PostgreSQL\8.0.0\lib\postgresql**

Cet ajout en début de votre variable PATH permettra à PostgreSQL d'accéder aux divers utilitaires accompagnant la distribution (cf. répertoire bin) mais aussi aux diverses librairies - fichiers *.dll - (cf. répertoires de lib). Il sera aussi possible par la suite d'accéder aux librairies de Geos et de Proj.

Figure 3.1. Les variables d'environnement de PostgreSQL



2. Le super-utilisateur de PostgreSQL

Nous allons ici voir quelques règles régissant les droits avec le serveur PostreSQL puis nous créerons - si besoin est - ce super-utisateur.

2.1. Le super-utilisateur?

Comme nous l'énoncions, au début de ce chapitre, la prochaine concerne l'initialisation du répertoire qui accueillerons nos futures bases de données: PGDATA. Ce répertoire pour l'instant n'existe pas encore. Or avec PostgreSQL, seul le super-utilisateur a le droit de le créer.

Mais alors qui est ce super-utilisateur?

Ce super-utilisateur - comme première propiété - doit être un utilisateur physique de votre machine. Oui mais depuis la version 8.0.0 de PostgreSQL, les programmeurs de PostgreSQL pour des raisons de sécurité (impliquant des failles/fuites de sécurités selon les droits accordés à cet utilisateur sous Linux qui peuvent être dûe à des attaques de requêtes SQL par le réseau) ont imposés que cet utilisateur devait avoir des droits limités sur la machine sur lequel tourne PostgreSQL.

Or sous Windows, on distingue deux types d'utilisateurs:

- ceux qui ont des droits administratifs - les administrateurs - ils peuvent tout faire sur la machine, installer/désintaller des logiciels, lancer des services, supprimer des utilisateurs, modifier les variables d'environnement physiques etc...

- ceux qui ont des droits limités: ils peuvent accéder aux logiciels mise à leur disposition par les administrateurs, ne peuvent accéder à certains répertoires, etc...

Le super-utilisateur -comme seconde propriété -sera donc un utilisateur ayant des droits limités

Oui mais - comme troisième propriété - comme son titre l'indique, il sera le grand manitou, le gourou au sens de PostgreSQL de la secte des utilisateurs de PostgreSQL. Il a le droit de créer des utilisateurs, d'étendre/restreindre leur droit de lecture/modification aux bases de données du serveur. Il aura aussi le droit de leur accorder le droit de créer ou non des bases par exemple, de leur attribuer des mots de passe etc...En un mot, il aura droit de vie/mort sur ses sujets :-)

Mais ici, nous resterons dans la politique dite du "trusting". Je m'explique. Pour la suite du document je vais supposer que vous êtes administrateur sur la machine sur laquelle vous avez installé PostgreSQL et les autres outils. Nous allons demander au super-utilisateur de nous confier ses supers pouvoirs (à vous) et de faire confiance à toute machine du réseau qui se connecterait au serveur PostgreSQL sans demander de mot de passe à sa majesté.

Il ne reste donc plus en première instance qu'à créer ce fameux super-utilisateur.

2.2. Création du super-utilisateur

Voyez si sur votre machine, s'il n'existe pas déjà un utilisateur ayant des droits limités et possédant un mot de passe. Si ce dernier n'existe pas, créez-le. Attention, il est impératif, que cet utilisateur ait des droits limités sur la machine et qu'il ait un mot de passe.

Avertissement

Pour faciliter la lecture des futures commandes de ce document, j'appelerais postgres le super-utilisateur de PostgreSQL et david un administrateur sous Windows - qui vous correspondra sous Windows. Sous Windows, postgres et david auront comme mot de passe miolze56. Donc pour récapituler - même si je me répète - postgres est donc un utilisateur limité sous Windows et david est administrateur de la machine (vous correspondant) à qui on souhaite conférer les pouvoirs de super-utilisateur de postgres le moment venu. Vos futures commandes seront donc à modifier en fonction des spécifications ici mentionnées. Les lignes de commandes permettant de distinguer les deux fenêtres des utilsateurs commenceront par '# sous david:' (correspondant à votre session) ou bien '# sous postges :' (correspondant à la fenêtre du super-utilisateur que vous aurez plus tard). TOUTES LES COMMANDES SUIVANTES SONT A SAISIR DANS DES FENETRES DOS.

2.3. Pouvoir accéder à la session du super-utilisateur sous DOS sans changer la session en cours

Imaginons que nous soyons connecter sous Windows sous la session de david. La prochaine étape va constituer à initialiser le répertoire `PGDATA`. Ceci se fait en ouvrant normalement une fenêtre DOS sous la session de postgres. Pour éviter de changer de session et accéder plus facilement à cette fenêtre, ceci est possible depuis la session de david.

Il faut pour celà utiliser la commande runas de Windows. Je sais qu'elle existe sous Windows XP mais je ne garantis pas ici son existante sosu Windows 2000. L'obtention de la fenêtre en question se fait en tapant dans une fenêtre DOS:

```
# sous david:
runas /user:postgres cmd
```

Il suffit alors comme demandé de saisir le mot de passe de postgres. Une nouvelle fenêtre DOS - appartenant à postgres - devrait s'ouvir. C'est cette fenêtre qui désignera comme nous l'avons dit '# sous postgres' maintenant.

3. Initialisation de PostgreSQL

L'initialisation consiste à la préparation de vos futures bases de données par défaut. Elle répond aux besoins suivants. Quel encodage choisir par défaut? Quel doit être le mode de connexion par défaut? etc...Mais le reste des configurations des paramètrages du serveur peut par la suite être reprise dans les fichiers .conf de PostgreSQL.

IPour l'initialisation,l faut utiliser l'utilitaire `initdb`.

Dans la fenêtre de postgres,

```
# sous postgres:
initdb -A trust -E SQL_ASCII
```

Par défaut ici, nous utiliserons le jeu d'encodage SQL_ASCII (cf . **-E SQL_ASCII**) et autoriserons toute personne voulant se connecter au serveur par le réseau sur demande de mot de passe (cf. **-A trust**).

4. Démarrer/Arrêter le serveur et autoriser les connexions TCP/IP

Il existe deux types de démarrage soit par la méthode manuelle ou en paramétrant PostgreSQL en tant que service sous Windows. Voyons en détail les deux méthodes. Il vous faudra choisir l'une des deux.

4.1. Démarrage/Arrêt manuel

Le démarrage manuel impose que vous deviez démarrer PostgreSQL à chaque démarrage physique de votre machine.Il faut pour celà, utiliser l'utilitaire `pg_ctl` . Si tel est votre choix, pour démarrer PostgreSQL

```
# sous postgres:
pg_ctl -o -i start
```

L'option **-o -i** permet d'autoriser les connexions TCP/IP. Pour arrêter le serveur, il suffit de taper:

```
# sous postgres:
pg_ctl -o -i stop
```

N'oubliez que si vous choisissez cette méthode, vous devez d'abord taper la commande donnée avec runas si vous démarrez votre machine depuis votre session habituelle (cd. david ici).

4.2. Démarrage automatique

Avec cette méthode - comme sous-entendu -, vous n'aurez pas besoin de vous soucier de démarrer PostgreSQL à chaque démarrage de votre machine. En effet, nous allons ici procéder à l'installation du service Windows, livré avec PostgreSQL. C'est toujours `pg_ctl` que nous utiliserons avec l'option **register**

4.2.1. Installation du service

Nous allons créer le service nommé postgresqlwin32, en autorisant les connexions TCP/IP vaec l'option **-o -i**. C'est le super-utilisateur qui doit avoir le droit sous Windows d'installer les services. Donc nous ferons depuis une fenêtre DOS de david:

```
# sous david:
pg_ctl register -N postgresqlwin32 -U postgres -P miolze56 -o -i
```

Le service est maintenant installé. Il faut maintenant le lancer. Ne pas oublier qu'ici miolze46 est le mot de passe de postgres

4.2.2. Lancer le service

Celà se fait en tapant

```
# sous david:
net start postgresqlwin32
```

5. Devenir soi-même super-utilisateur de PostgreSQL

Le fait d'utiliser PostgreSQL en ayant besoin du super-utilisateur peut devenir à la longue un handicap. En effet, chaque fois que sous votre session habituelle, vous aurez envie d'utiliser les utilitaires de PostgreSQL, vous serez obligé de saisir l'option -U postgres. Le plus simple est de vous rendre super-utilisateur de PostgreSQL.

Il est vrai que je suis en contradiction avec ce que j'ai mentionné dans la section du super-utilisateur (voir seconde propriété) et que normalement vous devriez ne pas pouvoir le faire. Mais je mentionnerais ici la commande qui le permet de manière à vous simplifier la vie et pour vous éviter de trimbaler le -U postgres en question à chaque occasion rencontrée. Que Saint-PostgreSQL me pardonne !!!

Il vous suffira de saisir

```
# sous postgres:
createuser -ad david
```

Voilà vous êtes maitre absolu de votre PostgreSQL. Vous ne saurez plus obligé de saisir la commande avec runas depuis voter session habituelle!!! Le prochain chapitre va concerner PostGIS.

Chapitre 4. PostGIS

Mon but ici n'est pas de m'étendre sur l'utilisation de PostGIS mais juste montrer comment créer une base avec PostGIS et importer des données depuis un fichier shapefile.

1. Créer une base avec PostGIS

La création d'une base - par exemple nommée madatabase - se fait en tapant

```
createdb madatabase
createlang plpgsql madatabase
psql -d madatabase -f %PGHOME%\share\postgresql\lwpostgis.sql
psql -d madatabase -f %PGHOME%\share\postgresql\spatial_ref_sys.sql
```

A COMPLETER