

# GraceTHD-Check - Manuel administrateur Postgis

**DRAFT !!!**

Révision : 01/12/2016 - sby

## Préambule

### Notes

- GraceTHD-Check est un projet GraceTHD-Community.  
<https://github.com/GraceTHD-community>
- Cette documentation est rédigée sur le wiki du Groupe Experts Contrôle sur le Redmine GraceTHD.  
<https://redmine.gracethd.org/redmine/projects/gracethd-check/wiki>
- Rédacteurs : Stephane Byache (Aleno).
- Licence : CC-by-sa v2.0

### Utilité

GraceTHD-Check permet d'augmenter une base de données GraceTHD-MCD avec un schéma gracethdcheck qui comporte des tables et des vues offrant des relevés unitaires d'anomalies (une vue par point de contrôle), des rapports d'anomalies (vues synthétisant les anomalies rencontrées dans toutes les tables pour un statut).

### Structure

Même arborescence que GraceTHD-MCD, et donc Gracelite.

La racine contient les scripts exécutables. Les scripts concernés par ce manuel sont nommés gracethdcheck\_pg\_\*.bat.

sql\_postgis contient les scripts SQL qui sont appelés par les scripts exécutables à la racine.

## I. Installer GraceTHD-Check

### option 1 : à partir du dépôt GraceTHD-Check.

Télécharger

<https://gracethd-community.github.io/GraceTHD-Check/>

### Option 2 : à partir de Gracelite.

#### GraceTHD-Check disponible à partir de Gracelite v0.06

Avec Gracelite, il n'y a pas besoin de télécharger GraceTHD-Check, Gracelite embarque tout. Gracelite est une boîte à outils qui intègre tous les projets GraceTHD. Cette option est préférable.

## II. La grille de remplissage

### II.1 Grille de remplissage conforme

Normalement, vous avez produit une (ou plusieurs) grille de remplissage conforme à votre projet (cahier des charges). Si ce n'est pas le cas, vous pouvez récupérer le modèle de grille de remplissage (adapté à ce jour à 3 cas) et l'adapter.

<http://gracethd-community.github.io/GraceTHD-MOD/>

### II.2 Mettre à jour les fichiers SQL

Chaque table produit un sql (colonne avec une police de petite taille en bleu clair).

Copier cette colonne pour les tables à fournir dans gracethdcheck\_31\_filltab\_insert.sql en remplacement du script par défaut. Bien conserver la ligne

```
SET search_path TO gracethdcheck, public;
```

Copier la colonne pour la grille de remplissage de vos attributs dans gracethdcheck\_32\_fillatt\_insert.sql en remplacement du script par défaut. Bien conserver la ligne

SET search\_path TO gracethdcheck, public;

## II.3 Mettre à jour la grille de remplissage dans la base de données

Il est bien évidemment possible de remettre à jour à tout moment l'intégration de la grille de remplissage. Exécuter ce script : gracethdcheck\_pg\_maj\_remplissage.bat

## III. Configurer config.bat

Dans la section GL\_CONFIG\_DEBUG :

- **GLPAUSE** :
  - GLPAUSE=PAUSE alors des pauses s'activeront pour visualiser l'affichage lors des opérations.
  - GLPAUSE= alors les pauses ne s'activeront pas.

Dans la section GL\_CONFIG\_APPS\_PG\_PATH :

- **GL\_PGBIN** : chemin version le dossier des binaires PostgreSQL (sans \ a la fin)
  - Normalement GL\_PSQL, GL\_PG\_DUMP, GL\_SHP2PGSQL sont dans ce dossier, donc inutile de modifier les valeurs.

Dans la section GL\_CONFIG\_GRACETHD\_POSTGIS\_PROD :

**Configuration serveur.**

- **PGUSER** : l'utilisateur PostgreSQL à utiliser. Attention de bien disposer des privilèges nécessaires (création et suppression de base, ...).
- **PGPASSWORD** : Il est attendu que le password file soit configuré, toutefois il est encore nécessaire de saisir le mot de passe de l'utilisateur à utiliser dans la variable d'environnement PGPASSWORD.
  - <https://www.postgresql.org/docs/current/static/libpq-pgpass.html>
  - Pour des raisons de sécurité, attention aux droits d'accès au dossier où sont déposés les scripts GraceTHD-Check.
- **PGROLE** : rôle postgresql à utiliser.
  - Par défaut postgres, mais il est important de créer un rôle spécifique pour les usages type GraceTHD-Check. Si, par exemple, il y a un partage des responsabilités traduit par un rôle admin et un rôle user, il sera souhaitable de d'installations GraceTHD-Check différentes pour les admin et les utilisateurs.
- **PGHOSTNAME** : host postgresql
- **PGPORT** : port de la base postgresql à utiliser

**Configuration base de données**

- **PGDB** : Base de données Postgis à utiliser
- **PGSCHEMA** : Schéma à utiliser dans la base de données.

- Idéalement ne pas modifier le nom du schéma, laisser gracethd.
- PGSRID : code du système de coordonnées utilise dans cette base de données.
  - Par défaut 2154 (code EPSG du Lambert 93). Si vous utilisez GraceTHD-Check pour les DOM-TOM, pour la France métropolitaine + DOM-TOM, ou tout autre territoire, cette variable est à modifier.
- PGCSVCONF : parametres pour PSQL pour la structure des csv. A priori ne pas modifier : PGCSVCONF= WITH DELIMITER ';' CSV HEADER ENCODING 'UTF8'.

#### Dans la section GL\_CONFIG\_GRACETHD\_POSTGIS\_PATH

L'utilisateur "PostgreSQL" défini par la variable PGUSER doit avoir les droits d'accès aux dossiers où se trouvent les fichiers à importer et exporter (shapefiles, csv, ...). Par défaut il s'agit de dossiers intégrés dans l'arborescence GraceTHD-Check (check, conf, shpcsv-in, shpcsv-out). Ces variables laissent la possibilité de configurer d'autres chemins. Si on utilise un partage qui accueille ces dossiers, il n'y a que la variable GLPGPATHSHARE à modifier.

- **GLPGPATHSHARE** : dossier pour lequel l'utilisateur référence dans la variable PGUSER a les droits de lecture et d'écriture.
- PGSHPINPATH : Dossier comportant les shp/csv à importer dans Postgis.
- PGSHPINPATH=%GLPGPATHSHARE%\shpcsv-in
- PGSHPOUTPATH : Dossier comportant les shp/csv à importer dans Postgis.
- GL\_PG\_DUMP\_FILE : fichier produit par PG\_DUMP par défaut.
- GLPGCONFPATH : Chemin où l'utilisateur PGUSER peut lire et écrire des fichiers de configuration.

#### Dans la section GL\_CONFIG\_GRACETHDCHECK\_POSTGIS

Il est possible de configurer des dossiers spécifiques pour l'usage de GraceTHD-Check. Par défaut ce sont les mêmes que pour GraceTHD-MCD.

De même il est possible de définir un nom de base de données et de schémas spécifiques pour GraceTHD-Check. Par défaut ce sont les mêmes que pour GraceTHD-MCD.

## IV. Déployer

### Rôles et privilèges

Les scripts vont créer de nombreuses vues qui doivent être accessibles aux utilisateurs et que les utilisateurs doivent pouvoir recréer. Il est donc hautement conseillé d'appliquer des privilèges aux utilisateurs au niveau du schéma gracethdcheck. Certaines tables comme les listes de valeurs (tables l\_\*) peuvent être protégées.

Les utilisateurs doivent pouvoir créer des bases de données temporaires, pour contrôler des données avant intégration. Il est également possible de déployer GraceTHD-Check sur une base GraceTHD-MCD qui est le référentiel (base de production). Dans ce cas il est souhaitable de protéger plus finement cette base. Dans l'absolu le référentiel (base de production) est sur un serveur de production, les bases de contrôle plutôt sur un serveur d'intégration. Il peut être envisageable que les requêtes de contrôle puissent consommer beaucoup de ressources, par conséquent l'utilisation de serveurs distincts peut-être une bonne stratégie.

Il peut également être utile de distinguer un rôle développeur qui autorise des modifications sur t\_c\_cat et les tables t\_ct\_code\_\*. Mais il est préférable de fournir aux développeurs des points de contrôle des bases de test, des jeux de données test et un partage de test, idéalement sur un serveur de développement. Lors des tests d'optimisation des requêtes il est courant de déclencher des requêtes consommatrices. Un serveur de développement n'a pas les mêmes besoins d'administration.

### Administration

Les bases de contrôle, qui doivent être créées à chaque nouvelle réception à contrôler, peuvent être conservées, archivées, ou simplement détruites après chaque utilisation. C'est une décision propre à chaque environnement.

Des scripts permettent d'exporter certaines tables en CSV ou en SQL.

- `gracethdcheck_pg_export_csv.bat`
- `gracethdcheck_pg_export_sql.bat`

Ils sont utiles pour les développeurs qui souhaitent exporter les données pour pouvoir transmettre tout ou partie.

Les tables GraceTHD-Check contiennent peut de données mais beaucoup de vues. Les dumps ne sont donc pas lourds.

## Intégration dans le SI

GraceTHD-Check n'est qu'un moteur livrés avec les scripts élémentaires pour le déploiement et l'administration des données. Il est bien évidemment possible d'intégrer cela dans un robot qui réceptionne automatiquement, gère la file d'attente, déclenche les contrôles, envoie ou publie (web ?) les rapports d'anomalies, etc. Il est possible d'interfacer cela avec un BPM ou outil de workflow. Le fonctionnement en base autorise toutes les formes d'urbanisation SI.

Si des développements sont envisagés, il est pertinent de se rapprocher de la communauté GraceTHD pour évoquer le besoin ou les projets. Il est absolument certain qu'un projet open source de type robot serait bénéfique pour le plus grand nombre. Toutefois à ce jour il n'est pas envisagé que la communauté GraceTHD ne développe des outils, auquel cas il serait pertinent d'envisager un autre projet open source externe à la gouvernance GraceTHD.

<https://redmine.gracethd.org>

## Développement

Pour ceux qui souhaitent développer ou modifier des points de contrôle, voir le manuel dédié aux développeurs.

## IV. Tester

Choisir un nom de base de test comme par exemple :

`SET GLCTPGDB=%PGDB%ctrl`

Cet exemple utilise la valeur de la variable PGDB et ajoute ctrl, soit un nom de base qui est par exemple `gracethd20ctrl`.

Le script `gracethdcheck_pg_create_db.bat` permet de créer une base de données, ou d'ajouter la structure GraceTHD-Check sur une base de données GraceTHD existante. Normalement c'est fait !

Pour plus de détails, voir le manuel utilisateur.