

# GraceTHD-Check - Manuel utilisateur Postgis

Révision : 30/05/2017 - sby

Applicable à partir de GraceTHD-Check v0.01-alpha6

- GraceTHD-Check est un projet GraceTHD-Community.  
<https://github.com/GraceTHD-community>
- Rédacteurs : Stephane Byache (Aleno).
- Licence de cette documentation : CC-by-sa v2.0
- GraceTHD-Check est sous licence libre : GPLv3

## Préambule

Cette documentation est rédigée sur le wiki du Groupe Experts Contrôle sur le Redmine GraceTHD.

<https://redmine.gracethd.org/redmine/projects/gracethd-check/wiki>

Elle fait partie d'un ensemble de documentations.

- GraceTHD-Check - Manuel utilisateur Postgis
- GraceTHD-Check - Manuel utilisateur avancé Postgis
- GraceTHD-Check - Manuel administrateur Postgis
- GraceTHD-Check - Manuel développeur
- GraceTHD-Check - Index des scripts
- GraceTHD-Check - Résumé du catalogue des points de contrôle

Notes :

- Préalablement à l'utilisation de GraceTHD-Check, un administrateur GraceTHD-Check doit être désigné et mettre en place GraceTHD-Check (cf. "GraceTHD-Check - Manuel administrateur Postgis").
- Un utilisateur initié pourra continuer ses lectures avec "GraceTHD-Check - Manuel utilisateur avancé Postgis".

## Procédure d'exécution simplifiée.

**Préalable** : Il est nécessaire que l'administrateur GraceTHD-Check ait, une fois pour toutes, correctement configuré GraceTHD-Check. Voir "GraceTHD-Check - Manuel administrateur Postgis".

### Etape 1 : Configuration utilisateur

- **.\conf\config\_db\_pg.bat** : à la ligne "SET PGDB=gracethd20ct" remplacer "gracethd20ct" par le nom de base de données que vous souhaitez créer pour le contrôle.  
*Note : Une base de contrôle ne doit pas être le référentiel, c'est une base temporaire. Il peut être judicieux de nommer cette base avec la référence du livrable à contrôler. Le code du livrable est ainsi par automatiquement le nom de la base, les rapports portent l'information dans les attributs ct\_liv et les exports portent ainsi la référence du livrable.*
- **config.bat** : si vous rencontrez un problème, vous pouvez configurer GraceTHD-Check pour qu'il opère une pause après chaque opération et laisse donc le temps de consulter l'affichage.
  - Remplacer "SET GLPAUSE=" par "REM SET GLPAUSE="
  - Remplacer "REM SET GLPAUSE=PAUSE" par "SET GLPAUSE=PAUSE"
- Vous pouvez également activer le mode debug pour certains scripts en particulier. Par exemple il peut y avoir une infinités de problèmes à l'import des shapefiles et csv, le script gracethdcheck\_pg\_import\_shpcsv-in\_psql.bat est en mode debug par défaut (pause après chaque chargement). Pour supprimer ce mode debug (donc ne plus avoir les pauses), il suffit de commenter la ligne "CALL:DEBUG" en remplaçant par "REM CALL:DEBUG".

### Etape 2 : Copie des shapefiles et csv

Copier vos shapefiles et CSV dans le dossier référencé par la variable PGSHPINPATH, par défaut shpcsv-in.

Attention, l'utilisateur postgresql référencé par la variable PGUSER (et dans le passfile) doit avoir les droits nécessaires.

*Note: Des scripts de contrôle des shapefiles et csv préalables à l'intégration Postgis et visant à s'assurer du bon chargement sont en cours de préparation.*

## Etape 3 : Exécuter gracethdcheck\_pg.bat

Ce batch ne fait qu'exécuter d'autres batchs permettant de créer la base et les schémas (gracethd, gracethdcheck et gracethdcheckpub), de charger les données et d'exporter les résultats (droits nécessaire pour le PGUSER). Il est possible d'exécuter ces opérations pas à pas sans utiliser ce script (voir "GraceTHD-Check - Manuel utilisateur avancé Postgis"). Le temps d'exécution est variable selon le volume de données, les performances du serveur, les points de contrôle à exécuter, etc.

Par défaut la référence du livrable est le nom de la base de données. Il est possible de configurer manuellement un code de livrable dans la table `t_ct_conf` au niveau de la variable `ct_1_liv`. **gracethdcheck\_pg\_conf\_liv.bat** permet de saisir cela en mode console. Ce code apparaît dans les rapports d'anomalies.

## Etape 4 : Consulter les anomalies

Important, la consultation se fait via des vues matérialisées. Les vues matérialisées offrent un affichage quasi instantané. Normalement à ce stade elles sont à jour. S'il y a des modifications dans les données GraceTHD de la base qui est contrôlée, il faut rafraichir ces vues avec le script `gracethdcheck_pg_refresh.bat`.

`gracethdcheck-mcd-v*.pdf` permet de comprendre la définition de chaque attribut et de son fonctionnement.

### Nommage des rapports et synthèses d'anomalies

Rapports d'anomalies :

- `mv_ct_anom` : nombre d'objets concernés par point de contrôle ayant détecté des anomalies.
- `v_ct_pub_anom_[statut]` : la même chose que `mv_ct_anom`, mais filtré par statut (DIA, PRE, AVP, etc.).

Synthèses d'anomalies par tables :

- `mv_ct_synt_[table]` : détail des anomalies pour les tables non spatialisables
- `mvs_ct_synt_[table]` : détail des anomalies pour les tables spatiales et spatialisables.

### Avec pgAdmin

Connectez-vous sur la base de données qui a été créée.

Schémas - `gracethdcheckpub` - vues

### Avec QGIS

Le projet fourni par défaut est `.qgis/qgs/gracethdcheck_qgis2.14_postgis.qgs`

Il est naturellement nécessaire de l'adapter à votre serveur et à la base que vous souhaitez utiliser. La solution la plus rapide est d'ouvrir une copie de ce qgs avec votre éditeur de texte. Faire un rechercher-remplacer de la chaîne suivante par vos propres paramètres.

```
dbname='gracethd20ct' port=5432 user='postgres' sslmode=disable
```

Le groupe de couches **GraceTHD-Layers (PostgreSQL)** affiche les objets de GraceTHD tels que proposés dans GraceTHD-Layers. Ce sont des vues dynamiques. Si vous modifiez des données des tables, les couches connectées sur des vues affichent les données à jour. A noter qu'il peut parfois être nécessaire dans l'affichage de la table de rafraichir avec l'icône avec des flèches circulaires.

Le groupe de couches **GraceTHD-Layers (PostgreSQL-Materialized views)** affiche la même chose que GraceTHD-Layers (PostgreSQL) mais avec des vues matérialisées. Les performances sont meilleures, mais s'il y a des modifications des données, il faut rafraichir les vues matérialisées avec `gracethd_pg_refresh.bat`.

Le groupe de couches **GraceTHD-Check (PostgreSQL-Materialized views)** affiche les anomalies "spatialisables". Les anomalies non spatialisables sont accessibles également.

A noter que vous pouvez choisir de modifier le comportement de GraceTHD-Check en modifiant les valeurs de la table d'exécution via `v_ct_exe`. Par exemple si vous ne souhaitez pas exécuter certains points de contrôle pour certains statuts sur cette base, vous pouvez changer les valeurs puis rafraichir les vues matérialisées avec `gracethdcheck_pg_refresh.bat`.

## Etape 5 : Envoyer les anomalies

Les anomalies publiées sous forme de vues dans le schéma `gracethdcheckpub` sont normalement exportées dans le dossier `.check\` :

- un dossier nommé **gracethdcheckpub\_[nom de la base]\_aammjj\_hhmmss** comporte les vues en shapefiles et csv
- un fichier zip nommé **gracethdcheckpub\_[nom de la base]\_aammjj\_hhmmss.zip** comporte le contenu du dossier.

Pour les rapports d'anomalies (v\_ct\_pub\_anom\_\*), il suffit de ne conserver que celui du statut du livrable contrôlé.

Pour les synthèses d'anomalies par tables, il y a un attribut par statut. Les utilisateurs doivent donc filtrer les anomalies selon le statut du livrable.

## **Etape 6 : Analyses complémentaires**

*Note: concerne plutôt les utilisateurs avancés.*

Possibilité d'utiliser la vue v\_in\_1\_rowsmd5 pour identifier les entités supplémentaires, celles qui ont disparu et surtout celles qui modifiées par rapport à la précédente livraison. Ces contrôles sont en cours d'implémentation, toutefois à ce stade il est possible de procéder à ces opérations via de simples jointures entre v\_in\_1\_rowsmd5 et v\_in\_1\_rowsmd5 de la version N-1.

Possibilité d'utiliser la vue v\_in\_rowsnb pour identifier les tables qui n'auraient pas la même quantité d'entités par rapport à une autre version, voire à la même version mais sous un autre format.