

GraceTHD-Check - Manuel développeur

Révision : 16/05/2017 - 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 - Présentation
- 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 en relation avec le DBA PostgreSQL. L'administrateur GraceTHD-Check peut être DBA, mais pas obligatoirement, par contre il doit comprendre GraceTHD-Check et doit pouvoir assurer un support à ses utilisateurs. Un manuel utilisateur avancé permet également à un utilisateur de partager cette responsabilité.

Procédure développeur

Note: Avant de consulter ce document, il est souhaitable de consulter au préalable "GraceTHD-Check - Présentation". De même un développeur devrait a minima consulter "GraceTHD-Check - Manuel utilisateur Postgis", et idéalement l'ensemble de la documentation.

Conformité des points de contrôle

Développement de points de contrôles « conformes » GraceTHD-Check.

- Nommage du point de contrôle « spécifique ». Ex : ad_code_9_s00123
- Soumettre des points de contrôle sous la forme d'un t_ct_cat_user.csv.
- Un point de contrôle validé intègre GraceTHD. Ex : ad_code_1_s09876

Développer un point de contrôle pour une utilisation interne, et en vue de le faire référencer par le projet GraceTHD-Check.

- Vous pouvez référencer la définition d'un point de contrôle dans la table t_ct_cat_user. Dans ce cas utiliser un code 9 au lieu du 1. Ex : ad_code_9_s00123.
- Cette seule définition peut déjà être soumise au projet GraceTHD-Check via une demande sur le Redmine de GraceTHD. Idéalement il peut s'agir d'un extrait de t_ct_cat_user en csv voire en SQL. Mais ça peut également être une simple demande écrite avec une définition du point de contrôle. A la création d'une base de données, t_ct_cat sera augmenté de t_ct_cat_user, par conséquent les tables en relation avec t_ct_cat disposeront de vos ajouts.
- Si elle est validée, un code définitif sera attribué. Ex : ad_code_1_s00123. C'est un moyen de contribuer au projet et quelque part de réserver au plus tôt le code définitif.
- Vous pouvez également développer ce point contrôle, pour Postgis et / ou Spatialite. Dans ce cas l'intégrer dans la table t_ct_code_pgs_user (Postgis) et / ou t_ct_code_spl_user (Spatialite). ATTENTION !! Le développement de GraceTHD-Check est suspendu sur Spatialite depuis la version 0.01-alpha2.
- Vous pouvez soumettre ce code via une demande sur le Redmine GraceTHD. Ça peut prendre la forme d'un ou des SQL ou CSV conformes à t_ct_code_pgs et / ou t_ct_code_spl. Mais à défaut ça peut être également la fourniture de la seule requête

SQL de contrôle sans les propositions d'attributs de la table t_ct_code*.

Structure du code SQL

La structure des points de contrôle doit toujours être identique.

- La première ligne doit être le code du point de contrôle entre /* */
- La partie SELECT doit être absolument identique pour chaque requête de contrôle.
- Utiliser un outil qui permet de gérer correctement la saisie multi-lignes en base et qui respecte les indentations du code. pgAdmin gère cela correctement pour l'édition.
- Pour Postgis il est possible d'utiliser une syntaxe PL/pgSQL, toutefois tant que le SQL suffit c'est préférable car il est plus simple de porter du SQL Postgis sur Spatialite, et inversement, que du PL/pgSQL. A réserver pour les cas nécessaires.

Exemple Postgis

Attribut ad_code ayant une valeur NULL

```
/*ad_code_1_r00001*/
SELECT
    'ad_code_1_r00001'::text AS ct_code,
    NULL::text AS ct_detail,
    *
FROM t_adresse
WHERE ad_code IS NULL
;
```

Exemple Spatialite

```
/*ad_code_1_r00001*/
SELECT
    'ad_code_1_r00001' AS ct_code,
    NULL AS ct_detail,
    *
FROM t_adresse
WHERE ad_code IS NULL
;
```

Architecture GraceTHD-Check

La table t_ct_conf

t_ct_conf permet de stocker des variables réutilisables dans les requêtes des points de contrôle. Les utilisateurs peuvent ainsi adapter le comportement de certains points de contrôle à leur contexte. t_ct_conf est livrée avec les variables validées par le projet GraceTHD-Check.

t_ct_conf_user permet de stocker les variables modifiées pour le contexte. Toutefois ce n'est qu'une forme de sauvegarde, c'est toujours t_ct_conf qui est utilisée. t_ct_conf doit être mise à jour (sans écraser les autres variables) avec les variables de t_ct_conf_user. Exécuter : gracethdcheck_pg_update_from_user.bat

Les tables _user

t_ct_cat_user et t_ct_code_pgs_user.

- Même principe que pour t_ct_conf_user. Ça permet d'isoler son propre travail, de disposer aisément d'un backup pour reporter cela lors d'un déploiement de mise à jour de GraceTHD-Check.

Les tables de codes (t_ct_code_xxx)

C'est la table qui stocke le code SQL des points de contrôles unitaires.

Note: t_ct_code_spl n'est plus maintenue, toutefois elle est à disposition de la communauté.

Structure t_ct_code_pgs

- ct_pgs_code : le code unique du point de contrôle dans la table. Devrait être égal à ct_pgs_cat_code, sauf si c'est un point de contrôle en cours de test et non référencé dans le catalogue.
- ct_pgs_cat_code : le code du point de contrôle dans le catalogue de points de contrôle (t_ct_cat).
- ct_pgs_statut : le statut de ce point de contrôle (l_ct_statut).
- ct_pgs_version : la version de l'implémentation du point de contrôle.
- ct_pgs_source : la source du point de contrôle, idéalement une adresse mail.
- ct_pgs_date : la date de la dernière révision du point de contrôle.
- ct_pgs_commentaire : commentaires.
- ct_pgs_script : le script du point de contrôle. Pour Postgis et Spatialite, le début du script doit être /*[le code point de contrôle]*/ (cf. exemple).

La table t_ct_exe_pgs

Cette table permet de définir quels points de contrôle doivent être exécutés aux différents statuts (PRE, DIA, AVP, ...) et selon le type de cas (Réseau FTTH concessif, etc.). La construction de t_ct_exe_pgs se fait à la création de la base.

1. Un point de contrôle est rattaché à un attribut principal. Par extrapolation des attentes de rendus de t_ct_conf_fillatt (grille de remplissage), les points de contrôle récupèrent dans un premier temps ces statuts.
2. Ensuite la table t_ct_conf_exe_pgs met à jour t_ct_exe_pgs. Cette table est livrée avec GraceTHD-Check. Un de ses objectifs principaux sera de propager la prise en compte ou non de points de contrôle selon la version du MCD. Un point de contrôle sur un attribut non présent dans une version de GraceTHD-MCD ne peut pas être exécuté.
3. Ensuite la table t_ct_exe_pgs_user met à jour t_ct_exe_pgs avec les valeurs que l'utilisateur souhaite appliquer systématiquement.
4. GraceTHD-Check ne fait pas appel directement à t_ct_exe_pgs, mais à la vue v_ct_exe. Ceci offre certaines souplesses comme par exemple : toujours pointer vers la même vue quelle que soit le SGBD, modifier la vue pour modifier facilement le comportement de GraceTHD-Check. Par exemple quelqu'un souhaite exécuter seulement les contrôles métier, il lui suffit de filtrer le type.

Les valeurs utilisables dans les attributs "statut" sont définies dans la table (liste) l_ct_exe.

gracethdcheck_qgis2.14_postgis.qgs permet d'éditer facilement la table t_ct_exe_pgs via la connexion sur la vue v_ct_exe (vue éditabile).