

pour comprendre le présent et construire un avenir durable

Avril 2013

Version 1.2



## Fichiers fonciers 2009 et 2011 Résolution des problèmes d'utilisation du plugin d'export des données PostgreSQL

# Sommaire

Introduction.....	3
Objectif de la fiche.....	3
Pré-requis.....	3
Après le démarrage de pgAdmin, le plugin d'export n'apparaît pas dans la liste des plugins.....	4
Diagnostic.....	4
Sources possibles du problème.....	4
Vérification du nombre de pgAdmin installés.....	4
Vérification de la bonne installation du plugin.....	5
Le plugin s'affiche, mais les données exportées ne sont pas valides.....	6
Diagnostic.....	6
Sources possibles du problème.....	6
À faire avant tout !.....	6
Renseigner un nom de fichier de sortie valide.....	6
Enregistrer les données sur le disque local.....	6
Corriger les erreurs de géométrie pour certains départements.....	7
Couper les tables trop volumineuse ( cas particulier du département 75).....	9
Annexe 1 : comment exécuter une requête dans pgAdmin ?.....	10
Étape 1 : ouvrir une fenêtre de requête dans pgAdmin.....	10
Étape 2 : écrire une requête SQL dans la fenêtre.....	10
Étape 3 : exécuter de la requête et visualiser le résultat.....	10
Annexe 2 : requêtes pour l'export des tables trop volumineuses.....	11
Rappel.....	11
Requêtes de séparation de table.....	11

Version	Date	Auteurs	Commentaire
1.0	14 juin 2012	Rémi BOREL	Document initial
1.1	3 août 2012	Rémi BOREL	Ajout du paragraphe « Enregistrer les données sur le disque local »
1.2	4 avril 2013	Rémi BOREL	Amélioration de la correction pour les géométries invalides

# Introduction

## Objectif de la fiche

Cette fiche a pour objectif de **résoudre des problèmes** liés à l'utilisation du plugin pgAdmin d'export des données. Ces problèmes peuvent être de plusieurs types :

- le plugin ne s'affiche pas,
- l'export des données ne fonctionne pas,
- l'export des données a l'air de fonctionner mais les tables sont inexploitables.

## Pré-requis

Avant d'utiliser cette fiche, il est important d'avoir suivi la procédure d'installation normale de PostgreSQL et du plugin.

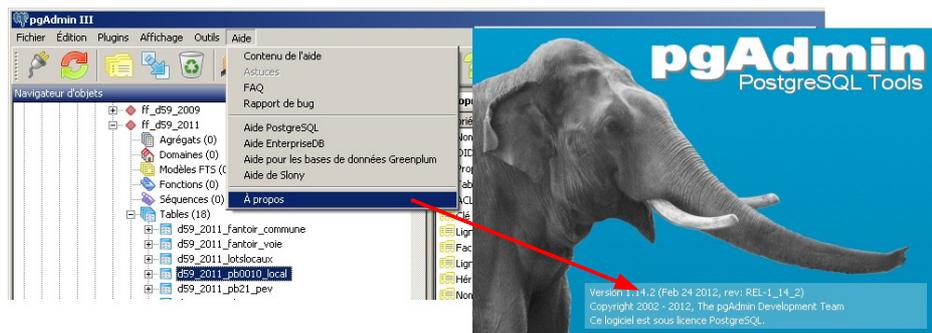
Il faut disposer a minima des versions :

- **1.14** pour pgAdmin,
- **1.2** pour le plugin d'export du CETE Nord Picardie.

Si ce n'est pas le cas, consulter le document correspondant dans la rubrique « Aide technique pour l'exploitation des données » du portail fichiers fonciers<sup>1</sup>.

### Vérification de la version de pgAdmin

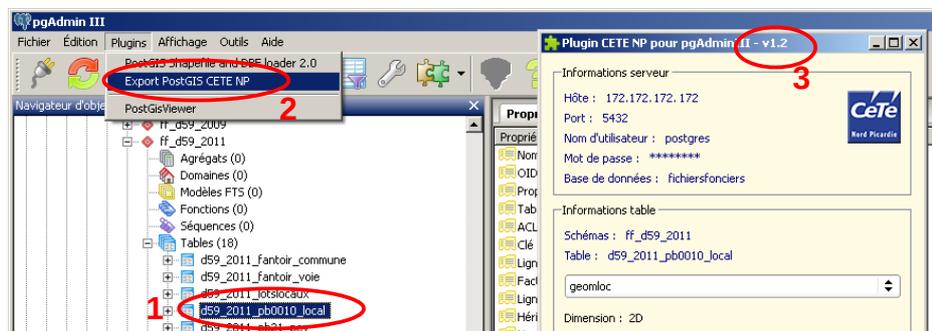
Après l'ouverture de pgAdmin, aller dans « Aide » puis « A propos ».



Accès à la version de pgAdmin

### Vérification de la version du plugin

Après l'ouverture de pgAdmin, se **placer sur une table**, puis dans le **menu « Plugins »** et « Export PostGIS CETE NP ». Le numéro de version du apparaît dans la barre supérieure du plugin.



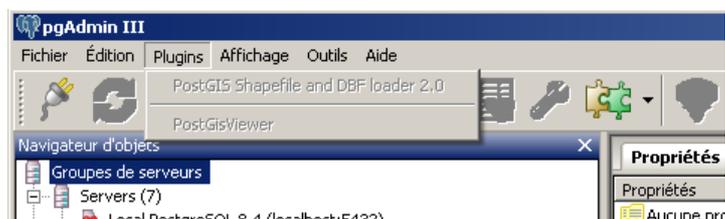
Accès à la version du plugin

1 <http://geoinfo.metier.i2/aide-technique-pour-l-exploitation-r757.html>

# Après le démarrage de pgAdmin, le plugin d'export n'apparaît pas dans la liste des plugins

## Diagnostic

Le plugin « Export PostGIS CETE NP » n'apparaît pas dans le menu « Plugins » de pgAdmin.



Plugin d'export non présent dans la liste

## Sources possibles du problème

Ce problème se présente si :

- plusieurs versions de pgAdmin sont installées sur le poste,
- et /ou le plugin n'est pas installé,
- et /ou le plugin est mal installé.

Les paragraphes suivants proposent des solutions de résolution des problèmes. À chaque étape, **relancer pgAdmin** afin de vérifier que le plugin fonctionne.

## Vérification du nombre de pgAdmin installés

pgAdmin peut être installé de deux manières différentes sur un poste de travail :

- soit en même temps que PostgreSQL,
- soit de manière « autonome ».

L'installation « autonome » est destinée aux utilisateurs « clients ». Ceux-ci n'ont aucune raison d'avoir PostgreSQL installé sur leur poste.

Pour ne pas créer « d'interférences » avec le plugin, il est nécessaire que pgAdmin ne soit installé **qu'une seule fois sur un poste de travail**.

### *Installation en même temps que PostgreSQL*

On peut vérifier que pgAdmin est installé en même temps que PostgreSQL en regardant dans le dossier « Program Files ».

Sous Windows XP, il doit exister un dossier :

**C:\Program Files\PostgreSQL\9.1**

dans lequel il doit y avoir un sous dossier :

**pgAdmin III**

### *Installation autonome*

On peut vérifier que pgAdmin est installé de manière autonome en regardant également dans le dossier « Program Files ».

Sous Windows XP, il doit exister un dossier :

**C:\Program Files\pgAdmin III**

dans lequel il doit y avoir un sous dossier correspondant au numéro de version :

par exemple : **1.14**

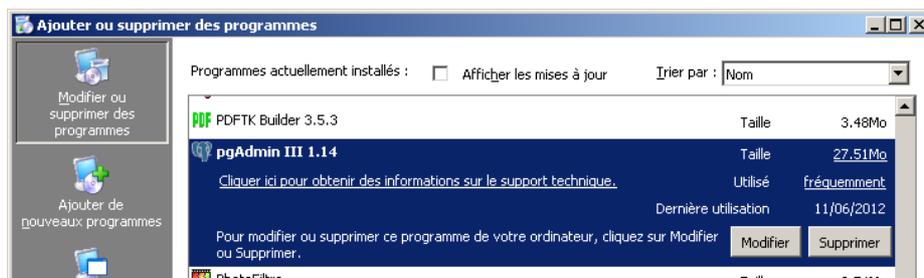
# Après le démarrage de pgAdmin, le plugin d'export n'apparaît pas dans la liste des plugins (suite)

## Vérification du nombre de pgAdmin installés (suite)

**Ne garder qu'une seule version de pgAdmin !**

Si pgAdmin est déjà installé via PostgreSQL, il faut **désinstaller** la ou les versions autonomes de pgAdmin.

Pour cela, passer par l'utilitaire Windows « Ajout / suppression de programmes ».



Fenêtre d'ajout / suppression de programmes

## Vérification de la bonne installation du plugin

Il est impératif d'installer le plugin dans le pgAdmin que l'on souhaite utiliser, que ce soit celui installé avec PostgreSQL ou en version autonome.

Si j'utilise pgAdmin	Alors le dossier d'installation du plugin est
installé avec PostgreSQL	C:\Program Files\PostgreSQL\9.1\bin
en version autonome	C:\Program Files\pgAdmin III\1.14

## Vérification de la bonne installation du plugin

Pour vérifier si le plugin est **bien installé**, il faut se placer dans le dossier indiqué dans le tableau ci-dessus. Celui-ci doit contenir :

- un sous dossier nommé « cete »,
- un fichier « postgis\_export\_cete.ini » dans le sous dossier « plugins.d ».

## Vérification de la configuration du plugin

**⚠ Avant** d'effectuer la manipulation suivante, relancer pgAdmin pour vérifier si le plugin est disponible. Il devrait fonctionner normalement.

Pour vérifier la configuration du plugin, ouvrir le fichier « postgis\_export\_cete.ini » avec un éditeur de texte.

Dans ce fichier, remplacer la chaîne de caractères **\$\$PGBINDIR** par le dossier indiqué dans tableau ci-dessus. Ce remplacement doit être effectué 2 fois, aux lignes « Command » et « KeyFile ».

**Attention** : il ne doit y avoir aucun espace avant et après le signe « = ».

# Le plugin s'affiche, mais les données exportées ne sont pas valides

---

## Diagnostic

Les **données** exportées sont **non valides** :

- quand il est impossible de les ouvrir dans un logiciel SIG,
- quand les fichiers sont anormalement peu volumineux ou même absents.

Généralement, les problèmes d'export se produisent pour les formats géographiques (Mapinfo, ESRI), et non pour le format CSV.

---

## Sources possibles du problème

Si les données sont invalides, une erreur s'est produite pendant l'export. Ces erreurs peuvent être liée :

- à une mauvaise version du plugin (rappel : version 1.2 au minimum),
- à un problème dans le nom du fichier de sortie,
- à des erreurs de géométries inhérentes aux données,
- au fait que la table PostgreSQL est trop volumineuse.

---

## À faire avant tout !

Le **mode « débogage »** du plugin permet d'identifier un certain nombre d'erreurs. Pour tout problème d'exploitation de données exportées dans les formats Mapinfo ou ESRI, suivre la procédure suivante :

- lancer le plugin depuis la table à exporter,
- sélectionner une géométrie, de préférence polygonale,
- ne choisir **que** le format de sortie **ESRI**.
- se placer en mode « débogage »,
- enregistrer le rapport d'erreur.

---

## Renseigner un nom de fichier de sortie valide

Les noms du fichier et du dossier de sortie ne doivent **pas contenir de parenthèses** ouvrantes et fermantes.

De manière générale, il est déconseillé d'utiliser des caractères autres que les lettres non accentuées, les chiffres et le caractère « \_ ».

---

## Enregistrer les données sur le disque local

Le dossier de sortie doit être localisé dans le **disque local** de l'ordinateur (généralement le disque [C:](#)).

Des problèmes ont été constatés pour des données exportées sur disque dur externe ou disque réseau

# Le plugin s'affiche, mais les données exportées ne sont pas valides (suite)

## Corriger les erreurs de géométrie pour certains départements

**Introduction** De manière générale, certains contours de polygones peuvent avoir des **géométries invalides** dans PostgreSQL. Il peut s'agir par exemple d'une auto-intersection. Ce cas se présente si le polygone est numérisé comme un « 8 ».

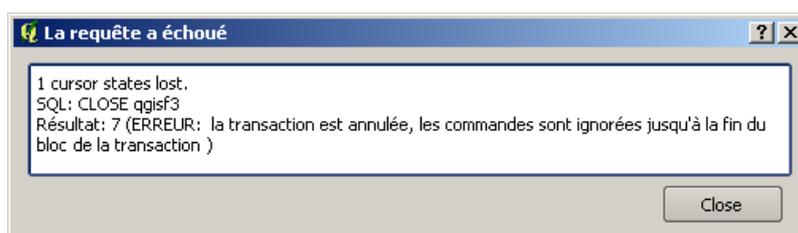
Dans la majorité des cas, ces géométries ne posent pas de problèmes et sont transparentes pour l'utilisateur.

En revanche, certains types d'invalidité de géométrie peuvent être **bloquants** pour des **calculs, l'affichage et l'export** des données. Dans les fichiers fonciers livrés par le CETE Nord Picardie, quelques parcelles de quelques départements présentent malheureusement ces propriétés. Cela vient d'erreurs déjà présentes dans la BD Parcellaire.

Le rapport de débogage du plugin contient alors les termes suivants :

**RecordFetch: ERREUR: geometry requires more points**

Ce problème peut également se manifester lors d'une tentative d'affichage de la géométrie invalide dans un logiciel SIG.



Erreur renvoyée par QGIS lors de la demande d'affichage d'une géométrie invalide bloquante

**Correction du problème en général** Le problème peut être corrigé en exécutant une requête SQL qui modifie la ou les géométrie(s) « défectueuses ». La requête générique est la suivante :

```
UPDATE nom_schema.nom_table SET champ_geometrique = Null
WHERE identifiant = 'valeur_de_l_identifiant_qui_pose_probleme'
```

**Correction du problème dans les fichiers fonciers** Les tables connues pour avoir un problème de géométrie invalide et bloquantes dans les fichiers fonciers sont indiquées dans le tableau ci-dessous.

Les départements concernés sont les 12, 14, 17, 19, 22, 30, 34, 40, 46, 49, 52, 54, 57, 61, 62, 68, 74, 76, 84, 94 et 95.

Afin de corriger ces géométries, un script SQL est disponible dans le même article que ce document.

Nom de table	Département	Année	Nombre de parcelles bloquantes
d12_2009_pnb10_parcelle	12	2009	1
d12_2011_pnb10_parcelle	12	2011	1
d14_2009_pnb10_parcelle	14	2009	2
d14_2011_pnb10_parcelle	14	2011	1
d17_2009_pnb10_parcelle	17	2009	1

Nom de table	Département	Année	Nombre de parcelles bloquantes
d17_2011_pnb10_parcelle	17	2011	1
d19_2009_pnb10_parcelle	19	2009	1
d19_2011_pnb10_parcelle	19	2011	1
d22_2009_pnb10_parcelle	22	2009	2
d22_2011_pnb10_parcelle	22	2011	2
d30_2009_pnb10_parcelle	30	2009	1
d30_2011_pnb10_parcelle	30	2011	1
d34_2009_pnb10_parcelle	34	2009	1
d34_2011_pnb10_parcelle	34	2011	1
d40_2009_pnb10_parcelle	40	2009	1
d40_2011_pnb10_parcelle	40	2011	1
d46_2009_pnb10_parcelle	46	2009	1
d46_2011_pnb10_parcelle	46	2011	1
d49_2011_pnb10_parcelle	49	2011	1
d52_2009_pnb10_parcelle	52	2009	4
d52_2011_pnb10_parcelle	52	2011	4
d54_2009_pnb10_parcelle	54	2009	1
d54_2011_pnb10_parcelle	54	2011	1
d57_2009_pnb10_parcelle	57	2009	1
d57_2011_pnb10_parcelle	57	2011	1
d61_2009_pnb10_parcelle	61	2009	1
d61_2011_pnb10_parcelle	61	2011	1
d62_2009_pnb10_parcelle	62	2009	1
d62_2011_pnb10_parcelle	62	2011	1
d68_2009_pnb10_parcelle	68	2009	1
d68_2011_pnb10_parcelle	68	2011	1
d74_2009_pnb10_parcelle	74	2009	4
d74_2011_pnb10_parcelle	74	2011	4
d76_2009_pnb10_parcelle	76	2009	1
d76_2011_pnb10_parcelle	76	2011	1
d84_2009_pnb10_parcelle	84	2009	1
d84_2011_pnb10_parcelle	84	2011	1
d94_2009_pnb10_parcelle	94	2009	1
d94_2011_pnb10_parcelle	94	2011	1
d95_2009_pnb10_parcelle	95	2009	1
d95_2011_pnb10_parcelle	95	2011	1

# Le plugin s'affiche, mais les données exportées ne sont pas valides (suite)

---

## Couper les tables trop volumineuse (cas particulier du département 75)

**Introduction** PostgreSQL permet de gérer des données sans quasiment aucune limite de taille. En revanche, les **formats exportés** peuvent avoir des **contraintes**.

C'est le cas en particulier pour les formats géographiques **ESRI Shapefile** et **Mapinfo**. Pour ces formats, les fichiers DBF et DAT (qui contiennent les données attributaires) doivent avoir une taille inférieure à 2 Go.

Dans les fichiers fonciers, seules deux tables sont trop volumineuses pour être exportées directement en Shape ou Tab :

- ff\_d75\_2009.d75\_2009\_pb0010\_local
- ff\_d75\_2011.d75\_2011\_pb0010\_local

En dehors des fichiers fonciers, pour vérifier si une table est trop volumineuse, il suffit de l'exporter et de regarder la taille du fichier dbf.

Si sa taille est supérieure ou égale à **2 147 483 647 octets**, elle n'est pas exportable directement.

**Pour résoudre le problème** Afin d'exporter les tables trop volumineuses, il faut les **séparer** selon une méthode ne provoquant aucune perte de données.

Pour le cas particulier de la table des locaux des fichiers fonciers sur le département 75, il est conseillé de couper la table selon les arrondissements :

- table 1 : les arrondissements 1 à 10,
- table 2 : les arrondissements 11 à 20.

Il suffit ensuite d'exporter ces deux tables avec le plugin.

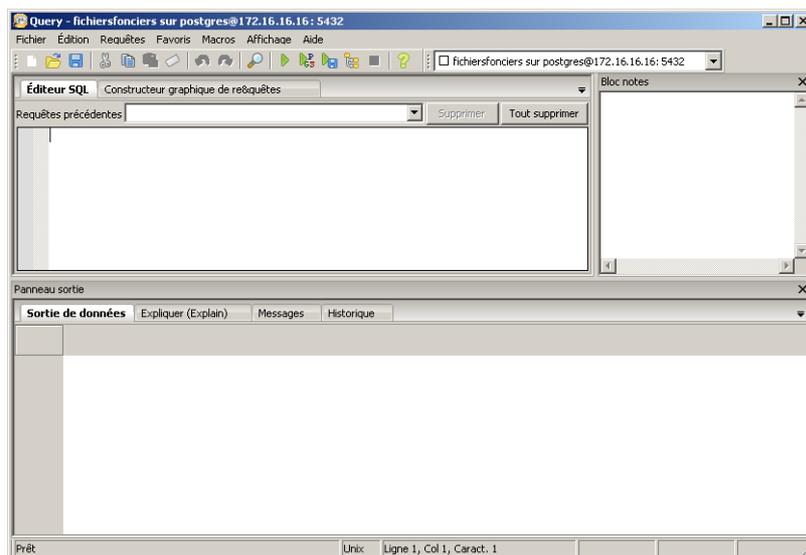
Les **requêtes** à exécuter pour créer ces deux tables sont écrites en **annexe 1 et 2**.

## Annexe 1 : comment exécuter une requête dans pgAdmin ?

### Étape 1 : ouvrir une fenêtre de requête dans pgAdmin

Pour ouvrir une fenêtre de requête dans pgAdmin, aller dans le menu "Outils" puis "Éditeur de requêtes".

Il est également possible d'ouvrir une fenêtre de requête en cliquant sur le bouton « SQL » :



Aperçu d'une fenêtre de requête vierge

### Étape 2 : écrire une requête SQL dans la fenêtre

Écrire une **requête SQL** standard dans la requête. Il s'agit de la fenêtre **supérieure**.

Ce document n'a pas pour objectif de décrire la syntaxe SQL. Celle-ci est largement détaillée sur cette page :

<http://docs.postgresql.fr/9.1/sql-commands.html>

Les **annexes 2 et 3** donnent des **requêtes prédéfinies** à copier/coller permettant de répondre à des problèmes d'export du plugin.

### Étape 3 : exécuter de la requête et visualiser le résultat

L'exécution d'une requête s'effectue soit :

- en cliquant sur le bouton « Play » vert ,
- en allant dans le menu « Requêtes » puis « Exécuter »

Le **résultat** de la requête d'affiche dans la fenêtre **inférieure**.

## Annexe 2 : requêtes pour l'export des tables trop volumineuses

### Rappel

Les tables PostgreSQL trop volumineuses pour être exportées aux formats TAB ou SHP doivent être séparées en plusieurs tables de taille indépendantes.

Dans les fichiers fonciers, seule la table des locaux 2009 et 2011 du département 75 présente cette caractéristique.

### Requêtes de séparation de table

La requête à exécuter est écrite ci-dessous. Voir l'annexe 1 pour l'exécution d'une requête SQL dans pgAdmin.

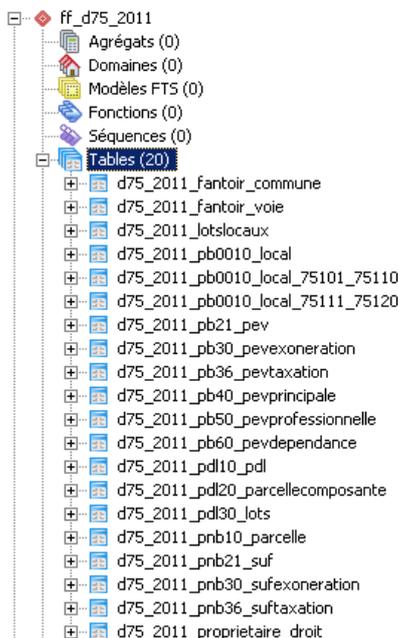
-- Creation de la premiere table

```
CREATE TABLE ff_d75_2011.d75_2011_pb0010_local_75101_75110 AS  
SELECT *  
FROM ff_d75_2011.d75_2011_pb0010_local  
WHERE idcom IN  
( '75101','75102','75103','75104','75105','75106','75107','75108','75109','75110');
```

-- Creation de la seconde table

```
CREATE TABLE ff_d75_2011.d75_2011_pb0010_local_75111_75120 AS  
SELECT *  
FROM ff_d75_2011.d75_2011_pb0010_local  
WHERE idcom IN  
( '75111','75112','75113','75114','75115','75116','75117','75118','75119','75120');
```

Après exécution de la requête, actualiser la liste des tables en cliquant gauche sur la liste des tables puis en appuyant sur « F5 ». Le résultat est indiqué ci-dessous.



Aperçu des 2 tables des locaux sur Paris par arrondissement

Enfin, utiliser le plugin sur les deux nouvelles tables pour les exporter.

*Remarque* : Pour effectuer la même manipulation sur la table 2009, remplacer simplement « 2011 » par « 2009 » dans la requête ci-dessus.