

## w.c.s. - Development #20410

### Passer convert-to-sql en commande de management

04 décembre 2017 13:54 - Frédéric Péters

<b>Statut:</b>	Fermé	<b>Début:</b>	04 décembre 2017
<b>Priorité:</b>	Normal	<b>Echéance:</b>	
<b>Assigné à:</b>	Christophe Siraut	<b>% réalisé:</b>	0%
<b>Catégorie:</b>		<b>Temps estimé:</b>	0:00 heure
<b>Version cible:</b>		<b>Planning:</b>	
<b>Patch proposed:</b>	Oui		
<b>Description</b>			
Prérequis pour pouvoir passer également hobo_deploy.			
Entre les paramètres de la ligne de commande et la configuration stockée, un nom diffère (dbname vs database; on avait pris dbname parce que c'est ce qui est utilisé pour la commande psql), c'est quelque chose qu'on pourrait changer à cette occasion.			
<b>Demandes liées:</b>			
Lié à w.c.s. - Bug #24716: convert_to_sql: créer la base de données si inexis...		<b>Fermé</b>	<b>22 juin 2018</b>
Lié à w.c.s. - Bug #24715: Ajouter une commande de management pour configurer...		<b>Fermé</b>	<b>22 juin 2018</b>

#### Révisions associées

##### Révision bb064934 - 16 juillet 2018 15:36 - Christophe Siraut

move convert\_to\_sql to management command (#20410)

#### Historique

##### #1 - 11 juin 2018 19:40 - Christophe Siraut

- Assigné à mis à Christophe Siraut

##### #2 - 15 juin 2018 16:41 - Christophe Siraut

- Fichier 0003-management-passer-convert-to-sql-en-commande-de-mana.patch ajouté

- Fichier 0002-tests-add-a-fonction-to-force-connections-close-when.patch ajouté

- Fichier 0001-management-convert\_to\_sql-add-failing-test.patch ajouté

- Patch proposed changé de Non à Oui

Voici une première version, je veux bien une relecture rapide pour confirmer que c'est la bonne direction.

Les questions que je me pose:

- est-ce ok d'avoir dépouillé wcs/ctl/convertsql.py et à terme le supprimer?

- est-ce ok de récupérer les paramètres de connexion depuis le skeleton? Faut-il aussi proposer les arguments manuels?

- je ne suis pas encore à l'aise avec la façon de travailler de quixote, notamment comment est-ce que sql.get\_connection\_and\_cursor() trouve les paramètres de connexion à postgresql?

##### #3 - 15 juin 2018 17:05 - Frédéric Péters

- Statut changé de Nouveau à En cours

- est-ce ok d'avoir dépouillé wcs/ctl/convertsql.py et à terme le supprimer?

Je ne vois pas ça dans les patches attachés. Mais 1) oui c'est utile de conserver la commande existante, dans un temps de transition, et oui elle peut devenir coquille presque vide, faisant appel à du code qui se trouverait désormais ailleurs.

- est-ce ok de récupérer les paramètres de connexion depuis le skeleton? Faut-il aussi proposer les arguments manuels?

Non, il faut passer par des paramètres sur la ligne de commande.

- je ne suis pas encore à l'aise avec la façon de travailler de quixote, notamment comment est-ce que sql.get\_connection\_and\_cursor() trouve

les paramètres de connexion à postgresql?

sql.get\_connection\_and\_cursor() vient de wcs, pas de quixote. sql.get\_connection() établit la connexion avec les paramètres présents dans la configuration, /var/lib/wcs/<tenant>/config.pck, paramètres chargés dans le publisher, obtenus via get\_cfg().

Dans convertsql.py aujourd'hui, le code :

```
pub.cfg['postgresql'] = {
    'database': sub_options.dbname,
    'user': sub_options.user,
    'password': sub_options.password,
    'host': sub_options.host,
    'port': sub_options.port,
}
```

```
sql.get_connection_and_cursor(new=True)
```

la config est remplie avec les paramètres passés à la fonction, une connexion est établie ensuite.

#### #4 - 21 juin 2018 12:29 - Christophe Siraut

- Fichier 0004-add-management-command-convert\_to\_sql-20410.patch ajouté

- Fichier 0003-add-management-command-configure\_connection.patch ajouté

Voici une proposition de patches et les tests associés, j'ai pris le parti de séparer une commande de configuration des paramètres de connexion postgresql (pouvant être utilisée dans le cas d'un changement de serveur comme actuellement SaaS v2) et la commande convert\_to\_sql qui s'occupe de migrer les objets.

#### #5 - 21 juin 2018 12:46 - Frédéric Péters

(lecture très rapide)

Tu peux modifier la configuration de ton éditeur pour autoriser des lignes plus longues, le bout suivant devrait tenir sur une ligne :

```
options_file = os.path.join(self.publisher.app_dir,
                             'site-options.cfg')
```

Il faut ajouter l'entête copyright/licence aux fichiers (sauf les tests).

Plutôt setup\_db que configure\_connection. Mais plutôt quand même continuer avec une seule commande, juste assurer qu'elle ne modifie pas les données si les tables existent déjà.

Utiliser add\_arguments plutôt que create\_parser.

Si tu veux factoriser un WcsTenantCommand, il faut adapter les autres commandes de gestion pour l'utiliser aussi. (commencer ça dans un commit isolé).

#### #6 - 21 juin 2018 13:57 - Christophe Siraut

Plutôt setup\_db que configure\_connection. Mais plutôt quand même continuer avec une seule commande, juste assurer qu'elle ne modifie pas les données si les tables existent déjà.

Pas sûr de comprendre: est-ce que tu proposes de garder une seule commande "convert\_to\_sql" qui soit abusable dans le scénario de reconfiguration d'un tenant?

Utiliser add\_arguments plutôt que create\_parser.

Dans le cas de WcsTenantCommand j'avais opté pour surcharger create\_parser pour que les classes dérivées n'aient pas à invoquer super().add\_arguments().

Quelle est la longueur de ligne maximale? (j'étais à 80 caractères)

#### #7 - 21 juin 2018 14:06 - Frédéric Péters

Pas sûr de comprendre: est-ce que tu proposes de garder une seule commande "convert\_to\_sql" qui soit abusable dans le scénario de reconfiguration d'un tenant?

Que convert\_to\_sql écrive les paramètres de configuration, sans écriture de données, quand la base de données "cible" contient déjà les tables, oui. (après, que ce ticket ne touche à rien sur le fonctionnement, et qu'on discute d'évolutions dans un autre, ça me va aussi).

Quelle est la longueur de ligne maximale? (j'étais à 80 caractères)

Je suis surpris de ne pas trouver de référence à ça dans le wiki; comme <http://docs.python-requests.org/en/latest/dev/contributing/#kenneth-reitz-s-code-style> (Line-length can exceed 79 characters, to 100, when convenient. / Line-length can exceed 100 characters, when doing otherwise would be terribly inconvenient.)

#### #8 - 21 juin 2018 17:14 - Christophe Siraut

Voici une version adaptée suite à tes remarques.

#### #9 - 21 juin 2018 17:15 - Christophe Siraut

- Fichier 0002-add-management-command-convert\_to\_sql-20410.patch ajouté

- Fichier 0001-tests-utilities-add-force\_connections\_close.patch ajouté

#### #10 - 21 juin 2018 17:28 - Frédéric Péters

```
cur.execute(''SELECT pg_terminate_backend(pg_stat_activity.pid)
FROM pg_stat_activity
WHERE pg_stat_activity.datname = '{}'
AND pid <> pg_backend_pid();'''.format(known_elements.sql_db_name))
cur.close()
```

Il faut indenter différemment l'SQL, là son indentation le confond avec le code Python. (tu peux regarder dans wcs/sql.py).

On n'utilise pas .format() (je n'avais pas fait gaffe dans le précédent où je l'aurais déjà dit).

1. Copyright (C) 2005-2017 Entr'ouvert

2018.

```
import os
...
```

On sépare les imports en différentes zones avec une ligne blanche entre (grosso modo modules systèmes, modules django, quixote, puis wcs).

Dans la version actuelle on ne touche pas au site-options.cfg, je n'aurais pas touché à ça mais ok, par contre utilise plutôt atomic\_write pour remplacer le fichier (de qommon.storage).

Je pensais que tu partagerais le code entre l'ancien et le nouveau convert-to-sql, que l'ancien ne devienne qu'un wrapper vers le nouveau.

La barre de progression, ce serait bien de la conserver. (quand on utilise la commande sur un site existant depuis un certain temps, c'est vraiment plus confortable).

Je note pour plus tard, il y aurait en évolution à voir dans quelle mesure la méthode initialize\_sql() de la classe Publisher pourrait être exploitée.

#### #11 - 21 juin 2018 20:24 - Christophe Siraut

- Fichier 0001-tests-utilities-add-force\_connections\_close.patch ajouté

- Fichier 0002-move-convert\_to\_sql-to-management-command-20410.patch ajouté

Nouvelle version, en tenant compte des commentaires.

#### #12 - 21 juin 2018 23:19 - Frédéric Péters

Restent les parties qui utilisent format, comme ('{} {}'.format(formdata.fromdef, formdata.id)); le reste d'un œil rapide me semble ok. (je n'ai pas exécuté pour de vrai et jenkins annonce une erreur qui est également à voir, <https://jenkins.entrouvert.org/job/wcs-wip-branches/178/>).

#### #13 - 22 juin 2018 18:27 - Christophe Siraut

Je ne comprends pas bien l'erreur de jenkins, et ne parviens pas à la reproduire en local (pytest/jessie). L'erreur ne survient pas si on garde intact wcs/ctl/convertsql.py.

#### #14 - 27 juin 2018 08:34 - Christophe Siraut

- Fichier 0002-move-convert\_to\_sql-to-management-command-20410.patch ajouté

- Fichier 0001-tests-utilities-add-force\_connections\_close.patch ajouté

J'ai mis à jour la branche wip afin de passer les tests, patches attachés.

La différence avec la version précédente c'est que j'ai laissé tomber la modification du code original pour pointer vers le nouveau.

#### #15 - 27 juin 2018 09:17 - Frédéric Péters

La différence avec la version précédente c'est que j'ai laissé tomber la modification du code original pour pointer vers le nouveau.

Qu'il soit donc supprimé après un tour un peu général voir où il pourrait être utilisé (peut-être même nulle part de manière automatique, je ne sais pas); on ne peut pas avoir du code dupliqué ainsi.

#### #16 - 27 juin 2018 09:25 - Christophe Siraut

- Fichier FAIL\_0003-link-old-converts sql-command-to-current-implementation.patch ajouté

(Juste pour l'historique, j'attache le patch que refuse jenkins, et les erreurs associées)

```
===== FAILURES =====
_____ test_cron_command _____

def test_cron_command():
    pub = create_temporary_pub()
    with mock.patch('wcs.qommon.management.commands.cron.cron_worker') as cron_worker:
>         call_command('cron')

tests/test_publisher.py:175:
-----
../../../../shiningpanda/jobs/22cf219b/virtualenvs/d41d8cd9/local/lib/python2.7/site-packages/django/core/management/
__init__.py:120: in call_command
    return command.execute(*args, **defaults)
../../../../shiningpanda/jobs/22cf219b/virtualenvs/d41d8cd9/local/lib/python2.7/site-packages/django/core/management/
base.py:441: in execute
    output = self.handle(*args, **options)
wcs/qommon/management/commands/cron.py:43: in handle
    publisher_class.register_cronjobs()
-----

cls = <class 'publisher.WcsPublisher'>

    @classmethod
    def register_cronjobs(cls):
>     super(WcsPublisher, cls).register_cronjobs()
E     TypeError: super() argument 1 must be type, not None

wcs/publisher.py:115: TypeError
_____ test_migration_12_users_fts _____

@postgresql
def test_migration_12_users_fts():
    conn, cur = sql.get_connection_and_cursor()
    cur.execute('UPDATE wcs_meta SET value = 11 WHERE key = %s', ('sql_level',))

    sql.SqlUser.wipe()

    user = sql.SqlUser()
    user.name = 'Pierre'
    user.store()

    # remove the fts column
    cur.execute('ALTER TABLE users DROP COLUMN fts')
    assert not column_exists_in_table(cur, 'users', 'fts')
    sql.migrate()

    assert column_exists_in_table(cur, 'users', 'fts')
    assert migration_level(cur) >= 12
```

```

# no fts, migration only prepare re-index
assert len(sql.SqlUser.get_ids_from_query('pierre')) == 0

assert sql.is_reindex_needed('user', conn=conn, cur=cur) is True
assert sql.is_reindex_needed('formdata', conn=conn, cur=cur) is True
> call_command('cron') # first cron = reindex

tests/test_sql.py:1112:
-----
../../../../shiningpanda/jobs/22cf219b/virtualenvs/d41d8cd9/local/lib/python2.7/site-packages/django/core/management/
ent/__init__.py:120: in call_command
    return command.execute(*args, **defaults)
../../../../shiningpanda/jobs/22cf219b/virtualenvs/d41d8cd9/local/lib/python2.7/site-packages/django/core/management
ent/base.py:441: in execute
    output = self.handle(*args, **options)
wcs/gommon/management/commands/cron.py:43: in handle
    publisher_class.register_cronjobs()
-----

cls = <class 'publisher.WcsPublisher'>

@classmethod
def register_cronjobs(cls):
> super(WcsPublisher, cls).register_cronjobs()
E     TypeError: super() argument 1 must be type, not None

wcs/publisher.py:115: TypeError
_____ test_migration_21_users_ascii_name _____

@postgresql
def test_migration_21_users_ascii_name():
    conn, cur = sql.get_connection_and_cursor()
    cur.execute('UPDATE wcs_meta SET value = 11 WHERE key = %s', ('sql_level',))

    sql.SqlUser.wipe()

    user = sql.SqlUser()
    user.name = 'Jean Sénisme'
    user.store()

    # remove the ascii_name column
    cur.execute('ALTER TABLE users DROP COLUMN ascii_name')
    assert not column_exists_in_table(cur, 'users', 'ascii_name')
    sql.migrate()

    assert column_exists_in_table(cur, 'users', 'ascii_name')
    assert migration_level(cur) >= 21

    # no fts, migration only prepare re-index
    assert sql.SqlUser.count([st.Equal('ascii_name', 'jean senisme')]) == 0

    assert sql.is_reindex_needed('user', conn=conn, cur=cur) is True
    assert sql.is_reindex_needed('formdata', conn=conn, cur=cur) is True
> call_command('cron') # first cron = reindex

tests/test_sql.py:1146:
-----
../../../../shiningpanda/jobs/22cf219b/virtualenvs/d41d8cd9/local/lib/python2.7/site-packages/django/core/management
ent/__init__.py:120: in call_command
    return command.execute(*args, **defaults)
../../../../shiningpanda/jobs/22cf219b/virtualenvs/d41d8cd9/local/lib/python2.7/site-packages/django/core/management
ent/base.py:441: in execute
    output = self.handle(*args, **options)
wcs/gommon/management/commands/cron.py:43: in handle
    publisher_class.register_cronjobs()
-----

cls = <class 'publisher.WcsPublisher'>

@classmethod
def register_cronjobs(cls):
> super(WcsPublisher, cls).register_cronjobs()
E     TypeError: super() argument 1 must be type, not None

wcs/publisher.py:115: TypeError

```

generated xml file: /var/lib/jenkins/jobs/wcs-wip-branches/workspace/test\_results.xml

----- coverage: platform linux2, python 2.7.13-final-0 -----  
Coverage XML written to file coverage.xml

===== 3 failed, 1398 passed, 21 skipped in 837.86 seconds =====  
Build step 'Virtualenv Builder' marked build as failure  
Enregistrement des résultats des tests  
Sending e-mails to: admin+jenkins-wcs@entrouvert.com csiraut@entrouvert.com  
[Cobertura] Publishing Cobertura coverage report...

[Cobertura] Publishing Cobertura coverage results...

[Cobertura] Cobertura coverage report found.

Finished: FAILURE

#### #17 - 27 juin 2018 09:28 - Christophe Siraut

(désolé pour l'objet Struct(), c'est une scorie)

#### #18 - 27 juin 2018 10:52 - Christophe Siraut

- Fichier 0003-remove-previous-convert-to-sql-command-20410.patch ajouté
- Fichier 0002-move-convert\_to\_sql-to-management-command-20410.patch ajouté
- Fichier 0001-tests-utilities-add-force\_connections\_close.patch ajouté

Voici le patch pour la suppression de l'ancienne implémentation.

#### #19 - 02 juillet 2018 10:17 - Christophe Siraut

- Lié à Bug #24716: convert\_to\_sql: créer la base de données si inexistante ajouté

#### #20 - 02 juillet 2018 10:17 - Christophe Siraut

- Lié à Bug #24715: Ajouter une commande de management pour configurer la base de données ajouté

#### #21 - 02 juillet 2018 10:19 - Frédéric Péters

Qu'il soit donc supprimé après un tour un peu général voir où il pourrait être utilisé. (...)

C'est au moins utilisé chez imio (mention dans leur module docker-teleservices), comme on n'assure pas de compatibilité, j'attendrais de pouvoir en discuter avec eux.

#### #22 - 09 juillet 2018 08:54 - Frédéric Péters

- Statut changé de En cours à Solution validée

Voilà, je pense que c'est ok de pousser ça, en réunissant 0002 et 0003 en un unique commit.

#### #23 - 17 juillet 2018 09:35 - Frédéric Péters

- Statut changé de Solution validée à Résolu (à déployer)

```
commit bb064934b3261b5c4dfdce7c93247193e5198070
Author: Christophe Siraut <csiraut@entrouvert.com>
Date: Tue Jun 26 17:13:15 2018 +0200
```

```
move convert_to_sql to management command (#20410)
```

```
commit 888622ea185b734dd684dd434454cf110dbe8bbf
Author: Christophe Siraut <csiraut@entrouvert.com>
Date: Thu Jun 21 17:07:00 2018 +0200
```

```
tests/utilities: add force_connections_close()
```

#### #24 - 23 décembre 2018 14:44 - Frédéric Péters

- Statut changé de Résolu (à déployer) à Solution déployée

## Fichiers

---

0003-management-passer-convert-to-sql-en-commande-de-mana.patch	18,47 ko	15 juin 2018	Christophe Siraut
0002-tests-add-a-function-to-force-connections-close-when.patch	1,43 ko	15 juin 2018	Christophe Siraut
0001-management-convert_to_sql-add-failing-test.patch	740 octets	15 juin 2018	Christophe Siraut
0004-add-management-command-convert_to_sql-20410.patch	6,23 ko	21 juin 2018	Christophe Siraut
0003-add-management-command-configure_connection.patch	6,11 ko	21 juin 2018	Christophe Siraut
0002-add-management-command-convert_to_sql-20410.patch	10,4 ko	21 juin 2018	Christophe Siraut
0001-tests-utilities-add-force_connections_close.patch	1,4 ko	21 juin 2018	Christophe Siraut
0001-tests-utilities-add-force_connections_close.patch	1,41 ko	21 juin 2018	Christophe Siraut
0002-move-convert_to_sql-to-management-command-20410.patch	16,2 ko	21 juin 2018	Christophe Siraut
0001-tests-utilities-add-force_connections_close.patch	1,5 ko	27 juin 2018	Christophe Siraut
0002-move-convert_to_sql-to-management-command-20410.patch	10,7 ko	27 juin 2018	Christophe Siraut
FAIL_0003-link-old-converts sql-command-to-current-implementation.patch	6,52 ko	27 juin 2018	Christophe Siraut
0003-remove-previous-convert-to-sql-command-20410.patch	6,18 ko	27 juin 2018	Christophe Siraut
0002-move-convert_to_sql-to-management-command-20410.patch	10,5 ko	27 juin 2018	Christophe Siraut
0001-tests-utilities-add-force_connections_close.patch	1,5 ko	27 juin 2018	Christophe Siraut