

## w.c.s. - Bug #15379

### crash suite à écrasement d'un formulaire, crash lié au format SQL

10 mars 2017 15:13 - Thomas Noël

<b>Statut:</b>	Fermé	<b>Début:</b>	10 mars 2017
<b>Priorité:</b>	Normal	<b>Echéance:</b>	
<b>Assigné à:</b>		<b>% réalisé:</b>	0%
<b>Catégorie:</b>		<b>Temps estimé:</b>	0:00 heure
<b>Version cible:</b>		<b>Planning:</b>	Non
<b>Patch proposed:</b>	Non		
<b>Description</b>			
<pre>LINE 1: ...ULL, submission_context = NULL, f23 = NULL, f22 = '2010-01-2...                                          ^ HINT: You will need to rewrite or cast the expression. ) To: admin+cch-omonville@entrouvert.com  Exception:   type = '&lt;class 'psycpg2.ProgrammingError'&gt;', value = 'column "f22" is of type text[] but expres sion is of type timestamp without time zone LINE 1: ...ULL, submission_context = NULL, f23 = NULL, f22 = '2010-01-2...                                          ^ HINT: You will need to rewrite or cast the expression. '</pre>			
Stack trace (most recent call first): File "/usr/lib/python2.7/dist-packages/wcs/sql.py", line 1194, in store 1192         self._table_name, 1193         ', '.join(['%s = %%(%)s' % (x,x) for x in column_na mes])) > 1194         cur.execute(sql_statement, sql_dict) 1195         if cur.fetchone() is None: 1196             column_names = sql_dict.keys()			
Le champ 22 du formulaire est de type date, mais dans le SQL on a un text[]			
Le formulaire a été "overwrité" :			
<pre>78.247.170.27 -- [08/Mar/2017:09:04:23 +0100] "POST /backoffice/forms/2/overwrite HTTP/1.1" 302 1 09 "https://omonville-la-rogue.test.entrouvert.org/backoffice/forms/2/overwrite" "Mozilla/5.0 (X11 ; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0"</pre>			
Je pose ce ticket sans trop savoir ce qu'il faut faire ; mais je dirais que lors d'un overwrite on devrait peut-être vérifier que le type des champs de même id correspond (et refuser l'overwrite) ?			
<b>Demandes liées:</b>			
Lié à w.c.s. - Development #36711: "écraser avec un nouvel import", "sans dan...		<b>Fermé</b>	<b>06 octobre 2019</b>

#### Révisions associées

##### Révision 2eea3725 - 25 octobre 2019 10:46 - Frédéric Péters

admin: revamp overriding forms (#36711)

- always display summary of changes if there are data
- -> skip summary of changes if there are no data (#13599)
- do not use scrolling in summary (#32177)
- ignore no-data fields when comparing types
- warn about data loss instead of data corruption/bugs
- remove data from incompatible columns (in SQL) (#15379)

## Historique

---

### #1 - 06 décembre 2018 17:51 - Thomas Noël

Le cas se produit quand même régulièrement, les gens cochent sur "Overwrite nevertheless" et hop...

```
get_request().method = 'GET'
get_request().form = {}
form = Form(enctype='multipart/form-data', use_tokens=False)
if different_type_fields:
    form.widgets.append(HtmlWidget('<div class="errornotice"><p>%s</p></div>' % _(
        'The form has incompatible fields, it may cause data corruption and bugs.')))
    form.add(CheckboxWidget, 'force', title=_('Overwrite nevertheless'))
else:
    form.add_hidden('force', 'ok')
form.add_hidden('new_formdef', ET.tostring(new_formdef.export_to_xml(include_id=True)))
form.add_submit('submit', _('Submit'))
form.add_submit('cancel', _('Cancel'))
r += form.render()
```

Il faut être plus violent, et ne plus proposer l'overwrite en cas de different\_type\_fields.

### #3 - 06 décembre 2018 18:04 - Frédéric Péters

régulièrement

Par rapport à combien de fois où forcer les choses "marche" ? (grep POST.\*overwrite sur les access.log, ça pourrait nous donner une indication ?)

### #4 - 06 décembre 2018 18:54 - Thomas Noël

Frédéric Péters a écrit :

régulièrement

Par rapport à combien de fois où forcer les choses "marche" ? (grep POST.\*overwrite sur les access.log, ça pourrait nous donner une indication ?)

Ok, j'ai exagéré, vous me connaissez. Régulièrement = 2 fois cette semaine, dont une fois en prod avec du jeu de backup pas funky, et ça me suffit pour dire qu'il faut interdire cette manip et c'est tout.

### #5 - 07 décembre 2018 10:26 - Benjamin Dauvergne

Et générer des id de field uniques plutôt ?

```
In [5]: 'f' + base64.urlsafe_b64encode(os.urandom(8)).strip('=')
Out[5]: 'f5FR91ww4FFs'
```

Aucune chance qu'en cas de fork d'un formulaire un ré-import crée une collision.

### #6 - 07 décembre 2018 10:28 - Benjamin Dauvergne

Plutôt de l'encodage base32 pour éviter les problèmes de casse (je ne sais pas si on quote ou pas les noms de colonne dans la partie SQL, mais le mieux c'est d'éviter le problème).

### #7 - 07 décembre 2018 11:12 - Thomas Noël

En fait on pourrait juste prendre des nouveaux fNNN pour ces champs en conflit.

Mais en fait, ces conflits de champs, ils traduisent une mauvaise procédure de travail (normalement on bosse sur une copie du formulaire en prod, et il est impossible de changer le type d'un champ, donc ça peut pas arriver). C'est pour ça que je pense vraiment juste interdire l'overwrite, et que la personne doit reprendre son travail correctement.

### #8 - 07 décembre 2018 11:38 - Benjamin Dauvergne

Thomas Noël a écrit :

En fait on pourrait juste prendre des nouveaux fNNN pour ces champs en conflit.

Mais en fait, ces conflits de champs, ils traduisent une mauvaise procédure de travail (normalement on bosse sur une copie du formulaire en prod, et il est impossible de changer le type d'un champ, donc ça peut pas arriver). C'est pour ça que je pense vraiment juste interdire

l'overwrite, et que la personne doit reprendre son travail correctement.

Sauf que c'est inexplicable aux gens (vous avez forké, et donc on a généré des id identiques pour des champs différents découlant d'un travail sur deux plate-formes différentes). Si on interdit de forcer l'overwrite sans rien faire de plus, on aura des tickets de support pour "j'arrive pas à mettre à jour" en pagaille.

Si on force l'overwrite et qu'on génère des id globalement uniques, on évite les problèmes et on permet aux gens de travailler (et de faire vraisemblablement de la merde quand ils verront qu'un champ x créé des deux cotés ne reprend pas les données du champ en prod, mais ça c'est le boulot de l'écran de dire les données du champs X vont être perdu, are you ok ? vous devriez partir d'un export d'un formulaire avant de le modifier pour éviter ces problèmes).

#### #9 - 07 décembre 2018 14:15 - Thomas Noël

mais ça c'est le boulot de l'écran de dire les données du champs X vont être perdu, are you ok ?

C'est déjà le cas on affiche déjà des warning, mais "les gens" (dont nous) cliquent sur "oui oui vazy je m'en fous", et boum. Moi j'ai vraiment plus envie, jamais, d'avoir à faire le pompier en prod sur ce genre de soucis, aussi petit soit-il (et parce qu'en plus, il ne peut être visible que x jours après).

Si on met pas le bouton, alors les gens sont bien obligés de lire ce qui est écrit, et on peut éventuellement rappeler la procédure (repartir du formulaire actuel).

En tout cas moi, je préfère ça, et si on le fait pas la prochaine fois c'est toi qui remonte les backups :)

#### #10 - 19 décembre 2018 09:51 - Benjamin Dauvergne

Encore un exemple, [#15379](#), je crois qu'il faudrait déjà informer les CPFs des écueils avec la situation actuelle, parce que quand je signale à Mike qu'il faut respecter un certain workflow entre test et prod il n'était visiblement pas au courant.

Vraiment je ne vois pas ce que nous apporter les identifiants séquentiels (en fait on pourrait même avoir les deux si on veut conserver une sorte de timestamp virtuel sur l'ordre de création des champs, genre `{counter}_{hash}`).

#### #11 - 19 décembre 2018 10:10 - Mikaël Ates

Je ne travail que sur la recette (pour mon cas présent) et pas souvenir qu'on est parlé de wf prod/recette (pour mon cas présent).

J'ai un formulaire avec des zones de formulaires qui se répètent, des contacts par exemple. Ma pratique c'est d'exporter le formulaire, dans un éditeur de texte de copier-coller mon ensemble de champs. De faire un recherche remplacer sur mes variables `_contact1` pour les remplacer par `_contact2`. Je lance ensuite un petit script sur mon fichier pour renuméroter tous les champs :

```
import sys
import xml.etree.ElementTree as ET

fn = sys.argv[1]

print "Renumbering..."
tree = ET.parse(fn)
root = tree.getroot()
i = 1
for elem in root.iter('id'):
    elem.text = str(i)
    i += 1
tree.write(fn)
print "Done"
```

J'écrase ensuite mon formulaire avec un nouvel import.

#### #12 - 19 décembre 2018 10:12 - Benjamin Dauvergne

Un truc en passant aussi, dans ce code de `wcs/admin/forms.py:FormDefPage` :

```
def overwrite_by_formdef(self, new_formdef):
    # keep current formdef id, url_name, internal identifier and sql table name
    new_formdef.id = self.formdef.id
    new_formdef.internal_identifier = self.formdef.internal_identifier
    new_formdef.url_name = self.formdef.url_name
    new_formdef.table_name = self.formdef.table_name
    # keep currently assigned category and workflow
    new_formdef.category_id = self.formdef.category_id
    new_formdef.workflow_id = self.formdef.workflow_id
    # keep currently assigned roles
    new_formdef.workflow_roles = self.formdef.workflow_roles
```

```
new_formdef.backoffice_submission_roles = self.formdef.backoffice_submission_roles
new_formdef.roles = self.formdef.roles
self.formdef = new_formdef
self.formdef.store()
```

Je me demande si on ne devrait pas aussi reprendre `max_field_id` de cette manière:

```
new_formdef.max_field_id = max(self.formdef.max_field_id, new_formdef.max_field_id)
```

histoire de toujours rester "en avance".

### **#13 - 19 décembre 2018 10:13 - Frédéric Péters**

J'écrase ensuite mon formulaire avec un nouvel import.

Et tu n'as pas d'avertissement (attention champs incompatibles, ne faites pas ça.) ?

### **#15 - 19 décembre 2018 10:28 - Mikaël Ates**

(Dans cette phase de développement en recette je n'ai pas besoin des formdata. Je vais donc supprimer l'ancien formulaire et en créer un par import plutôt que de faire écraser avec un nouvel import sur le formulaire existant.)

### **#16 - 06 octobre 2019 19:26 - Frédéric Péters**

- Lié à *Development #36711*: "écraser avec un nouvel import", "sans danger" ajouté

### **#17 - 25 octobre 2019 10:47 - Frédéric Péters**

- Statut changé de *Nouveau* à *Fermé*

Traité via [#36711](#).