

## w.c.s. - Development #62293

### Ne pas considérer une variable comme étant live si le champ associé est caché

01 mars 2022 11:59 - Benjamin Dauvergne

<b>Statut:</b>	Fermé	<b>Début:</b>	01 mars 2022
<b>Priorité:</b>	Normal	<b>Echéance:</b>	
<b>Assigné à:</b>	Benjamin Dauvergne	<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>	Oui		

**Description**

Dans `field.has_live_condition()` on traite tous les varnames comme étant "vivant" mais si le champ associé est définitivement caché ce ne peut pas être le cas.

J'ai testé le code suivant sur <https://demarches-montoulouse.cutm-publik-preprod.nfrance.com/backoffice/submission/allo-toulouse-sigalements/>

On passe de :

```
1 0.000 0.000 4.820 4.820 /usr/lib/python3/dist-packages/wcs/qommon/form.py:38
8(_render_widgets)
```

à

```
1 0.000 0.000 0.374 0.374 /usr/lib/python3/dist-packages/wcs/qommon/form.py:38
8(_render_widgets)
```

parce que la plupart des champs "bloc de champs" se retrouvent caché définitivement.

```
diff --git a/wcs/fields.py b/wcs/fields.py
index 2e9f8fbbe..2405189c1 100644
--- a/wcs/fields.py
+++ b/wcs/fields.py
@@ -617,7 +617,7 @@ class Field:
     def get_condition_varnames(self, formdef):
         return self.get_referenced_varnames(formdef, self.condition['value'])

-     def has_live_conditions(self, formdef):
+     def has_live_conditions(self, formdef, visibility=None):
         varnames = self.get_condition_varnames(formdef)
         if not varnames:
             return False
@@ -633,7 +633,7 @@ class Field:
         if field.type == 'page':
             # stop at next page
             break
-         if field.varname in varnames:
+         if field.varname in varnames and (visibility is None or field.varname in visibility):
             return True
         return False

diff --git a/wcs/formdef.py b/wcs/formdef.py
index e18a14d68..d14388969 100644
--- a/wcs/formdef.py
+++ b/wcs/formdef.py
@@ -826,6 +826,7 @@ class FormDef(StorableObject):
     transient_formdata=None,
 ): # a FormData
     on_page = page is None
+    visibility = set()
```

```

        for field in self.fields:
            field.formdef = self
            if field.type == 'page':
@@ -838,9 +839,11 @@ class FormDef(StorableObject):
                continue
            visible = field.is_visible(form_data, self)
            if not visible:
-                if not field.has_live_conditions(self):
+                if not field.has_live_conditions(self, visibility=visibility):
                    # no live conditions so field can be skipped
                    continue
+            if field.varname:
+                visibility.add(field.varname)
            if isinstance(displayed_fields, list):
                displayed_fields.append(field)
            value = None

```

À discuter parce qu'un des effets de bord c'est qu'on ne pourra plus avoir de condition "live" dépendant d'un champ plus loin dans le formulaire, ce qui me semble tout à fait acceptable.

## Révisions associées

### Révision 75d6cdcc - 04 mars 2022 10:34 - Benjamin Dauvergne

fields: ignore hidden fields in has\_live\_conditions (#62293)

## Historique

### #2 - 01 mars 2022 12:11 - Frédéric Péters

À discuter parce qu'un des effets de bord c'est qu'on ne pourra plus avoir de condition "live" dépendant d'un champ plus loin dans le formulaire, ce qui me semble tout à fait acceptable.

Quasi sûr que c'est des choses qui se pratiquent, dans des champs commentaire, il faut vérifier tout ça avant de pouvoir en discuter; éventuellement ça peut être un patch qui loggue la situation, qu'on ferait tourner pendant quelques cycles avant d'aller voir les résultats.

### #3 - 01 mars 2022 12:50 - Frédéric Péters

Quasi sûr que c'est des choses qui se pratiquent, dans des champs commentaire

Quoique, en fait, il me semble qu'on pose déjà comme limite que les conditions ne peuvent pas être sur des champs qui viennent plus loin, il me semble avoir vu un ticket plus ou moins récent sur le sujet et y avoir écrit ça (mais je ne retrouve pas).

### #4 - 01 mars 2022 12:59 - Benjamin Dauvergne

On ne remonte pas avant le dernier champ page, par contre ensuite on déroule formdef.fields sans s'arrêter :

```

# rewind to field page
for field_position in range(field_position, -1, -1):
    if formdef.fields[field_position].type == 'page':
        break
else:
    field_position = -1 # form with no page
# start from there
for field in formdef.fields[field_position + 1 :]:

```

### #5 - 01 mars 2022 17:35 - Benjamin Dauvergne

- Assigné à mis à Benjamin Dauvergne

### #6 - 01 mars 2022 17:35 - Benjamin Dauvergne

- Fichier 0001-fields-ignore-hidden-fields-in-has\_live\_conditions-6.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

J'ai inversé la condition, si le champ est après l'optimisation ne marchera juste pas. Ça me parait encore plus acceptable.

#### #7 - 04 mars 2022 10:35 - Frédéric Péters

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

Je balance ça, si jamais ça fait planter des trucs semaine prochaine qu'on hésite pas à l'annuler, et à figer la version sur toulouse en attendant.

```
commit 75d6cdcc63b77b374c3ae1a408fbc21b038ec894
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>
Date: Tue Mar 1 11:36:07 2022 +0100
```

```
fields: ignore hidden fields in has_live_conditions (#62293)
```

#### #8 - 05 mars 2022 10:17 - Transition automatique

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

#### #9 - 08 mai 2022 04:42 - Transition automatique

Automatic expiration

### Fichiers

---

0001-fields-ignore-hidden-fields-in-has_live_conditions-6.patch	3,54 ko	01 mars 2022	Benjamin Dauvergne
---	---------	--------------	--------------------