Passerelle - Development #41612

csvdatasource: ne pas faire .splitlines() sur le contenu du CSV

10 avril 2020 17:12 - Benjamin Dauvergne

Statut:	Rejeté	**Début:**	10 avril 2020
Priorité:	Normal	**Echéance:**	
Assigné à:	Benjamin Dauvergne	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	Non

Description

Les fichiers CSV contenant des sauts de ligne les perdent (tester par exemple :

```
id,text,comment
1,a,"un
long
texte"
2,a,b
```

parce que le fichier est parsé de cette manière :

```
            reader = csv.reader(content.splitlines(), **self.dialect_options)
```

et str.splitlines supprime les sauts de ligne de chaque ligne (contrairement à open(..).readlines()).

Historique

#1 - 10 avril 2020 17:12 - Benjamin Dauvergne

- Tracker changé de Support à Bug

#3 - 10 avril 2020 17:27 - Benjamin Dauvergne

- Assigné à mis à Benjamin Dauvergne

#4 - 10 avril 2020 17:27 - Benjamin Dauvergne

- Fichier 0001-csvdatasource-keep-inline-newlines-when-parsing-CSV-.patch ajouté

- Tracker changé de Bug à Development

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

#5 - 10 avril 2020 18:33 - Benjamin Dauvergne

- Statut changé de Solution proposée à En cours

Et donc personne n'avait jamais vu le problème parce que personne n'a réussi à faire avaler un fichier CSV avec des sauts de ligne, le sniffer en 2.7.13 n'y arrive pas (sur ma machine en 2.7.16 ça marche très bien).

```
https://github.com/python/cpython/blob/2.7/Misc/NEWS.d/2.7.15rc1.rst

Fixed guessing quote and delimiter in csv.Sniffer.sniff() when only the last field is quoted. Patch by Jake Da
vis.
```

#6 - 10 avril 2020 18:45 - Benjamin Dauvergne

- Statut changé de En cours à Solution proposée

Bon ben voilà, en rajoutant une colonne ça passe.

#7 - 11 avril 2020 12:38 - Benjamin Dauvergne

Le changement est assez minime entre 2.7.13 et 2.7.16 :

```
$ diff -ub /tmp/csv.py /usr/lib/python2.7/csv.py
--- /tmp/csv.py    2020-04-11 12:37:50.081188000 +0200
+++ /usr/lib/python2.7/csv.py    2019-09-04 10:19:57.000000000 +0200
@@ -217,7 +217,7 @@
         matches = []
         for restr in ('(?P<delim>[^\w\n"\'])(?P<space> ?)(?P<quote>["\']).*?(?P=quote)(?P=delim)', # ,".*?",
                       '(?:^|\n)(?P<quote>["\']).*?(?P=quote)(?P<delim>[^\w\n"\'])(?P<space> ?)',   #  ".*?",
-                      '(?P<delim>>[^\w\n"\'])(?P<space> ?)(?P<quote>["\']).*?(?P=quote)(?:$|\n)',  # ,".*?"
+                      '(?P<delim>[^\w\n"\'])(?P<space> ?)(?P<quote>["\']).*?(?P=quote)(?:$|\n)',   # ,".*?"
                       '(?:^|\n)(?P<quote>["\']).*?(?P=quote)(?:$|\n)'):                            #  ".*?" (
no delim, no space)
             regexp = re.compile(restr, re.DOTALL | re.MULTILINE)
             matches = regexp.findall(data)
```

J'ai bien envie de la backporter manuellement.

#8 - 11 avril 2020 13:43 - Benjamin Dauvergne

- Fichier 0001-csvdatasource-keep-inline-newlines-when-parsing-CSV-.patch ajouté

- Fichier 0002-csvdatasource-backports-fix-on-csv.Sniffer-41612.patch ajouté

Avec le backport.

#9 - 08 octobre 2020 18:55 - Benjamin Dauvergne

- Statut changé de Solution proposée à Rejeté

De toute façon ça ne marchera jamais bien, il faut un formulaire explicite de définition des paramètres du CSV, il n'existe pas d'autre moyen (Sniffer est capable de prendre les espaces pour des séparateurs sur un CSV suffisamment compliqué).

Fichiers

test.csv	42 octets	10 avril 2020	Benjamin Dauvergne
0001-csvdatasource-keep-inline-newlines-when-parsing-CSV-.patch	2,55 ko	10 avril 2020	Benjamin Dauvergne
0001-csvdatasource-keep-inline-newlines-when-parsing-CSV-.patch	2,37 ko	11 avril 2020	Benjamin Dauvergne
0002-csvdatasource-backports-fix-on-csv.Sniffer-41612.patch	7,07 ko	11 avril 2020	Benjamin Dauvergne