Michael Mannion

From: Michael Mannion
Sent: Thursday, May 30, 2019 1:25 PM
To: Danny Layfield
Subject: RE: Script Error

Danny,

The cause of the error is the single quote symbol used to indicate feet, as in the description for object ID 25272:

```
      Fence damage point, 75’ of damage
```

I’ll give a short summary of a long topic to help explain what is happening.

Computers store text as numbers. Each number has a symbol associated with it. If I store the number 39, for example (most) computers will draw a single quote symbol, ', like the one we typically use to mean “feet”. Take a look at the first chart on this page.

One problem we encounter when storing text in computers, however, is that there are lots of characters – over a million of them in common use (e.g. the Arabic characters we use in English, Chinese symbols, punctuation, etc.). Each of these characters needs its own unique number.

In the early days of computing, bytes were more precious than they are now. So, engineers commonly used a single byte to store the code for each character. Unfortunately, that only allowed for 256 characters. So, people storing text in English, Japanese, Arabic, etc. used different or *code pages* to assign which number meant which symbol. (Note that I’m not talking about fonts here. Think of fonts like handwriting; you and I can both write the letter “A” and, although it will look different, we mean the same thing. The code page determines the semantics, and the font determines the style.)

To improve interoperability, computing is largely moving towards the *Unicode* standard, which assigns a unique number to those million-plus popular characters. (This includes all of those emojis on your phone. 128513, for example, is a smiley face. It’ll look different with different phones/fonts, but we all agree that that number means a smile now).

Nowadays, with our faster computers and copious memory/storage, we’re largely moving towards a universal encoding called UTF-8. UTF-8 describes how to encode all of those Unicode numbers into a variable number of bytes (common characters get fewer bytes, less common ones require more bytes).

Unfortunately, there are still plenty of holdovers from the bad-old-days of computing – including some Windows and Python 2.x defaults, which are causing our problem here. To help minimize these problems, UTF-8 cleverly uses the “old style” encoding for some characters. If you stick to the characters/punctuation we commonly use in English, UTF-8 works just like the old encodings with the old code pages. If you use one of the extended characters, though, problems can occur.

Back to the specific issue at-hand, the single-quote symbols in the problematic fence damage points use a “fancy” single quote symbol – number 2019 – instead of the “normal” single quote symbol, number 39. Because the number 2019 doesn’t fit into a single byte, my Python 2 script couldn’t write it to the CSV file by default. I can fix that by forcing the output to UTF-8.

We’ve just been lucky so far, in that none of the comments (or other attributes) have used extended characters. So, the UTF-8 characters from SQL Server were able to sneak into my old-style Python 2 script without causing a problem. Until now.

So, we have two things to address:

1. I need to fix my script to write the CSV file in UTF-8 encoding, so it won’t fail when we get a non-standard character.

2. We should look into why Collector and/or iOS and/or c_taylor’s specific device is using that fancy right-quote symbol instead of the normal straight-quote one

#2 is probably something akin to Word’s “replace straight quotes with smart quotes” setting, where the software causes compatibility issues like this in the interest of making the quotes (ostensibly) look better.

I’ll post a fix for #1 shortly and let you know when it’s ready. Once I do, I’ll call to discuss whether / what we should do about #2.

Thanks,

-Michael

Michael Mannion
Database Consultant | Mannion Geosystems | +1 617.939.9958 | http://www.manniongeo.com

From: Michael Mannion
Sent: Thursday, May 30, 2019 12:04 PM
To: Danny Layfield <Danny.Layfield@nwfwater.com>
Subject: RE: Script Error

Danny,

It looks like all of the Fence Damage Points for the Altha tract are failing, except for one: object ID 25280.

The comment for that one differs from the others - it just says “Repair completed” instead of “Fence damage point” followed by a description of the damage.

- Can you think of anything off of the top of your head that would make the Altha points different than the others?

I’m thinking of something like a different device, OS, or Collector version for user c_taylor. I doubt it’s something that the user is doing, but rather something differently in the way their software is encoding the values they’re typing in (just a working theory at this point).

Don’t spend your time digging; I’ll figure it out. I was just curious if there was something obvious that I might be missing, not knowing the users/devices/etc. as well as you do.

Thanks,

-Michael

Michael Mannion
Database Consultant | Mannion Geosystems | +1 617.939.9958 | http://www.manniongeo.com

---

From: Michael Mannion
Sent: Thursday, May 30, 2019 11:36 AM
To: Danny Layfield <Danny.Layfield@nwfwater.com>
Subject: RE: Script Error

Danny,

There’s definitely something with character encoding, but I haven’t figured out the root problem yet. Just wanted you to know I’m still working on it.

-Michael

Michael Mannion
Database Consultant | Mannion Geosystems | +1 617.939.9958 | http://www.manniongeo.com

From: Michael Mannion
Sent: Thursday, May 30, 2019 10:14 AM
To: Danny Layfield <Danny.Layfield@nwfwater.com>
Subject: Re: Script Error

I'll be right there.

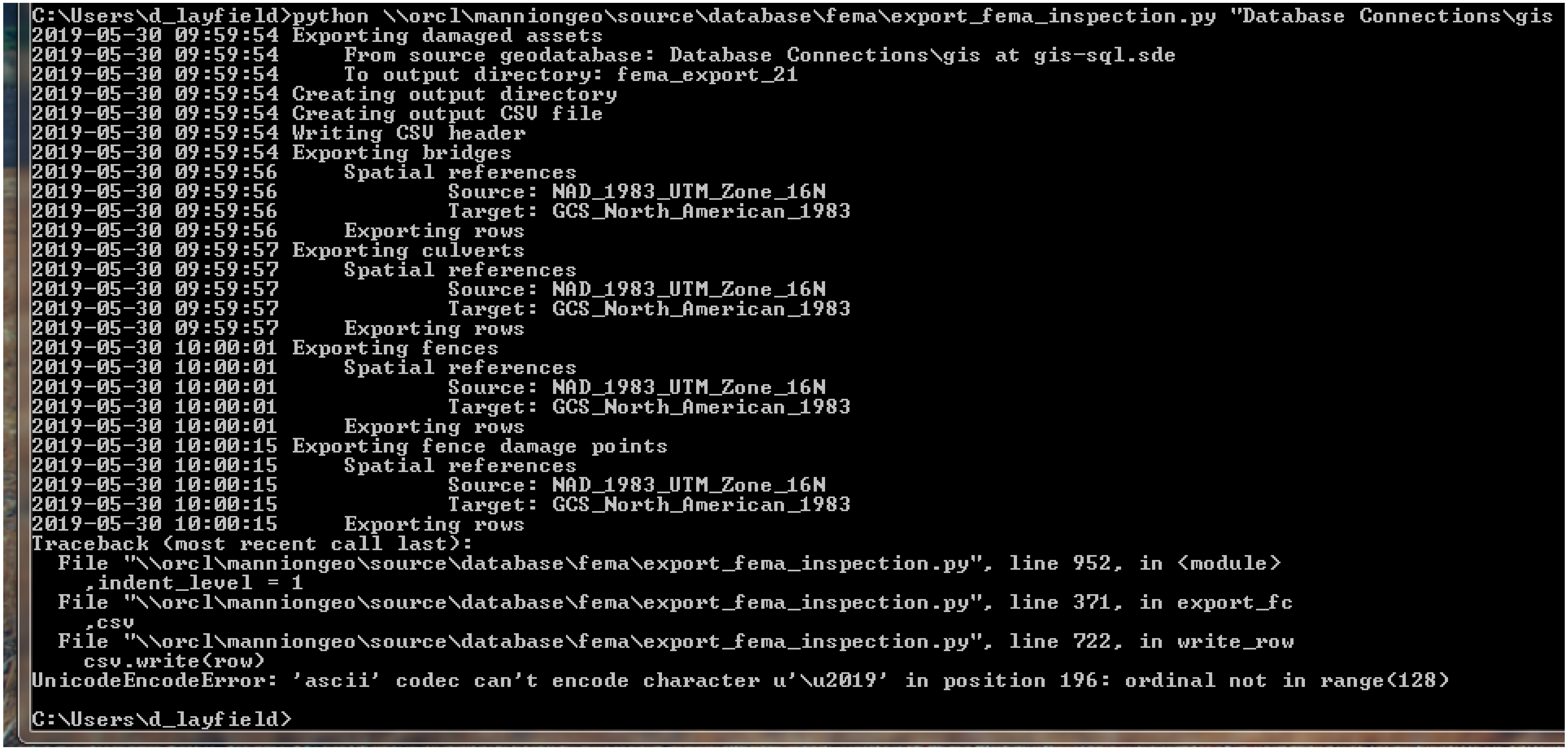
This error rings a bell. I think it's failing on a special character. I should be handing everything as Unicode, but might have missed a string somewhere. Or I'm just wrong :)

Sent from my phone

From: Danny Layfield <Danny.Layfield@nwfwater.com>
Sent: Thursday, May 30, 2019 10:10 AM
To: Michael Mannion
Subject: Script Error

Hi Michael,

The script returned an error when I ran this morning. The only thing that changed in the database was some of the miscellanous fence point were moved to improve location accuracy.



```
C:\Users\d_layfield>python \\orcl\manniongeo\source\database\fema\export_fema_inspection.py "Database Connections\gis
2019-05-30 09:59:54 Exporting damaged assets
2019-05-30 09:59:54     From source geodatabase: Database Connections\gis at gis-sql.sde
2019-05-30 09:59:54     To output directory: fema_export_21
2019-05-30 09:59:54 Creating output directory
2019-05-30 09:59:54 Creating output CSV file
2019-05-30 09:59:54 Writing CSV header
2019-05-30 09:59:54 Exporting bridges
2019-05-30 09:59:56     Spatial references
2019-05-30 09:59:56             Source: NAD_1983_UTM_Zone_16N
2019-05-30 09:59:56             Target: GCS_North_American_1983
2019-05-30 09:59:56     Exporting rows
2019-05-30 09:59:57 Exporting culverts
2019-05-30 09:59:57     Spatial references
2019-05-30 09:59:57             Source: NAD_1983_UTM_Zone_16N
2019-05-30 09:59:57             Target: GCS_North_American_1983
2019-05-30 09:59:57     Exporting rows
2019-05-30 10:00:01 Exporting fences
2019-05-30 10:00:01     Spatial references
2019-05-30 10:00:01             Source: NAD_1983_UTM_Zone_16N
2019-05-30 10:00:01             Target: GCS_North_American_1983
2019-05-30 10:00:01     Exporting rows
2019-05-30 10:00:15 Exporting fence damage points
2019-05-30 10:00:15     Spatial references
2019-05-30 10:00:15             Source: NAD_1983_UTM_Zone_16N
2019-05-30 10:00:15             Target: GCS_North_American_1983
2019-05-30 10:00:15     Exporting rows
Traceback (most recent call last):
  File "\\orcl\manniongeo\source\database\fema\export_fema_inspection.py", line 952, in <module>
    ,indent_level = 1
  File "\\orcl\manniongeo\source\database\fema\export_fema_inspection.py", line 371, in export_fc
    ,csv
  File "\\orcl\manniongeo\source\database\fema\export_fema_inspection.py", line 722, in write_row
    csv.write(row)
UnicodeEncodeError: 'ascii' codec can't encode character u'\u2019' in position 196: ordinal not in range(128)

C:\Users\d_layfield>
```