

Re: Best way to parse file into db-type layout?

Source: <http://coding.derkeiler.com/Archive/Python/comp.lang.python/2005-05/msg00434.html>

- *From:* Peter A. Schott <paschott@xxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Tue, 03 May 2005 19:12:07 GMT
-

Thanks for the comments. Quick notes:

1. Yes, ultimate goal is to output various fields into Excel, but I need to enter the data into a DB as well to avoid having any strange problems. That may be handled by a completely different process. I should have been more exact in saying that my current goal is to get this data into a database.

2. I'm relatively new to Python and definitely new to trying to parse Ragged-right type CSV files that are pairs of data. I'm trying to get my syntax correct when wording my questions/phrases – please bear with me. :-) Your help (all posters) is quite appreciated.

3. regarding read/write each row – I started thinking about working my way through the dictionary. This file could have up to 1900 pairs of data per row. That's an awful lot of columns to put into a database table, especially if quite a few will be NULL. I starting thinking of stepping through each pair by each row. The rows are identified by Letter type and an account number (always the same field #). I would want to use those, and insert something like:

LetterType, Account #, Field#, Field Data, (current date)

I wasn't quite sure how to phrase that and apologize for being unclear. I guess if I'm doing row-by-row, I can hold that for each row without too much trouble.

4. No control over file format. I can define a lookup table that will match Field# to a real name, perhaps code that into my scripts, but I'm stuck with that. As to other posters' responses – it's ANSI-coded, CSV, No Quotes (no Commas in values). Coming from the mainframe, I don't have to worry about embedded special characters within the file as they can't be entered into these fields to start with.

File always starts with letter_type, 1 – no idea why they didn't do that as 0, letter_type for consistency but it isn't my file. (Heck, I would have just specified to output all fields all the time and avoid any need to do special parsing :-)

Anyway, I appreciate the help. The code looks like it will work for me and I'll be doing some basic tests to get up to speed on splitting/parsing the data and putting it into a form that will be usable by all. I may have to write certain rows out to a new file if they don't match what I'm looking for. I don't anticipate any issues there – just read the row, write it to a new file if it doesn't match my criteria.

Re: Best way to parse file into db-type layout?

Thanks to all. Off to search for dictionaries and databases now. :-)

-Pete

John Machin <sjmachin@xxxxxxxxxxxx> wrote:

> On Fri, 29 Apr 2005 18:54:54 GMT, Peter A. Schott
> <paschott@xxxxxxxxxxxxxxxxxxxx> wrote:
>
>> That looks promising.
>
>> The field numbers are pre-defined at the mainframe level.
>
> Of course. Have you managed to acquire a copy of the documentation, or
> do you have to reverse-engineer it?
>
>> This may help me get to my ultimate goal which is to pump these into a DB on a
>> row-by-row basis (:-P)
>
> That's your *ultimate* goal? Are you running a retro-computing museum
> or something? Don't you want to *USE* the data?
>
>> I'll have to do some playing around with this. I
>> knew that it looked like a dictionary, but wasn't sure how best to handle this.
>>
>> One follow-up question: I'll end up getting multiple records for each "type".
>
> What does that mean?? If it means that more than one customer will get
> the "please settle your account" letter, and more than one customer
> will get the "please buy a spangled fritzolator, only \$9.99" letter,
> you are stating the obvious — otherwise, please explain.
>
>> Would I be referencing these by row[#][field#]?
>
> Not too sure what you mean by that — whether you can get away with a
> (read a row, write a row) way of handling the data depends on its
> structure (like what are the relationships if any between different
> rows) and what you want to do with it — both murky concepts at the
> moment.
>
>>
>> Minor revision to the format is that starts like:
>> ###,1,1,val_1,....
>
> How often do these "minor revisions" happen? How flexible do you have
> to be? And the extra "1" means what? Is it ever any other number?
>
>>
>>
>> I think right now the plan is to parse through the file and insert the pairs

Re: Best way to parse file into db-type layout?

Re: Best way to parse file into db-type layout?

>>directly into a DB table. Something like RowID, LetterType, date, Field#,
>>Value.
>
> Again, I'd recommend you lose the "Field#" in favour of a better
> representation, ASAP.
>
>> I can get RowID and LetterType overall, date is a constant, the rest
>>would involve reading each pair and inserting both values into the table. Time
>>to hit the books a little more to get up to speed on all of this.
>
> What you need is (a) a clear appreciation of what you are trying to do
> with the data at a high level (b) then develop an understanding of
> what is the underlying data model (c) then and only then worry about
> technical details.
>
> Good luck,
> John
>

• *Follow-Ups:*

◆ *Re: Best way to parse file into db-type layout?*

◇ *From:* John Machin

- Prev by Date: *Re: Read / Write image file*
- Next by Date: *Re: Read / Write image file*
- Previous by thread: *Re: Best way to parse file into db-type layout?*
- Next by thread: *Re: Best way to parse file into db-type layout?*
- Index(es):
 - ◆ *Date*
 - ◆ *Thread*