**Converting FIA data to a PSQL database**

**1. Overview**

The FIA data are available as a Microsoft Access database, or as text files in .csv format. This tutorial explains how to convert the latter into a PSQL database. This is easy if you already have a schema file, which is essentially a text file containing all the commands needed to create the an empty FIA database with all tables and variables appropriately defined. A schema file current as of March 2015 accompanies this tutorial. It can be used as long as the structure of the FIA database (i.e table and/or variable definitions) doesn’t change, even if the actual FIA data does get updated. If you know this is the case, use the existing schema and skip ahead to Section 3. If you’re not sure, skip ahead anyway and try it—if you don’t get any errors when importing the data then the existing schema still works. In the unfortunate event that a new schema does need to be created, the nex section will hopefully help.

**2. Create a PSQL schema for the FIA database**

*2.1 Convert the FIA Microsoft Access schema to PSQL*

USFS provides an empty MS Access file that specifies the structure of the database. This can be converted to the desired PSQL schema, although unfortunately the only way we’ve been able to do so is using a Windows-only application:

* Download the .mdb defining the FIA database (currently at <http://apps.fs.fed.us/fiadb-downloads/images/FIADB_version5_1.accdb>).
* Obtain the free Windows program *Access to PostgreSQL* (<http://www.bullzip.com/products/a2p/info.php>).
* Use the program to convert the downloaded .mdb file to PSQL format. Brief instructions:
* Run *Access to PostgreSQL*
* On the *Source Database* screen, choose the source file (the downloaded .mdb) but don’t worry about security settings. (We’re going to write to a file, not to an existing DB, so no security needed.)
* On the *Destination Database* screen select *Create dump file*, and set a file name for the output.
* On the *Select Tables* screen select all tables (should be default).
* On the *Transfer Options* screen the default values are fine.

*2.2 Fix the PSQL schema*

The result of the preceding procedure is a text file containing a complete description of the tables and variables in the FIA database. However, it has various errors (as of March 2015, at least) that will prevent a smooth import of the FIA data. Here are notes on how to clean the thing up to make it usable:

* Remove “CREATE DATABASE” command
  + Open the file in a text editor and delete the line  
    CREATE DATABASE “”;
    - It’s the first one that doesn’t begin with ‘--‘
* Convert to lowercase
  + Keeping all table and variable names lowercase will avoid problems later, but in the file you just created they are all uppercase. One way to fix this is:  
    cat mixed\_case\_input\_file | tr '[A-Z]' '[a-z]' > all\_lowercase\_output\_file
* Add missing table definitions
  + Any tables that are available to download (as .csv) but not defined in the converted schema will have to be defined by hand.
    - E.g., as of March 2015, the schema available from USFS is missing five tables: plot\_regen, seedling\_regen, subplot\_regen, tree\_grm\_estn, beginend
  + Define these by adding new “create table” entries in the schema
    - Follow the format of the other tables in the PSQL schema
    - Find names and types for all variables in the definitive FIADB User Guide.
* Fix data types
  + Some variables may not be defined correctly in the schema. You’ll know this is the case if you get errors about data type mismatch when trying to copy in the .csv data.
  + Required corrections as of March 2015 are below. Actual change is in bold, the rest is notes.
    - Table dwm\_coarse\_woody\_debris, field cover\_pct\_rgn
      * User Guide defines these as “NUMBER” (no precision specified)
      * Schema defines “integer”
      * Other variables that are classified the same way in the Guide (i.e. “NUMBER”) are given variable type “double precision null” in the schema
      * **Change variable type in schema to “double precision null”**
    - Table pop\_estn\_unit, field p1source
      * Guide: “VARCHAR2(50)”
      * Schema: “varchar(30)”
      * **Change type to “varchar(50)”**
    - Table subplot, field waterdep
      * Guide: “NUMBER(2,1)”
      * Schema: “integer default 0”
      * **Change type to “float null default 0”**
    - Table ref\_fiadb\_version, fields created\_... and modified\_...
      * **Changed to match other tables**
    - Table ref\_species, fields dwm\_carbon\_ratio and standing\_dead\_decay\_ratio#
      * Guide: “NUMBER(6,5)”
      * Schema: “integer”
      * **Change type to “double precision null”**
* Add missing variables
  + Some table definitions may be missing required variables entirely. These need to be added by hand.
  + As of March 2015, there is only one such instance:
    - Table ref\_pop\_attribute, field cn
* Reorder fields
  + PSQL apparently does not match column headers to field names when copying data in from .csv. As a result, the table definitions must list fields in the same order that the .csv columns occur.
  + As of March 2015, changes required are
    - Table dwm\_visit
    - Table ref\_fiadb\_version

**3. Create an empty database from the FIA schema**

Whether you obtained an existing PSQL schema or created your own following the above procedure, it can be imported to a PSQL server at the command line as follows (replace bold variables as appropriate):

psql -U **username** -d **new\_database\_name** < **PSQL\_sschema\_file**

This will create the database and populate it with empty tables containing all the right variable definitions.

**4. Import FIA data to the empty database**

Now, raw FIA data downloaded from USFS in .csv format can be inserted into the database using the command:

psql -U **username** -d **database\_name** -c "\COPY **table\_name** FROM '**csv\_file\_name**' WITH CSV HEADER DELIMITER AS ',' NULL AS '' ENCODING 'WIN1252'"

Note that **table\_name** can be obtained from the .csv file name. E.g. for statewide files ‘XX\_BOUNDARY.CSV’ contains table ‘BOUNDARY’ for state ‘XX’. Table names should not be case sensitive, but file names are.

A few notes on encoding:

* The USFS .csv files may contain nonstandard character specifications, probably because they were dumped out of a Microsoft product.
* The one example as of March, 2015 is:
  + Table pop\_eval, field EVAL\_DESCR,
    - Rows 54-55 (cn=257338689489998, 257388037489998) contain ‘en dash’ characters
* PSQL thinks this file is encoded UTF-8, but these ‘en dash’ characters are not defined correctly under that encoding
* Based on my research I think the files are actually in WINDOWS-1252 encoding. Apparently MS products have a problem of using this encoding but incorrectly identifying it as something standard.
* There are several possible fixes:
  + 1. Manually replace offending symbols in any files that have them.
  + 2. Convert files, e.g. using iconv:  
    iconv -f WINDOWS-1252 -t UTF-8 < POP\_EVAL.CSV > POP\_EVAL.CSV
    - This can be run on all files safely—it has no effect if there are no special characters, but fixes files where they do occur.
  + 3. Specify the Windows encoding in the PSQL COPY command, with option ENCODING ‘WIN1252’
    - Again, I confirmed that this had no effect on a table without special characters, but it fixes the problem for a table that does have them.
    - Also checked and it seems to have no effect on speed of the copy command.
* Currently, I favor Option 3, as it doesn’t require modifying the FIA files at all. This is reflected in the command given above.

**5. Converting FIA data to a different database format, e.g. MySQL**

While not covered here, the procedure for importing FIA to a format other than PSQL are probably similar. The same software company that provides the Microsoft Access --> PSQL converter even has [equivalent programs for some other database formats](http://www.bullzip.com/download.php).