Property DataBase Application Technical Doc for UCR and CRM/RES Admin

* Source code is available in Python on the V: drive across multiple .py files.
* Compile with the command “python PropertyDB.py py2exe” while in the source code deirectory.

**Front End:**

**Source Code:**

**Passwords are NOT handled correctly! They are viewable by debugging the .exe and searching for ASCII or referenced strings. Also, they are hardcoded into the source code.**

**Database:**

DB name: inspect1

DB usern: inspect1 (admin)

All property tables start with “property\_” and there are domain tables for “Set” fields within the main viewable tables (property\_LnLs and property\_reserves) that start with “domain\_”

**See PropertyDB\_BackEnd.docx for table information.**

**Backend server:**

**PHP:**

Php scripts have been created to update tables, archive tables, and send emails regarding future end dates for lease and licenses. They are located on vhostdb1.ucr.edu under resdev/php/ directory. They are executed via a crontab viewable by crontab –e. The scripts all require a valid connection to the database which is stored in connections.php. connections.php is only viewable/editable from a login with administrator credentials, but it is executable by every user. One script will write files as csv’s to the server located under resdev/table\_archives/monthyear/ every month. **Passwords are handled and stored correctly.**

**Issue:** Day’s until value is incorrect (1 day off) from actual value

**Future Version:**

Fix security issues and ~~standardize pkeys to be the first entry as an AI int. Create alias tables to standardize schema. New classes for add and edit that pull domains from db and contain date pickers.~~ Fixed

Ability for the end user to add/delete fields to/from tables and for the application to react dynamically to them.

~~Could implement a “delete option” similar to “active” col. This “deleted” field would hold a yes/no flag that is defaulted to “no” and when the user “deletes” the entry, the flag gets toggled to yes. Then I could run a report that actually deletes entries (or moves them to a new table)~~ Just made the delete button set ‘Active Entry’ = No.

File and parcel handler could be updated to delete on the primary key rather than a set of entries for a particular row

~~All “\*\_files” tables should be combined into one with a new field for identifier. Would need a way to separate files based on table and entry.~~ Maybe not the best idea

**THIS MAY ALREADY BE HANDLED BY THE DATABASE ON SOME LEVEL:** Query heavy. It should have a middle connection that manages queries and caches select query tables until tables are updated and then re-cache the table (LnLs, reserves, files, apn, aliases, domains). This will take **a lot** of work. Like seriously. It would require a rewrite of all commands to the database to route them through some other program or application that would then handle the calls. (or I could find and use some api, but screw that, I write my own code).

Basic solution: App <---talks to---> middle server< ---talks to--> database

Alternatively: Have the tables cached locally (on the v:\) and have the application re-pull tables on updates. (this can lead to compromise because it will allow the user to modify the cached tables without using the application and may trick users into seeing false information)

^Probably only needs to be implemented if people actually use the application.

**Noted Issues:**

~~When editing an entry, it will always update the dates regardless if there is a change or not. This is because of converting dates back and forth between yyyymmdd and mmddyyyy. Essentially, the edit function always recognizes a change in date because it compares the true date in the DB to the converted on in the application.~~ Fixed 8/19/2016 by Mark Saxer – Used a “try convert to date function”

~~The program crashes after being on standby for an extended period of time (~20 mins). I believe it has to do with refreshing a dead connection to the database.~~ Fixed 8/22/2016 by Mark Saxer – “try except” on query executions was missing an identifier. Self.db = DB.Database(). The DB was missing. Errors still may occur.

Security Issue with db calls. Using manual string concate when I should be letting the mysql library handle the calls. Execute(“sql\_string”,(values))

~~The reload table function passed into the Add\_Edit class does not seem to be working. The table does not reload after updates.~~ Fixed

~~Warnings when adding values into db because the values are truncated is left blank~~ Fixed

~~There is no input validation done by the application. Even though there are combobox dropdowns for a handful of fields, the user is not required to pick one within those boxes.~~ The user is now forced to pick a dropdown option. ~~Still no validation on text or date or int fields though.~~ Integer and Decimal field types are now validated with a regex and add/updates will not be made with invalid entries. All values (except sets because they are forced) are validated in some way (length or values)

Query heavy. Hurtful to database.

A lot of libraries are fully imported. Many only need to import a function or 2. Could change a variety of import to the format “From %s import %s”

Did not use a dragto function for selecting multiple rows via mouse movement. Dragging to select multiple rows does not work properly because it doesn’t select everything if moving the mouse too quickly.

~~Property log no longer works with %s in queries.~~ Maybe

Also, the deltas should be saved on updates rather than logging the update query.