Questions

* How long is the average name? Like, “MAX7221” or “8 Digit Seven Segment Display Driver”?
  + The unique identifier will be the NXDN (see below) and be something like the following:
    - NXDCnnnn – This would be a Nextdroid Component, Capacitor, and nnnn would be a serially enumerated ID…starting at 001
* How long are typical part numbers?
  + The part number will be 7 to 8 characters long

Notes

* database changes live on GitHub, file lives locally on disk (database will live in the repo with the app)
* schematic files and datasheets live in subfolders, can assume the files are in the right place when selecting
* database changes are pushed to GitHub on form close (submit changes the database, closing the window commits and pushes) – probs also add a button for manual push?
  + repo is going to be in the ND org so I need to pull credentials from an existing Git installation
* file locations are just links, not actual files - no need to open the file from the DB viewer

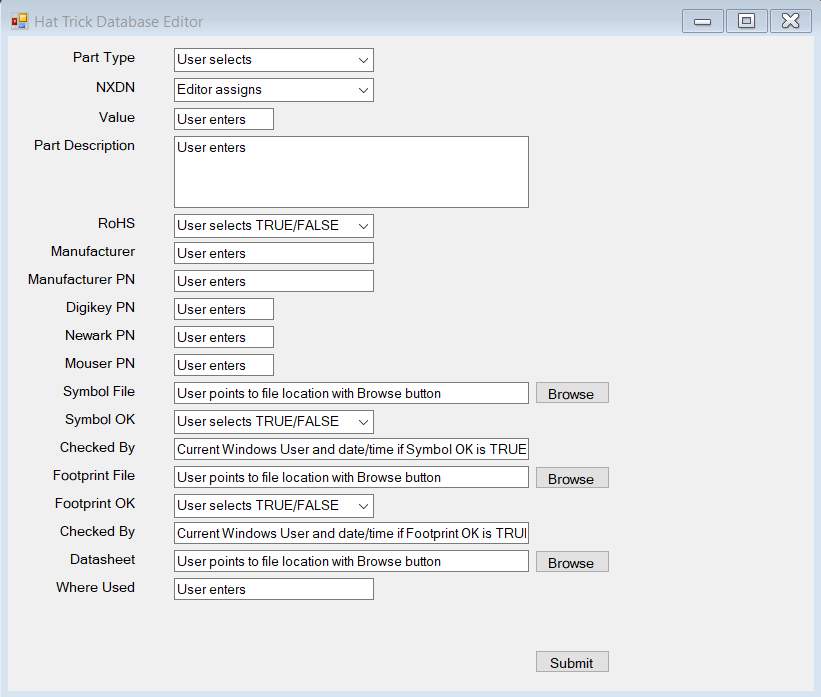
Create an app that accesses the component symbol/footprint database and displays one entry at a time. The required fields to be displayed are as follows:

* Name
  + The name field will actually be ‘NXDN’ for Nextdroid Number. NXN is the key field in the database.
* Value
  + This will contain data like ‘10uF/25V’ for caps, ’10.0k/1%’ for resistors, etc.
* Description
  + This will be called ‘Part Description’
* Manufacturer
* Manufacturer Part Number
  + This will be called ‘Manufacturer PN’
* Supplier
  + This should be broken up into three fields:
    - Digikey
    - Newark
    - Mouser
* Supplier Part Number
  + This, then, should be broken into three fields:
    - Digikey PN
    - Newark PN
    - Mouser PN
* Price
  + Price is not necessary
* Schematic Symbol (image)
  + We won’t need an image
* Footprint Symbol (image)
  + We won’t need an image
* Notes
* Applications (where the part has been used previously)
  + This one can be called ‘Where Used’
* Datasheet / Attached Documents

The user must be able to browse a list of entries with the ability to filter based on a free text search across every property of every entry.

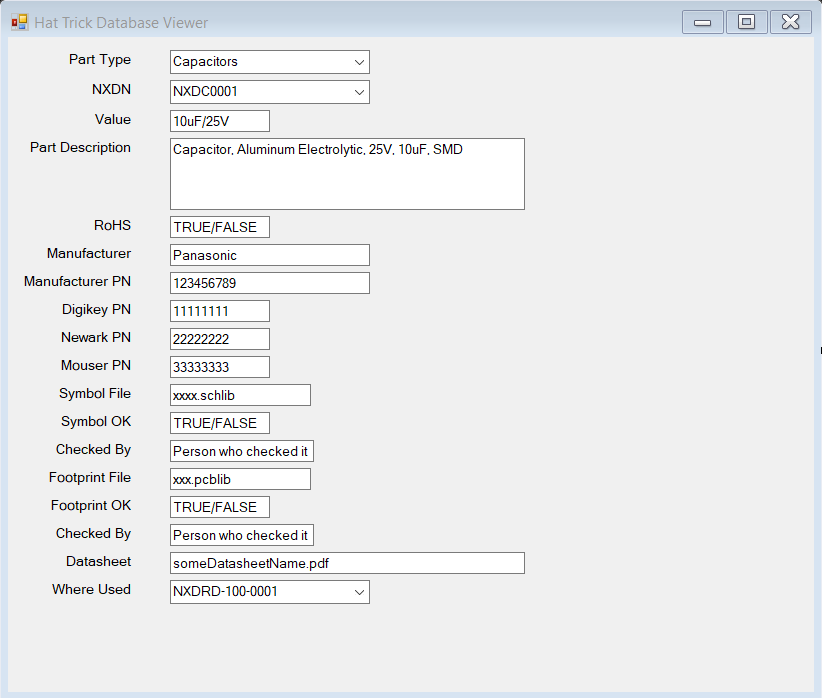
Clicking on an applicable data point (manufacturer, supplier, application) should bring up a selection window pre-loaded with a search on that metric.

Editor ideas



* When the editor is opened, it opens a connection to the database (will only allow one user at a time)
* The combobox for part type should auto-populate with a list of all the tables in the database
  + There will be several tables: one for resistors, one for capacitors, one for ICs, etc. The app should query the DB and report all that exist
* The textbox next to NXDN should auto-populate with the next available record ID in the selected table. The app will need to read back the information from the database to determine the next available ID.
  + For example, Resistor is selected under part type, and there already exists 12 unique parts in the Resistor table, the next available record ID will be NXDR013, which is what should populate in the NXDN text field.
* There should be a black text box next to the Value label for a user to enter the value information
* There should be a multiline text box for the user to enter the Part Description information
* There should be either a combobox or radio button (no preference) for the user to select whether the part being edited is RoHS compliant (will equate to a TRUE or FALSE in the database)
* There should be a text box for the user to enter a Manufacturer
* There should be a text box for the user to enter a Manufacturer PN
* There should be a text box for the user to enter a Digikey PN
* There should be a text box for the user to enter a Newark PN
* There should be a text box for the user to enter a Mouser PN
* The user should be able to browse to a symbol file on their computer
* There should be either a combobox or radio button (no preference) for the user to select whether the part being edited has been checked for symbol correctness (will equate to a TRUE or FALSE in the database). If made to equal true, the app should populate current Windows user and date/time in the text field below. Date/time might make more sense as a separate field…
* The user should be able to browse to a footprint file on their computer
* There should be either a combobox or radio button (no preference) for the user to select whether the part being edited has been checked for footprint correctness (will equate to a TRUE or FALSE in the database). If made to equal true, the app should populate current Windows user and date/time in the text field below. Date/time might make more sense as a separate field…
* The user should be able to browse to a datasheet file on their computer
* The user should be able to enter schematic part number into the Where Used text box to add to the database (need to think this through a little more…how best to be able to add multiple? How many should we support? I’ll think on this…)

Viewer ideas



* User should be able to select the particular table of interest from the Part Type combobox
* The NXDN combobox should populate with all the relevant parts in the selected table, starting with the first entry.
  + The example above, a user has selected the Capacitors table. NXDN then displays all records, starting with NXDC0001. Expanding the combobox, you’d see successive entries, all the way up to the last entry before a white space (NXDC0002, NXDC0003…NXDCnnnn)
* Part Description, RoHS, Manufacturer, Manufacturer PN, Digikey PN, Newark PN, Mouser PN, Symbol File, Symbol OK, Checked By, Footprint File, Footprint OK, Checked By and Datasheet populate with static data as read from the DB
* Where Used should be a combobox (I think) that fills in with any non-whitespace entries from additional schematics where the part has been used.