New project to expand the tax calcs…

Starting Date: 6/15/17

Branch: BWM-28

Things that need to be done:

* Is there a special order to what gets pushed… Fill in as needed.
  + Sales tax calcs are huge… we do need to push this early or late in the day (non-peak hours). There are some dependencies and we are changing some database fields.
  + Database updates and scripts
  + CFC’s
  + Code
  + Maybe do one server at a time so we can check everything…
* Build the database update page (script it)
  + Add new fields for 6 – 10 on custom tax fields. Done on 6/29/17.
  + Add new fields for all 10 limit by customer type list fields. Done on 6/29/17.
  + Change all decimal(5,2) fields to decimal(8,5) – this is for 5 decimal places. Done on 6/29/17.
  + Update all new store\_location lists to the value of “all”. Done on 6/29/17.
  + We may also want to add in some custom filters and such for the state, county, and city taxes. We may want those dynamics up there in the top level. Done on 6/29/17.
  + Add the version control database update script. The link was missing as of 6/15/17. We want to record this change as it is fairly big. Done on 6/20/17.
  + On the existing store and location table… We may want to change the other\_[1-10]\_account\_id to other\_[1-10]\_owe\_to\_payee\_id (who do we owe this collected tax to) and the other\_[1-10]\_tax\_fee\_id to other\_[1-10]\_expense\_type\_id (what expense type to tie it to – the expense type will also have and hold the chart of account numbers through a backwards look-up). This will be a drop-down and may be left at a value of 1 if unknown. Done on 6/29/17.
  + When adding in the dynamics to the normal state, county, and city taxes…. What if we added in similar dynamics for the other account ids, other fees (see notes above for new naming). We may need the numerics from above and the new varchar fields for what there tax id’s are (these could be strings – id numbers given to the different businesses). Done on 6/29/17.
  + Script the quote line items changes
    - Change all columns that collect taxes to 5 decimals. Done on 6/29/17.
    - Add the new 6-10 custom tax fields. Done on 6/29/17.
    - Add a new column to help with tax rounding so we don’t have to hurt the state tax calc field. Done on 6/29/17.
    - We want a new field to help us know if the taxes were auto calculated or manually calculated. Done on 6/29/17.
    - We have some new mini conversion stuff… maybe think about getting all of the new tweaks done at once. Done on 6/29/17.
    - We may want to run the dollars off discount stuff as well. Done on 6/29/17.
    - Run the script to update all line items
  + Script the invoice line items changes (this is corp-specific) this will be a big click, click, click migration.
    - Change all columns that collect taxes to 5 decimals. Done on 6/29/17.
    - Add a new column to help with tax rounding so we don’t have to hurt the state tax calc field. Done on 6/29/17.
    - Run the script to update all line items – this will be big
    - We have some new mini conversion stuff, dollar off discount stuff, elements of time, and sub inventory stuff… maybe think about getting all of the new tweaks done at once. What does this look like.? Done on 6/29/17.
      * As a side note… the sub inventory stuff needs two columns. One is the normal int(10) for the sub reference id and the other will be a varchar 255 for a possible list of sub reference id’s. Defaults are 0 for the int and blank for the varchar. Done on 6/29/17.
    - Alan will be adding in the time\_id on po\_invoice\_lines\_[53] table. Done on 6/29/17.
    - We scripted this in both ColdFusion (done by Brandon) and Calvin’s auto process but both took quite a bit of time… Calvin’s runs without extra clicks but still takes hours to cycle through multiple corporations. Calvin’s update rate is about 1,500 records per 1 second. It may be run unattended but still takes tons of time. Brandon’s takes thousands of clicks per server and does about 3,000 records in about 8 seconds.
      * As a side note, and as a mini break through, we may want to setup an internal process (internal user queue) and help to harness the users clicks and/or page navigation. That could be huge and could really help us get some extra work done without taxing any one person or company too heavily. Fun idea and concept.
* Work on the add\_edit\_store.cfm page…
  + Add the new 6 – 10 custom values. Done on 6/29/17.
  + Add the new 1 – 10 customer type lists with builders. Done on 6/29/17.
    - Make sure that builders work… default to All. Done on 6/29/17.
  + Expand the samples for the percentages, expand the max number of digits for the percentage fields. Maybe show a max of 5 decimal places. Done on 6/29/17.
  + There is both add and edit modes… both need to be updated and checked. Done on 6/29/17.
  + Maybe help format things back to the user so that they can see that there are 5 decimal points of accuracy. Done on 6/29/17.
  + The body will be in the secure/classic folder. Done on 6/29/17.
  + The old field that say “state tax id” maybe rename to “state tax license number”. We also need a new field for all of the other custom tax fields. Done on 6/29/17.
  + Maybe help the part category builder to remember the prior choice. Also, maybe include the word discount in the top selection for Internal Use – N/A System (id=2). Maybe in the add mode, precheck that box to help them. This is a helper but may not be the main priority. Done on 6/29/17.
  + Steve is wanting to pull in the state to the add/edit location page. This is on the add mode and gets pulled from the corp level. Extra icing stuff. Done on 6/29/17.
  + Look through changes made by Steve. He was working hard on this file. Done on 6/29/17.
* Add the new values to the action page add\_edit\_store\_action.cfm
  + Setup new params, validation, update loops, etc. Done on 6/29/17.
  + Add new fields to the add method and the update method. Done on 6/29/17.
  + There may be a rouge argument called storeOtherTax that doesn’t do anything. Done on 6/29/17.
  + Look through changes made by Steve. He was working hard on this file. Done on 6/29/17.
* Make sure and check the code when a new corp is created. A new location is also created at the same time. Also, when making a new po/invoice line item table, there will be new fields and values there as well. Done on 6/29/17.
* We want to include a history note about who changes what on the store and/or location level. This is also true on who changes what on the different tax settings. Maybe even a review of what it was vs. what it is after the change. We ended up storing JSON in the 3rd party error logs… currently not showing it but we are recording it. Done on 6/29/17.
  + This may include passing in old values or looking them up before doing an update. We may want to record even down to the detail level of values and percentages. Only record where changes occur. Done on 6/29/17.
* On locations… we will eventually need both a start and an end date to help with the life cycle of the those pieces. This is more for the balance sheet, but will be needed at some point. Done on 6/29/17.
  + Make the end date conditional based off of status. Done on 6/29/17.
* On banks… not part of this project, but we really need both start and end dates on banks.
* We are also seeing the ice-down date starting to play into things. At some point, we need to lock things down and keep things stable. Think of water forming into slush, snow, and eventually ice.
  + Instead of doing the ice-down clear down to a crazy deep level (system wide)… what if we roll in the ice-down date for sales tax. Start the process.
  + Steve had the idea of breaking the ice-down into smaller pieces and sections.
  + As a side note… we do have a high-level ice-down setting (master date of sorts). That setting might not be enough… think one-to-many on almost every level. We are seeing the entire system as a series of relationships and logic that plays into the puzzle. Maybe an ice-down date for sales tax, bank stuff, inventory stuff, sales, etc.
* Go over the calculateTaxes method with a fine-tooth comb. This is the main crux of the system and all taxes are calculated
  + Very important to check the number of decimals and run through tons of different scenarios… Try to trick it and break it.
  + Add the new logic for the limit by customer types – test this out
  + Follow all occurrences of this method… you will have internal carts, invoices after the fact, edits on invoice line items, recipe stuff, custom code, and ecommerce. Most things we do have functions that have a way to add them, a way to update them one by one, and a way to update things in bulk. Be aware of that.
  + Remember the new rounding error bucket.
  + We need to figure out what to do if the part category filter or customer type filter is not available. Kind of an order of operations. As a side note, we did a first-round test on the tax calcs… see the “getTheTaxRate” method on the cfc/cart.cfc page. It already has some concepts and logic built in as a front runner.
* There is some JavaScript that calculates taxes on the advanced add to cart page, the advanced edit cart page, and even my cart favorite buttons (backing into taxes). We need to pull in the new pieces to those pages. It could get really tricky with the part category id and customer type id filters. Be aware of that as well as more variables and values playing into the calcs.
  + see the “getTheTaxRate” method on the cfc/cart.cfc page. It already has some concepts and logic built in as a front runner.
* Things get crazy with discounts…
  + This may need to be expanded and looked into.
  + On the corp-wide settings… We need to set the “Discount” item set to taxable by default. Done on 6/29/17.
  + We know that there is a dollars off discount project and it really needed… do we want to roll the two projects together or keep them separate? We decided to roll things all together.
  + See ADI-146 or pull request #166 for some older code dealing with in-line dollar off discounts. We have already spent over $4,000 on this project but have never rolled it out. Tons of changes since the code was written originally. Some of the original code was done by both Bryan and Brandon. We may be able to cherry pick some of that.
* We have add to cart stuff
  + Follow every known option in… basic search, advanced searches, view items, all kinds of barcodes, my cart favorite buttons (all different types), QR codes, sub inventory, recipes, restore to cart, duplicate to cart, ecommerce, etc.
    - We made a pretty good list on the .txt file called working\_on\_the\_sales\_tax\_expansion\_project.txt file. See that page for more notes and a list of ways of how things get into the cart.
  + We have to go through the stock/unit sales process… That is a multi-step process for big ticket items. This process could use some help… Maybe standardize what we can and help it roll through. We may need to circle back on this process later on. This could get us off track.
  + Lots of new math calcs and changing tons of formatting from 2 to 5 decimals. We need to be strategic on those changes and make sure we are capturing good enough data. Basically, don’t just globally force things to a certain value.
* We have edit or update cart stuff
  + Similar cascading things and checking all kinds of angles
  + There are different edit modes (basic, advanced, multi, and other custom options)
  + This may fit somewhere else, but on the quote line items and invoice line items tables… we want a new little flag that helps us know if the value was auto calculated or manually calculated (plugged). We need this inside the cart on both add and edit modes. It also needs to flow out of the cart and into quotes and invoices. Later on, we could pull reports on this.
* We will have printable quote stuff. Normal and PDF and ecommerce
* We will have printable invoice stuff. There are multiple versions of the invoice. There is normal, printer friendly, limited, summary, multi, mini version, PDF, ecommerce. We have bulk options we also have summary and tax summary invoices. There may be more…
* We will have report stuff. This could get pretty deep…
  + Sales tax reports (already 4 different layers exists)
  + What about reports based off of watchers and feeders?
  + Advanced invoice line searches
  + Super advanced invoice line searches
  + Sums and other values cascaded around to sales reports, daily/weekly, and even Balance Sheet totals. The balance sheet currently pulls some running totals. Where or what is the plan there?
* We are seeing some of the operations pieces flowing over to the financials. Keep looking to the full life cycle and full story of the objects and data over time.
* On watchers and feeders… we need to collect data per day, per category, and per location. We also need base numbers (what to run calcs on), what was collected (what was actually stored), what should be attributed (possible projections or calculations done after the fact), and what the difference are (diff = projected - actual).
  + We also need to record the human readable options (maybe 1-3 or 1-5 for special flags), what main table to look back at, and what column name and main id to reference. See graphic or screen shot.
  + If we don’t get all of the financial connections made at the first… make sure that it can still flow through using the human readable flags and such. Then when it gets fully connected in… we could do a reverse update back on the watchers and feeders to make sure that everything is updated and maintained. Think of this as a global backward re-alignment process or something.
  + On the watchers and feeders, add flags to help us know if they are a manual entry (standalone declarations) or auto process or auto calculated (system generated). This is huge to help us know what is going on… We can show this on reports and what not. Once again, this could help us show and maintain the things we want to call standalone declarations.
* On expense types, deposit types, and balance sheet types, open up the chart of accounts for Kelly. This is a whole other project.
* What about help files and new documentation? What is needed?
* What about news and updates and possible videos?
* What about new arguments for API calls for adding quote line items, calculating taxes, and adding invoice line items.
* What about updating a part number. If the unit of measure changes… what about all of the underlying details (quote and invoice line items) being updated… Does that affect any of the mini conversions… We could really make a mess if we leave things too open. Just a thought.
* We have a new variable dealing with taxes being calculated based off of the original price vs. the discounted price. Some examples are: tires, cigarettes, gas, etc. They have a thing called the “excise tax”. The tax needs to be collected on the original price (the advertised price or the published price) not the price after a discount. This is somewhat special.
  + Is this controlled on the store/location level, the part category level, or on the individual item level. We ended up going with a location level and then sub filters on the part category and even item level.
  + This could be a discount rule thing… for example: don’t allow discounts on such and such items or such and such part category.
  + We may need a setting that allows the items to say allow discounts or not.
* We are seeing a few key pieces playing into the cart…. time, discounts, coupons, pricing engines, tiers, quantities, barcodes, buttons, normal searches, QR codes, parents, subs, mini conversions, part categories, customer types, exceptions, recipe/builds, etc.
  + We are seeing that the cart needs to be smart and needs to know how each item plays based on the other pieces that are available and/or required. Truly smart cart logic.
* We need to pull in the tax settings based off of the individual customer. If they are set to say “government” what does that mean. Do we automatically flip the cart for them or what? Currently is just a setting that we show and maybe showing a small warning on the checkout page. How deep does that need to go?
* Once we get everything all cleaned up… we have a field on the quote line items table called “sub”. It is being replaced by one called “sub\_reference\_id”. It was left until everything gets fully updated and code is pushed up and live. This is an after the fact clean-up piece that is needed.
* As a side note… we are seeing that the PO/invoice line item table is really a hub of activity… eventually the watchers and feeders will sum up the data and make it easier to search. Also, we are seeing that we may need one-to-one relationships between sub inventory and PO/invoice lines as well as time id’s to PO/invoice lines. For the record, we are planning on adding in two fields per… one for the id’s (actual numeric values) and another for a possible list (varchars or list of id’s). This will help with the flexibility of that table to connect it with all kinds of stuff.
  + This is kind of a side question, but do we need a similar thing with the mini conversions? Not sure on that, it is still kinda out there and fuzzy but it may benefit from a similar two-part connection. Look deeper before jumping on this one.