End-of-Semester Reconciliation

Mike Robertson

Executive Summary

I am an accountant in BYU's Student Leadership department. One of my responsibilities includes reconciling all of the expenses we approved throughout the semester with all of the expenses that actually came through on our reports. Our department does most of the activities put on by the university, and it is mostly run by students. By the end of one semester there are hundreds of transactions to go through in ten different areas, handled by hundreds of students. This job is very tedious and can consume up to weeks of time.

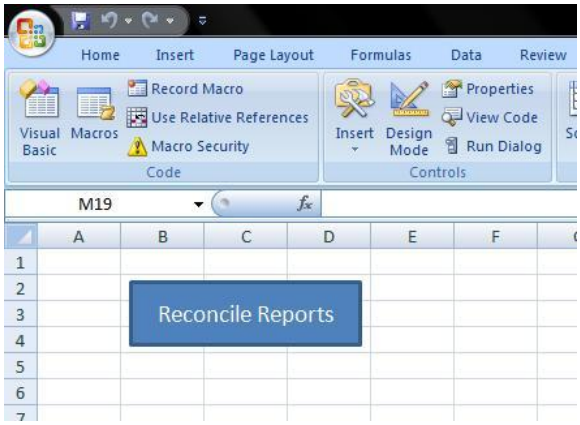
This program automates the semester-end reconciliation process by matching budgeted expenses with actual expenses and organizing them into four different categories: 1) already-reconciled and completed expenses, 2) expenses that need to be marked as reconciled, 3) budgeted expenses that never occurred and should be voided, and 4) actual expenses that occurred but were not budgeted for, and so must be documented in the database. Further review and processing is required for the last three categories of expenses, so the program also sorts them into separate spreadsheets for quick analysis.

Implementation Documentation

This program automates the semester-end reconciliation process in the following steps:

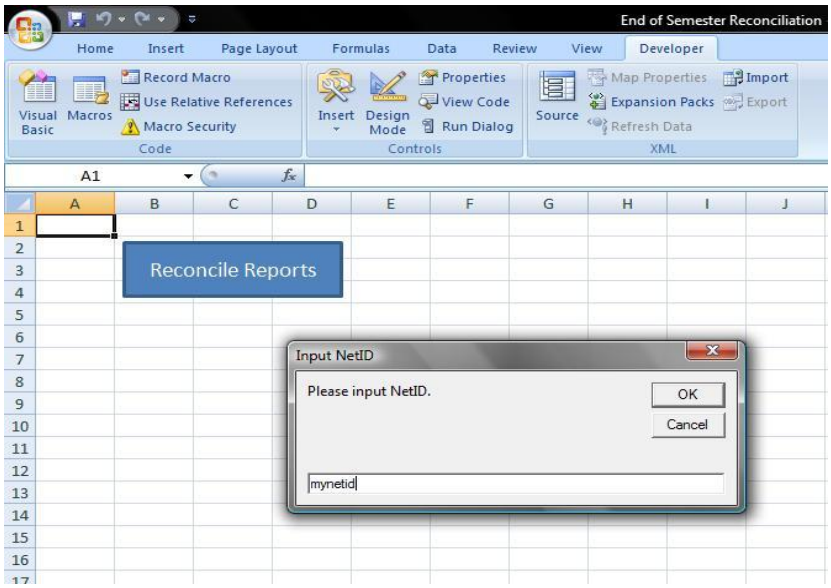
1. Logs into Route Y with user-provided NetID and password
2. Accesses BYU's Business Objects system to allow easy download of expense reports.
3. Looks up all budgeted expenses on the password-protected online database and plots them into Excel, organized by different areas within the department.
4. Matches all budgeted expenses that have already been reconciled with actual expenses from BYU's financial reports; marks them such that they will not be considered further.
5. Locates all budgeted expenses that have not been reconciled and matches them with actual expenses; marks these budgeted expenses as reconciled, and sets them also aside.
6. Takes all remaining unmarked budgeted expenses and puts them into a separate worksheet of budgeted expenses that should be reviewed and voided.
7. Takes all yet-to-be-reconciled budgeted expenses and puts them into a separate worksheet of budgeted expenses to be reviewed and reconciled.
8. Takes all remaining unmarked actual expenses and puts them into a separate worksheet of actual expenses to be reviewed and added to our online database of budgeted expenses.

The process begins simply by clicking the Reconcile Reports button in the Excel workbook, End-of-Semester Reconciliation. The macro that runs the program is attached to this button. The following steps then take place as part of the automated reconciliation process.



1. Logging into Route Y

The process of logging into Route Y was actually quite simple, as this very example was provided in class. The only change needed was to require the user to input NetID and password. This is accomplished through input boxes. The code then opens the Route Y login page, provides the authentication information, and submits it. Now, as long as the user is authorized to be viewing the required pages, the program will allow access and download of those pages.

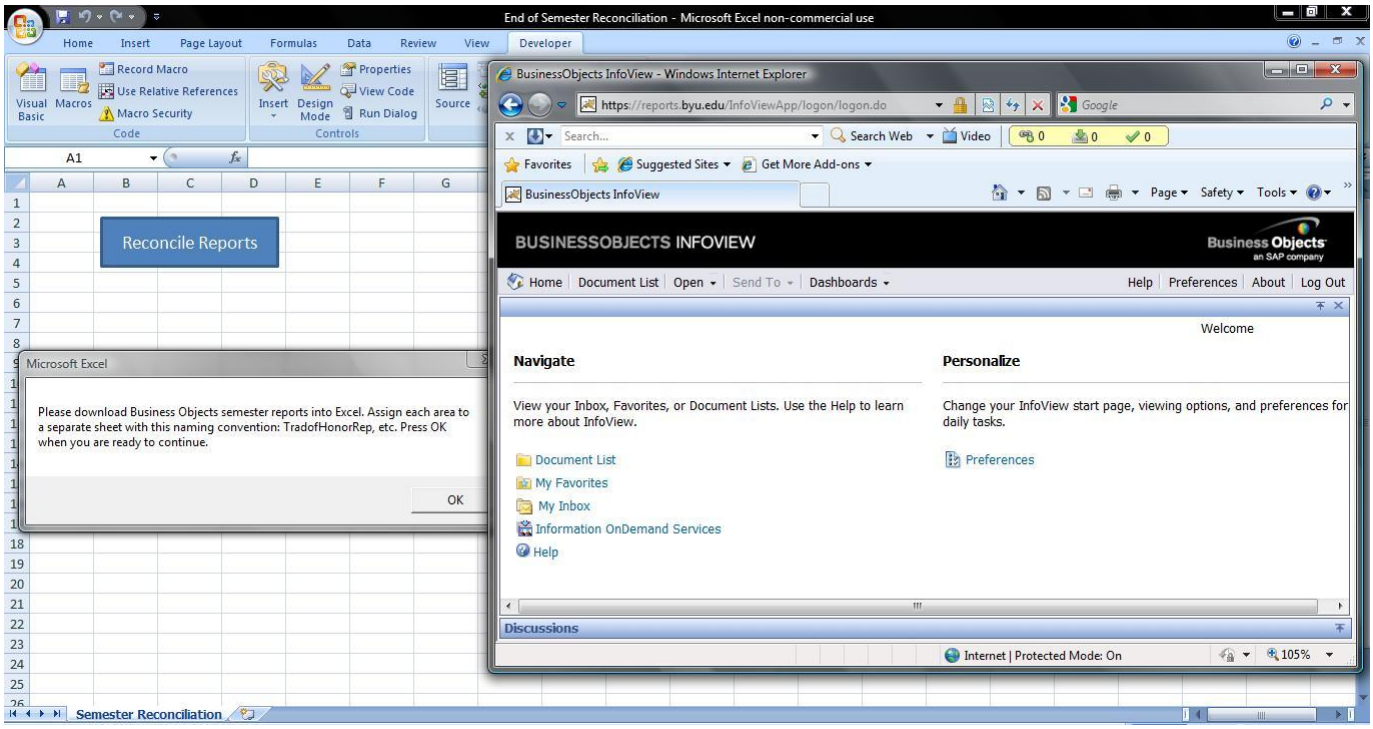


2. Accessing Business Objects

This step started out causing me problems. Using Professor Allen's "agent" class module, I was easily able to access the website for Business Objects. However, accessing pages within the website provided a huge challenge. For some reason, the link I needed to use was not accessible. I tried every bit of html code that I could possibly think of that might be tied to that link, but nothing worked. I asked Professor Allen about the problem, and he suggested I try running it using the javascript shown at the bottom of the page. I tried doing so with no better result.

I believe the problem I had was that the link did not actually take the user to a separate url. Business objects (at least the portion I needed to manipulate) was entirely self-contained within a single url; even when clicking links and running queries in Business Objects, the url remains constant.

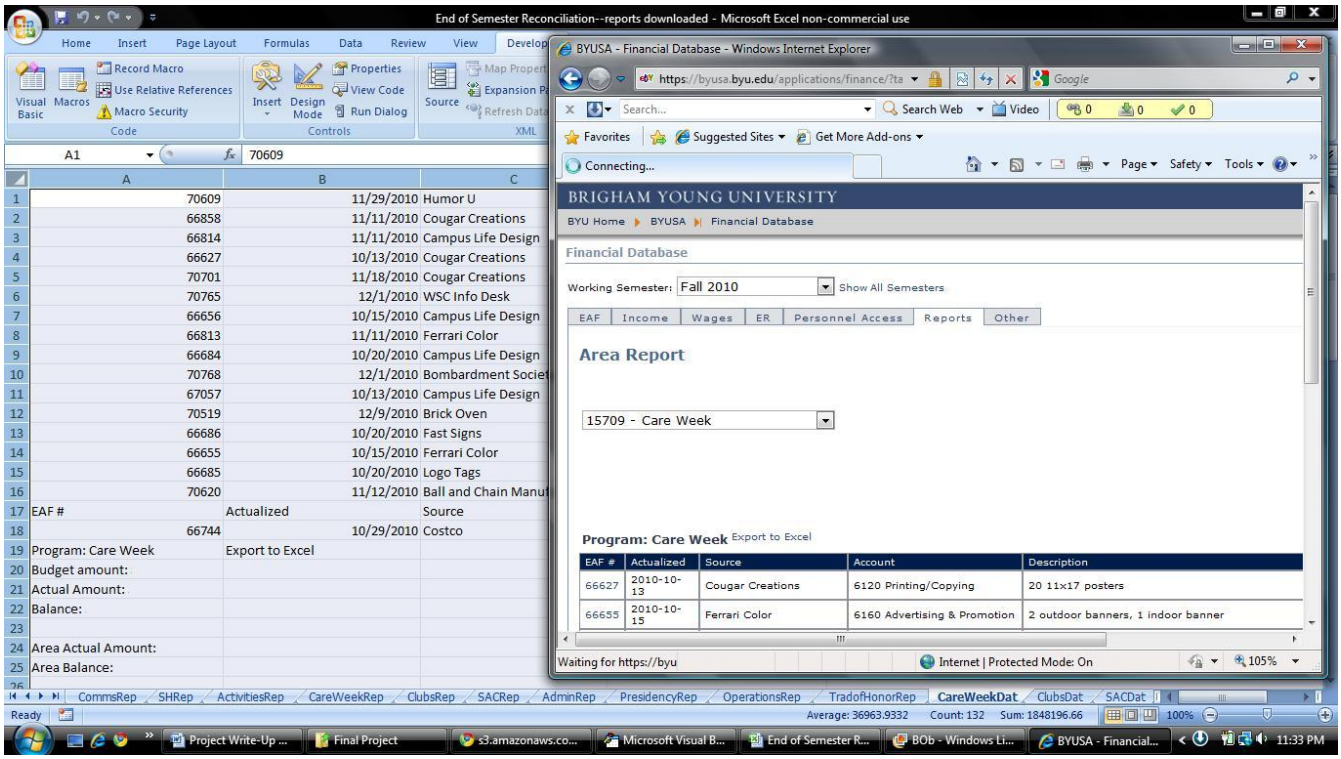
Upon thinking further about this step in the program, I realized that not automating this part of the process would actually be beneficial for the user. This would allow the user to choose exactly the type of data to manipulate. As it turns out, Business Objects has a function in which data can be easily exported into Excel. Also, the information used to query the needed data varies so much between semesters that the user would need to enter the same amount of data whether the process was automated or not. The difference was whether to enter the information into Excel input boxes or enter it into Business Objects input boxes.



This being the case, I chose instead to give the user ability to manipulate data entered and simply import the pages into Excel. A message box pops up, informing the user to import information from Business Objects into Excel, including the naming convention to use. At this point, the program takes the user to the Business Objects website. From there the user can access the data they need and export it into Excel. As the message box instructs, once the Business Objects data has been imported, pressing the OK button will continue the program.

3. Exporting budgeted expenses to Excel

This was a surprisingly easy step. Unlike Business Objects, the online budgeting database utilizes separate urls for different areas within the database. The code for importing these websites into Excel involves simply accessing each area's individual url and copying the page into Excel. Because this requires no user interaction, no message box pops up.



4. Matching reconciled budgeted expenses with actual expenses
After both actual (Business Objects) expenses and budgeted (database) expenses are plotted into Excel, the program matches expenses that have already been reconciled. This prevents possible matching of reconciled expenses with non-reconciled expenses that may have matching prices.

This takes the program to the first of the ten areas to be reconciled, and goes through this process for each other individual area. The program starts with the first actual expense and loops through all budgeted expenses in its matching area. If it finds a match between the actual expense and a budgeted expense that has already been reconciled in the database, it highlights both in yellow and immediately moves on to the next actual expense. If it loops through the entire list without finding a matching reconciled budgeted expense, it moves onto the next actual expense without highlighting anything.

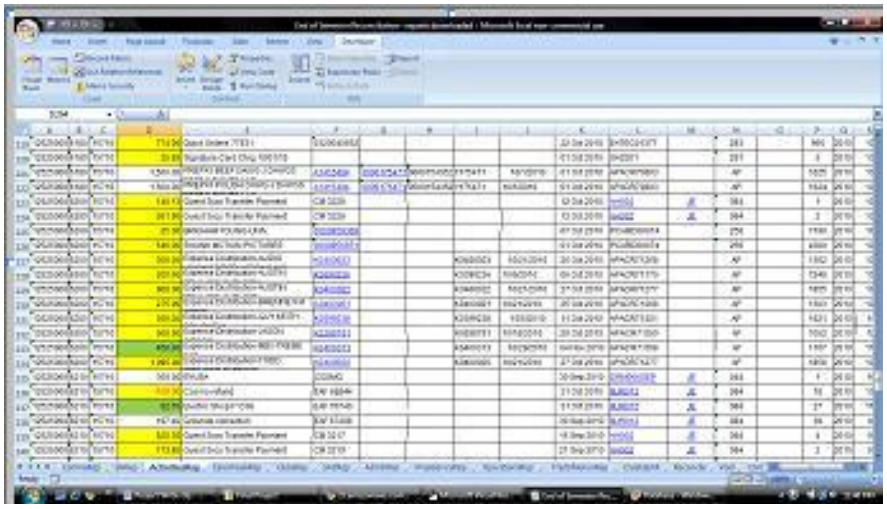
In this manner, each actual expense that came through that has already been reconciled in the database is highlighted and not considered further. This process prevents double-counting of expenses; each actual expense is either matched with only one budgeted expense, or it is skipped.

5. Matching non-reconciled budgeted expenses with actual expenses
This part of the program matches actual expenses with all budgeted expenses that have not yet been reconciled. This is achieved using the same process as before, only this time skipping any expenses that have been already been highlighted in yellow. These expenses are instead highlighted in green. Although these expenses are also matched for reconciling purposes, they have yet to be marked as "reconciled" in the database, and so require further attention. For this reason they are highlighted in green instead of yellow.

6. Copies budgeted expenses that never occurred into a separate spreadsheet
The next step takes all budgeted expenses that did not come through in the database and sorts them into a separate sheet. This allows the user to have a one-stop quick analysis of the data, making it easy to review to determine if the purchase was just yet to be made, or if it should be voided.

This is accomplished by copying each line in the budgeted expenses worksheets that have not been highlighted and pasting them into a single worksheet. The program loops through each expense on each database sheet, copies the ones that have not been highlighted, and pastes them line-by-line into the newly created worksheet, called simply "Void". For ease of analysis, the expenses on the Void worksheet are divided by area.

7. Copies budgeted expenses that need to be reconciled into a separate spreadsheet
The budgeted expenses that are yet to be reconciled are then sorted into another worksheet. This allows the user to quickly go through item by item and reconcile each of these in the database. As above, the program loops through each expense on each database sheet, only this time it copies the expenses highlighted in green into the newly created "Reconcile" worksheet. Again, expenses here are divided by area.



8. Copies unmarked actual expenses into a separate spreadsheet
Finally, the program takes any actual expenses that were not highlighted and sorts them into a separate worksheet. Because these actual expenses did not have matching expenses budgeted in the database, these are set aside for review to determine if they should be accounted for in the database, transferred to a different area, or contested.

This time, the program creates a "CreateEAF" worksheet. It then loops through the actual expenses from each area, copies all the non-highlighted expenses, and pastes them into the CreateEAF worksheet. Similar to the previous examples, the actual expenses are divided by area.