

# Knowledge Base Analytics Generator

Bradford Melliush

## Summary:

The Knowledge Base Analytics Generator is a macro-enabled Microsoft Excel 2010 workbook produced by Bradford Melliush for Brigham Young University Office of Information Technology's Knowledge Management Group under the direction of the Knowledge Manager, Kent Ockey.

The KB Analytics Generator utilizes the Visual Basic for Applications (VBA) programming language to retrieve data from the BYU Knowledge Base and BYU CAIMS (Change Management) to retrieve KB article and product data for use in data analysis. This data is then replicated by the Generator into several different formats specific to individual user needs, and then distributed to c-suite, department, and product level managers for use in cost reduction and product maintenance strategies. This data is also made available to managers of customer-facing organizations such as OIT Service Desk and Knowledge Management itself for use in gauging incident-KB association rates, gaps in documentation, and other potential areas of improvement.

## Value Proposal:

While existing cost-management solutions provide by-service cost information, KB Analytics provide the context to make this information truly relevant to management. By identifying the number of times a given KB article was associated with an incident ticket, managers are able to see not only the cost of a given service, but also gain insights into the issues driving maintenance costs. This in turn allows management to prioritize development and documentation efforts to eliminate the most frequently occurring and/or expensive issues to drive down costs, improve customer satisfaction, and provide greater value to the University. Also, the gathering of aggregate call data provides additional insights into 'pain points' for customers than cost data alone might not be able to portray.

## History – On the Job Training:

During the first stages of this process, I had a firm understanding of Microsoft Excel, but little knowledge of VBA. As Kent approached me with the business problem – he wanted a way to start analyzing the effectiveness of the KB and get a peek at the top problems each month until the University finished its transition to the new KB site and its associated Business Objects software – and I began to catch the vision, though I wasn't sure how I was to accomplish it. At

the time I was considering the Marriott School's Strategy major, and the VBA Information Systems 520 class turned out to be a possible elective for the program, so it seemed to be a fit.

As soon as I began VBA in September of 2011 I started working on the project, trying to find a way to use a recorded webquery to log into the legacy KB, Service Now (BYU's new KB solution), and CAIMS. I hit two major walls at this point: my own inexperience and the difficulty of making Excel interface with OIT's systems. I didn't know how to log into the various websites to retrieve the data I needed, especially when all I had been taught to do was how to grab data from html tables on open sites like Yahoo! Finance. This proved to be even more difficult due to Service Now's .php interface, the lack of any user-accessible metrics in the KB, and the unwieldy of the Excel web interface when attempting to log into CAIMS.

After much deliberation, a great deal of help from my ISYS 520 Professor, Dr. Allen, and a fair amount of elbow-grease, I decided to drop the automation for Service Now, instead going for a manual export of the database that would then be pasted into an "exportdata" sheet in the Generator workbook. I also convinced one of the engineers on the 2<sup>nd</sup> floor of the ITB to build a data dump for CAIMS and the KB to store the Product Manager and KB Name fields that I needed to complete the report, then wrote a program to export the data from each site and integrate it with the user-exported data in "exportdata". It took a lot of work to clean up to code and make the program work reliably and efficiently – even then, it still requires 45 minutes to an hour to run through the 28,000-some incident entries since July.

Once the Generator was up and running, I was able to begin working on the analysis side of the data. The idea was to first build ad-hoc versions of the final product, get it finalized and approved, and then write the code to automate the process thereafter, having the Generator create separate workbooks for each department specific to their analysis needs. I decided to use Pivot Tables for the user interface due to their flexible nature and ability to quickly handle vast amounts of information. Since the workbooks were intended for use by users of all levels of expertise in Excel, I decided to create each workbook with different defaults for each user group, such as setting up filters to show services with the highest incident volume *within a department*. Presetting the Pivot Tables to show data relevant to each group made the interface easy to use for inexperienced users, but also left the workbook open to be changed by more experienced users at will, depending on their preferences.

#### **Things of note in the submission version of this project:**

- For security reasons, the urls pointing to internal OIT databases have been disabled, rendering the "Get KB" button inoperable. For this purpose, the button has been disabled. The code remains in the button, but it has been commented out.

- For the purpose of this project, only one sample pivot table is generated within each report to display the functionality. The final design of the pivot table deliverables is as of yet an ongoing debate at OIT and cannot be demonstrated at this time.
- The “KB Help” button has also be disabled, since the accompanying pdf help file will not be uploaded. To view this functionality, simply create a folder called “Documentation” within the same location as the final project excel file, then within that folder create a new pdf file called help. The button’s code has been commented out, so that also will need to be reverted if a user wishes to test the functionality.

### Functions of the KB Generator:

The project’s functionality is complete contained within the “Knowledge Base Reports” ribbon tab. Within this tab reside the 4 buttons below:



### Description:

- **“Generate KB Reports”**  
This button creates a new sheet called “SamplePivot” and pivot table using a dynamic selection of the data stored in the “exportdata” spreadsheet, and then moves the new sheet to a new workbook titled “Pivot Report <date & time>” within a new “KB Report <date & time>” folder. In practice, this button would repeat the process several times to generate pivot reports tailored to the needs of individual users within OIT (Product Managers, C-Suite, Service Desk, etc.)
- **“Get KB Data”**  
This button asks the user to ensure the workbook is saved locally or in Drop Box (as opposed to SharePoint, for instance), and prompts them to log into the BYU Knowledgebase (KB) and Change Management system (CAIMS). Upon login, the program then downloads data from both databases and inserts data into a table exported from the administration side of it.byu.edu and input manually by the user beforehand. In addition to inserting downloaded data, the program also inserts departmental information stored within the workbook itself and managed manually by the Knowledge Management team. Lastly, the button replaces the “Last Updated” sheet with a new one listing the date and time the data update was completed.

- **Clear KB Data:**  
This button clears the data in “exportdata” in preparation for the user to paste a fresh export from it.byu.edu.
- **KB Help:**  
This button opens a file called help.pdf stored in a folder called documentation residing in the same folder as the project.

For further information on the functionality of the project, please see the source code. The code has been meticulously reviewed and commented on, so each piece should be understandable even to those with remedial understanding on Excel and VBA.