Table of Contents

Execute Summary

There are countless “companies” out there that consist of a single programmer developing a clever app in his or her spare time in his or her pajamas. KickingApp is one of these “companies” that recently released an app on the Windows 8 app store (called *GroupThink*).

The app allows users to share, find, rate, and comment on ideas with other users. It is driven by the data that users submit while they are using the app. Because this is the case, the best way to understand the growth and usage of the application is to look at the growth of the data in the database.

The developer behind GroupThink could look at the number of downloads, but this gives no indication of how users are actually using the application (if they are using it at all). Besides this, glancing at the number of downloads gives no indication of the growth rate. It is simply a snapshot at the current time of how many downloads have occurred in history, even if all of them were more than a year ago, for example.

Additionally, KickingApp plans to release GroupThink for Android devices within the next month. Again, data in the database is the best indicator of growth in usage because this will be an aggregation between the Android app and the Windows 8 app since both use the same database. If the application were released on several platforms, tracking several different developer consoles to see how many users have downloaded the app would be a terribly inefficient (and pointless) way to determine if and how people are using an application.

The Excel program described in this document, Database Growth Tracker, will allow small developers the ability to track the usage of their data-driven applications quickly and easily. No longer will the developer of KickingApp have to check the number of downloads or log into the database through a webbrowser and run SQL statements to see a snapshot of how much data has been collected. The Database Growth Tracker can be run quickly and easily once per day to download and store the size of database tables. Overtime, developers will have a useful dashboard that describes the growth of the usage of their application.

This is meant specifically for small developers that do not have access to more expensive and complex dashboard resources. It is very easy to set up and very quick to run on a daily (or less, as desired) basis. It gives useful information quickly and easily.

Implementation Documentation

A developer that wants to track the growth of their database can set this up very quickly and easily in the Database Growth Tracker Excel worksheet. The two main components of the program are the (1) configuration, which will be set up the first time the developer uses the worksheet, and (2) pulling data.

Configuration

A correct configuration needs to be supplied to create the database connection string. The configuration is set up by clicking the Configure button in the Database Analysis tab of the Excel Ribbon. There are several parameters that need to be configured before a connection to the database can be made, such as the following: Provider, Username, Password, Datasource, InitialCatalog, and finally Tables.

Provider

The provider will depend on what type of database the developer is using (note that as of today, this worksheet has only been tested with SQL Server). If unsure what provider to use, connectionstrings.com can be a great resource. Find the database to which the connection will be made and look for instructions regarding the OLE DB provider to use, as this is the information that the worksheet needs. For SQL Server, SQLOLEDB should work perfectly.

Username & Password

These fields are where a user supplies the username and password that are used to connect to this database. The password is stored in a hidden worksheet in this workbook, so it is not entirely secure.

Datasource

The datasource is the way that the workbook can find your data. If you are connecting to a SQL server somewhere, this will be the address to the SQL Server, such as IdeaMachine.db.10021657.hostedresource.com. It would also be possible to connect to a local MS Access database as well, so this would be the file path to the database file.

Initial Catalog

The Initial Catalog is the name of the database that you want to connect to. This is the final parameter needed to create the connection to the database.

Tables

The tables list is where a user can add or remove the tables that he or she would like to be tracked. These tables must exist in the database and must match the exact name of the table.

Keep Raw Data

The user can check this box if he or she would like to download and keep raw data from the database. This might be the case if he or she wishes to customize the workbook further to add more analysis and dashboard features. If left unchecked, only the totals of each worksheet will be saved in the workbook.

Pulling Data

After a connection has been setup through the configuration userform, the process is very simple. A user simply needs to click the “Pull Data” button in the Database Analysis tab of the Excel ribbon.

This explains how to use the worksheet, but it is a bit more involved to explain everything that happens when the Pull Data button is clicked. The three basic processes are the following: (1) connect to database, (2) download record count for each table in list of tables, (3) add this data to data sheet, (4) create chart for table, and (5) finally create one chart with all tables.

Connect to Database

The program uses the information saved in the Configuration to connect to a database. If something is going wrong here, double check the configuration and internet connection.

Download Record Count

The program downloads all data for each table and saves the number of records. If the user has decided to keep raw data (for further analysis), then this data will be saved in a worksheet (with the name of the table); otherwise, the raw data will be deleted. It might seem like overkill to download all records, and this can be altered if needed. However, because this workbook is meant to be used by small developers on small applications, it should not be a big deal that all records are downloaded.

Add to Aggregated Data Sheet

The aggregated data sheet holds the information of table size for every day that the program is run. After the record count is determined for a table, the aggregated data sheet is examined to determine if a new date needs to be added (if the program is being run today for the first time). If so, then today’s date is added as a new column and the data is added to the correct row underneath it. If today’s date does not need to be added, then we replace the data that was posted here previously.

Create Table Chart

Once the data is added to the aggregated data sheet, all data previously recorded for this table is used to create a new chart. This chart is a visual representation of the data that is easily looked at to see the growth trend. A chart is created for each table and placed in its own worksheet.

Create Aggregated Chart

Finally, after all tables have gone through the processes above, an aggregated chart is created to display all data on a single visual aid. This chart is probably the most useful because all growth trends can be seen at one time.

Discussion of Learning

Building this program was a really good learning experience for me because I learned how to use Excel to connect to one of my own projects. The other projects of the semester have been great in teaching me skills, but this one taught me the usefulness of Excel because I took the opportunity to solve a personal problem I have been having. Now, using this tool, tracking the growth of various applications I work on and plan to work on will be a breeze.

I think the most difficult thing for me was figuring out how to interface the objects in VBA to the data in the worksheets. I had to be very careful to reference the right cells or ranges on the right worksheet so that I could turn objects in memory into hard data and vice versa. Towards the end of the project I realized the usefulness of naming ranges as I put data into the various worksheets.

I also ran into some issues while creating the charts. I used an online resource for the basic chart creation code, but I had to modify it to correctly display the charts as I wanted them to show up. For example, originally the chart was displaying the date on the Y-axis, when in reality it should be displayed on the X-axis. I also had to figure out how to use the Union method to create a range from two mutually exclusive ranges. This was absolutely necessary to create a separate chart for each data table while still showing the correct dates on the X-axis.

The most important thing I learned, however, is greater than any one solution to a specific problem. I learned how to use Excel and VBA to solve one of my own problems. I was getting sick of logging in to the SQL Server database through a web browser to find out how many new users or ideas had been added every day. Not to mention the fact that this was pointless because I never saved the data anyways so a growth trend was nothing more than my vague memory of how many users I had yesterday or last week. Now I have an easy solution to the problem that will keep a history and doesn't require me to open a DBMS on my computer. Next time I have a similar problem, I will consider Excel as a great solution.

And best of all, next time I build an app I can use this exact same workbook to track that database's growth as well!

Assistance

I did not receive assistance from any other persons, but I did use an online example for the code to create a chart. I had to modify this code to work as I wanted it to work, but it helped me a lot. I also used the RibbonWizard from Professor Allen to add two buttons to the Excel ribbon. Thank you!