VBA FINAL PROJECT – 5-Year Plan Upload
Spencer Hafen

EXECUTIVE SUMMARY

In my last job at Intermountain Healthcare, part of my job was assisting people that had problems uploading 5-Year Plan data into the databases Intermountain Healthcare uses. I found myself helping people who had problems uploading data into these databases **a lot.** This was largely because the 5-Year plan was only done once a year, and by the time the next year rolled around most users had forget how to do everything. Even with detailed instructions, problems were inevitable. Year in and year out this was a stressful time for whoever had to deal with these problems. I was struggling to find something useful to do for my Final Project, so I decided I would give my former manager Chris a call to see if he wanted some help automating this 5-Year Plan process. He said he would love whatever help I could give. Chris said ideally he would like a single workbook with simple buttons **on each worksheet** that did the following things:

1. **Connect users to the correct database** (several different databases were being used by different hospitals)
2. **Pull historical data** (previously done manually)
3. **Input & Load data** (previously done manually in different workbooks)
4. **Run 5-Year Plan Calculations from the Workbook** (previously done on the database application itself to avoid problems)
5. **Create a Report to Compare Ratios** (currently there was a report to do this, but it wasn't built into the same workbook)

- **I also created Ribbon Buttons.** I told Chris I could create ribbon buttons at the top of the page to do the same thing, but he said that for now he wanted individual buttons on each page to make it as simple as possible for users. While not part of the final deliverable to Chris, for the purpose of the project I created the ribbon buttons anyways. That is why on the screen shots below you will see buttons on both the worksheets themselves as well as on the ribbon.

- **Instruction Manual:** As part of this project I also created an instruction manual to guide users through the new 5-Year plan process. I debated whether to include the whole instruction manual, but it had a lot of supplementary stuff that doesn't relate to the project so I just included the necessary pieces in my implementation documentation.

IMPLEMENTATION DOCUMENTATION

PREFACE ABOUT "ESSBASE"

Intermountain Healthcare uses a built-in excel tool called Essbase to pull data from different databases into Excel, as well as do the reverse and load data from Excel back into a given database. Essbase has built in functions to do different things in Excel. For example, here are a few Essbase functions that I had to use in my VBA code:

- **Retrieve** - Essbase function "**EssMenuVRetrieve":** This function allows you to "retrieve" data from a database as long as you have the correct inputs. For instance, in the database I was working with, if you wanted the number of pediatric admissions for a hospital you needed 5 inputs; the year (2011), the hospital #(116), the code for pediatric admits (9001PED), the type (V or Value), and the scenario (e.g. UNBASE – Uninflated Base). If you arranged the inputs correctly you can pull whatever data you need from the database you were connected to using the Essbase.
- **Lock and Send** - Essbase function **"EssMenuVLock" & "EssMenuVSend":** This function enables you to "lock" a set of data in a spreadsheet and "send" it or load it to a specific database. More simply this is how you upload data to a database.
- **Run Calculation** - Essbase function "**EssVCalculate"**: This enables you to run a calculation on a database itself using an Excel function. This became useful in my VBA because it allowed users to run calculations on a database without having to use confusing menu items.

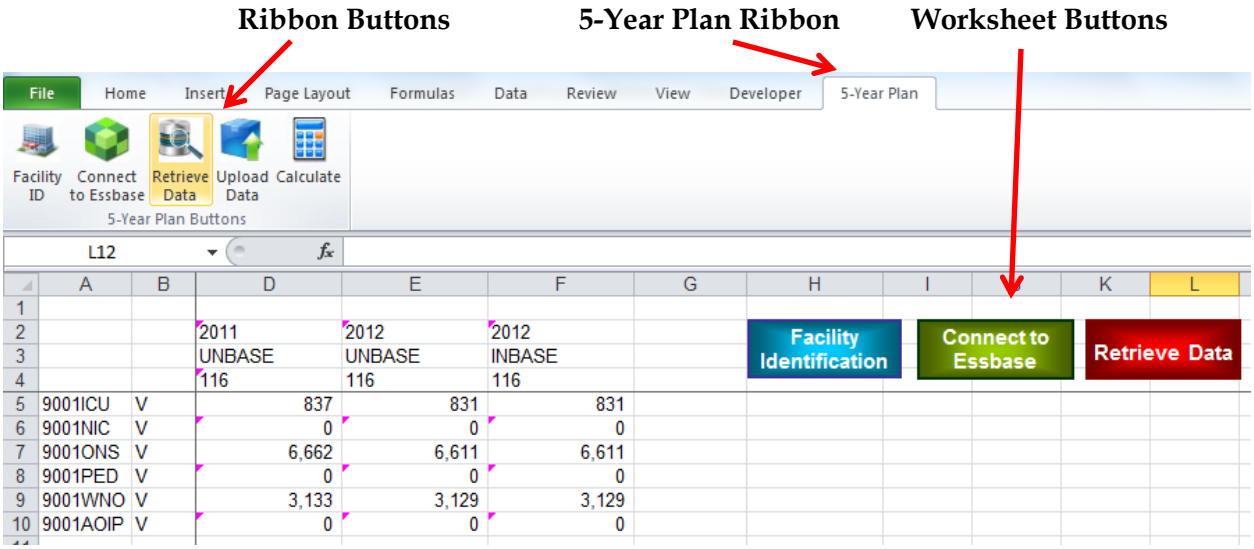
Before starting the project, I already knew how to use these functions **using** the Essbase add-in tool, but I had no idea how to make these functions work in VBA. I will talk later about how figuring out how to incorporate these functions was a one of my biggest challenges.

Important Note: You will not be able to run my program for a two main reasons, but I assure you that my program runs and does what is outlined in the Implementation Section.

1. My VBA utilized many of these Essbase functions which will not work unless you have purchased the Excel add-in tool.
2. Some of these Essbase functions run calculations in databases where the data is actually stored. This is similar to the first point, but if you don't have access to those databases (which you won't) you can't run the calculations.

WHAT MY PROGRAM DOES

As mentioned in my Executive Summary, my former manager Chris wanted me to only use worksheet buttons, but for the purpose of the project I also created ribbon buttons, I will show screen shots and refer to both buttons in each step of the process.



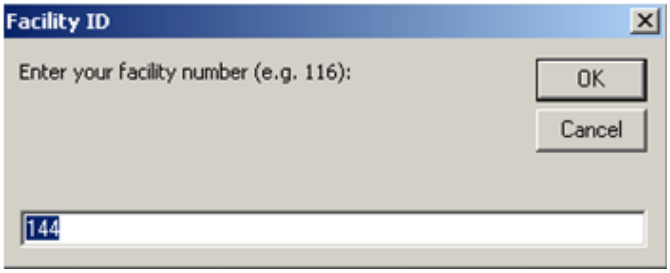
	A	B	D	E	F
2			2011	2012	2012
3			UNBASE	UNBASE	INBASE
4			116	116	116
5	9001ICU	V	837	831	831
6	9001NIC	V	0	0	0
7	9001ONS	V	6,662	6,611	6,611
8	9001PED	V	0	0	0
9	9001WNO	V	3,133	3,129	3,129
10	9001AOIP	V	0	0	0

STEP 1: Facility Identification

Ribbon Button

Worksheet Button

When the user enters the Facility ID and presses OK, this sets the correct Facility number in all the necessary places in the worksheet.

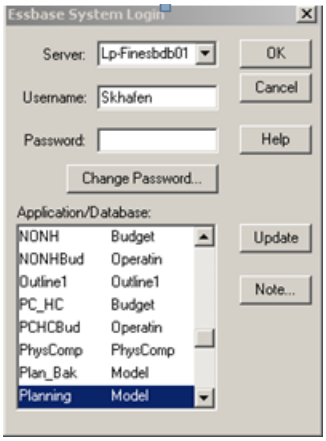


STEP 2: Enter Username and Password

Ribbon Button

Worksheet Button

This is where the user enters their username and password that is necessary to access the databases where they will be pulling data from.

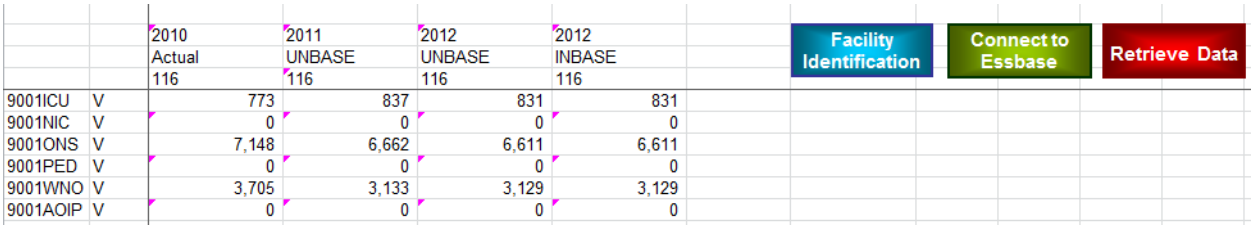


STEP 3: Retrieve Data

Ribbon Button

Worksheet Button

These buttons retrieve the data on the current sheet the user is on. For the purpose of the 5-Year Plan these buttons are used to pull historical data that is used to help people come up with the number they want to upload in the next step.



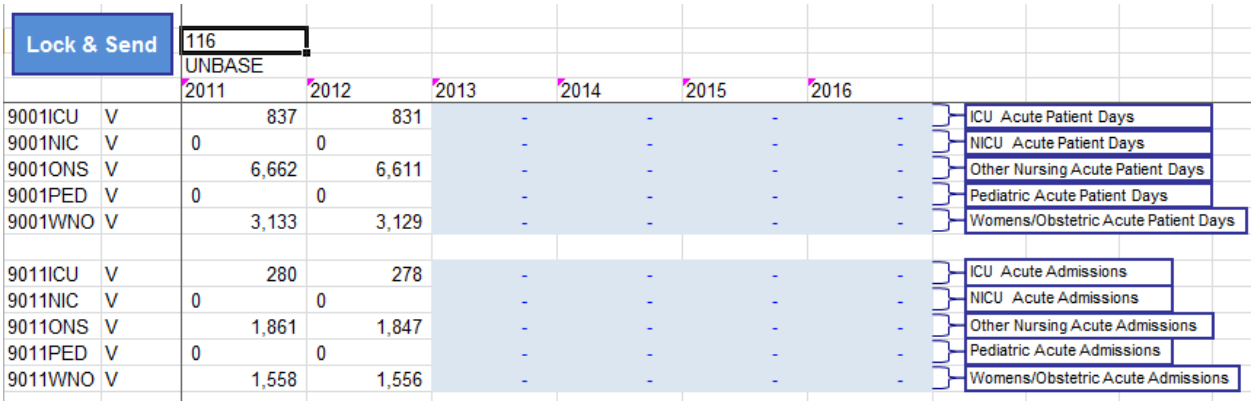
		2010 Actual 116	2011 UNBASE 116	2012 UNBASE 116	2012 INBASE 116
9001ICU	V	773	837	831	831
9001NIC	V	0	0	0	0
9001ONS	V	7,148	6,662	6,611	6,611
9001PED	V	0	0	0	0
9001WNO	V	3,705	3,133	3,129	3,129
9001AOIP	V	0	0	0	0

STEP 4: Upload Data (Lock & Send)

Ribbon Button

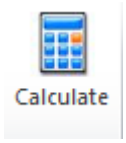
Worksheet Button

All the necessary data for the 5-Year Plan can be uploaded using the Lock & Send or Upload button while on the following worksheets: Stats, Revenue, Expenses, Deductions, Productivity. The data the user wants to uploaded just needs to be input in the blue shaded cells.



		2011	2012	2013	2014	2015	2016
9001ICU	V	837	831	-	-	-	-
9001NIC	V	0	0	-	-	-	-
9001ONS	V	6,662	6,611	-	-	-	-
9001PED	V	0	0	-	-	-	-
9001WNO	V	3,133	3,129	-	-	-	-
9011ICU	V	280	278	-	-	-	-
9011NIC	V	0	0	-	-	-	-
9011ONS	V	1,861	1,847	-	-	-	-
9011PED	V	0	0	-	-	-	-
9011WNO	V	1,558	1,556	-	-	-	-

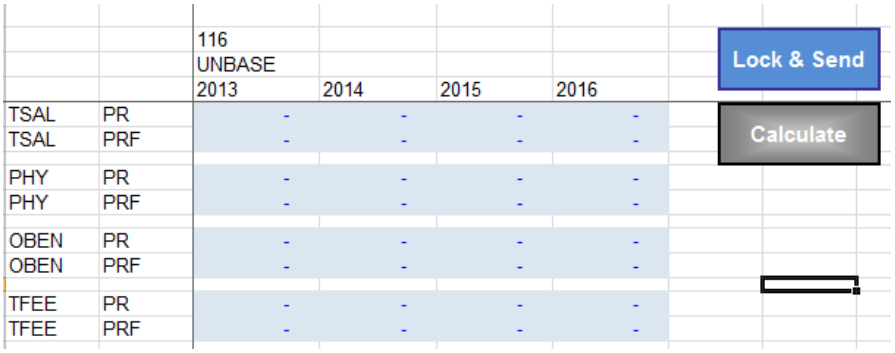
STEP 5: Run 5-Year Plan Calculation

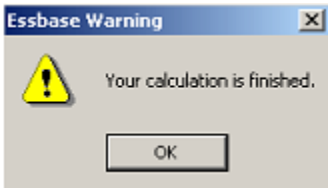


Ribbon Button

Worksheet Button

These buttons allow the user to run the calculations that need to be run on the data that has just been uploaded to the 5-Year Plan database. When the calculation is finished a message will come appear saying the calculation is finished



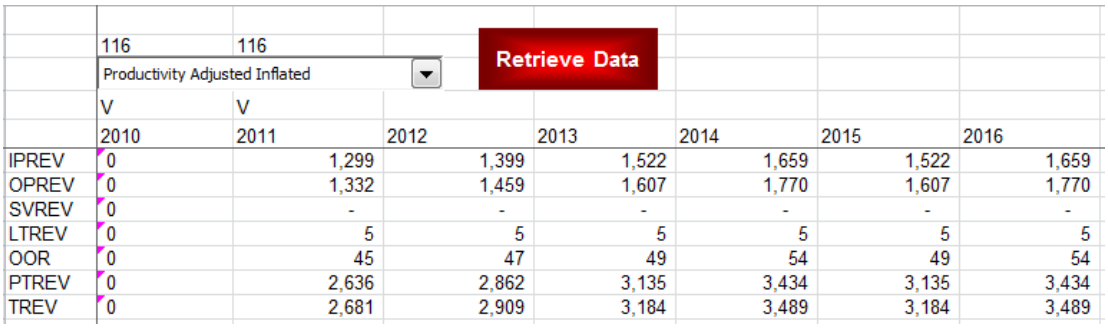


STEP 6: Retrieve Data for Report

Ribbon Button

Worksheet Button

Similar to retrieving data above, the user can use the same buttons on this new sheet to generate the report that shows up on the Final Report tab. The data below is made up data because the actual data is sensitive data.



After retrieving the data, a report is generated based on the Facility ID that was entered at the beginning of the process and the scenario selected in the drop-down list above. I cleared out most of the data on this report as well because this is sensitive data:

Alta View Hospital
Productivity Adjusted Inflated Financial Projections
April 11, 2012
(in thousands)

	2011		2012		2013		2014		2015		2016	
Inpatient Revenue	$ 1	xx.x%	1	xx.x%	2	xx.x%	2	xx.x%	2	xx.x%	2	xx.x%
Outpatient Revenue	1	xx.x%	1	xx.x%	2	xx.x%	2	xx.x%	2	xx.x%	2	xx.x%
Other Patient Revenue	0	x.x%	0	x.x%	0	x.x%	0	x.x%	0	x.x%	0	x.x%
Other Operating Revenue	0	x.x%	0	x.x%	0	x.x%	0	x.x%	0	x.x%	0	x.x%
tal Operating Revenue	3	xxx.x%	3	xxx.x%	3	xxx.x%	3	xxx.x%	3	xxx.x%	3	xxx.x%
Medicare Discount	1	xx.x%	1	xx.x%	1	xx.x%	1	xx.x%	1	xx.x%	1	xx.x%
Medicaid Discount	0	x.x%	0	x.x%	0	x.x%	0	x.x%	0	x.x%	0	x.x%
Commercial Discounts	0	xx.x%	0	xx.x%	0	xx.x%	0	xx.x%	0	xx.x%	0	xx.x%
Charity	0	x.x%	0	x.x%	0	x.x%	0	x.x%	0	x.x%	0	x.x%
Bad Debt	0	x.x%	0	x.x%	0	x.x%	0	x.x%	0	x.x%	0	x.x%
Other Deductions	0	x.x%	0	x.x%	0	x.x%	0	x.x%	0	x.x%	0	x.x%
Total Deductions	1	xx.x%	2	xx.x%	2	xx.x%	2	xx.x%	2	xx.x%	2	xx.x%
Net Revenue	1	xx.x%	1	xx.x%	1	xx.x%	2	xx.x%	1	xx.x%	2	xx.x%
Salaries	0	xx.x%	0	xx.x%	0	xx.x%	0	xx.x%	0	xx.x%	0	xx.x%
Benefits	0	x.x%	0	x.x%	0	x.x%	0	x.x%	0	x.x%	0	x.x%
Total Fees	0	x.x%	0	x.x%	0	x.x%	0	x.x%	0	x.x%	0	x.x%
Total Supplies	0	x.x%	0	x.x%	0	x.x%	0	x.x%	0	x.x%	0	x.x%
Other Purchased Services	0	x.x%	0	x.x%	0	x.x%	0	x.x%	0	x.x%	0	x.x%
All Other Expenses	0	x.x%	0	x.x%	0	x.x%	0	x.x%	0	x.x%	0	x.x%
Interest	0	x.x%	0	x.x%	0	x.x%	0	x.x%	0	x.x%	0	x.x%
Depreciation	0	x.x%	0	x.x%	0	x.x%	0	x.x%	0	x.x%	0	x.x%
Corporate Charge	0	x.x%	0	x.x%	0	x.x%	0	x.x%	0	x.x%	0	x.x%
Total Expenses	1	xx.x%	1	xx.x%	1	xx.x%	1	xx.x%	1	xx.x%	1	xx.x%
Net Operating Income	$ 0	x.x%	0	x.x%	0	x.x%	0	x.x%	0	x.x%	0	x.x%
EBIDA	0	x.x%	0	x.x%	0	x.x%	0	x.x%	0	x.x%	0	x.x%
FTE's	0		0		0		0		0		0	
Key Indicators												
Admissions	0		0		0		0		0		0	
Patient Days	0		0		0		0		0		0	
Surgeries	0		0		0		0		0		0	
ER Visits	0		0		0		0		0		0	
Imaging Procedures	0		0		1		1		1		1	
Cardiovascular Procedures	0		0		0		0		0		0	

DIFFICULTIES ENCOUNTERED

By far, the biggest challenge in doing my project was learning how to incorporate the Essbase functions I talked about above into my VBA code. At first I ran some Google searches and got a little frustrated messing around with help forums. As you might guess, there aren't as many people out there to help you figure out VBA for an add-in tool that not a lot of people use. I finally did what seems obvious to me now; I realized that there are enough people at Intermountain Healthcare that use Essbase that there had to be some VBA code somewhere

within the company that utilized these Essbase functions. I tracked down a workbook that was used in the budgeting process and finally found what I was looking for.

So what I found was a big list of Essbase functions in the workbook I tracked down. Here is a small excerpt of the list:

```
Declare Function EssMenuVRetrieve Lib "ESSEXCLN.XLL" () As Long
Declare Function EssMenuVKeepOnly Lib "ESSEXCLN.XLL" () As Long
Declare Function EssMenuVZoomIn Lib "ESSEXCLN.XLL" () As Long
Declare Function EssMenuVZoomOut Lib "ESSEXCLN.XLL" () As Long
Declare Function EssMenuVPivot Lib "ESSEXCLN.XLL" () As Long
```

Apparently you have to declare all these Essbase functions inside your VBA code. After that you can just call upon them whenever you want. It seems kind of obvious now, but it was a headache for me to figure it out. After that it was easy to use them because they are all pretty self-explanatory and rather easy to incorporate in your VBA once you know they are there.

My second major struggle had to do with figuring out why when other people ran my code on their own computers during testing, it whacked their spreadsheets out. What I finally figured out was that each individual user who installs Essbase on their computer has their own pre-set options. I probably wasted hours trying to figure out why the code was messed up, before I realized it probably wasn't something with the code but something else. I ended up brainstorming with someone else at Intermountain Healthcare who had written some VBA code. He told me there was a good chance it was because they had different "Essbase Options" set up on their computer. So in my code I had to include some changes to some of these Essbase options. Here is an example of some of those options:

```
Sub Suppress()
    Warnings_Off
    x = EssVSetSheetOption(CurrentSheet, 23, False) 'formula fill
    x = EssVSetSheetOption(CurrentSheet, 22, False) 'retain on zooms
    x = EssVSetSheetOption(CurrentSheet, 21, False) 'retain on keep and remove only
    x = EssVSetSheetOption(CurrentSheet, 11, False) 'retain on retrieval
    x = EssVSetSheetOption(CurrentSheet, 6, True)   'suppress #Missing
    x = EssVSetSheetOption(CurrentSheet, 7, True)   'suppress zeros
    Warnings_On
End Sub
```

SUMMARY

In summary I think I did a really good job of accomplishing the goals I had set out to accomplish. The biggest lesson I learned was that next time I should get help **sooner** when I ran into a problem, especially before I start pulling my hair out. I probably wasted too much time trying to figure out some things on my own when I could have tracked down some similar code or found someone with some VBA expertise a to help me out.