

## **VBA Project Report**

### **Executive Summary**

#### **Business description**

I did my project for the company at which I work, Anglepoint. Anglepoint is an IT consulting company, we have partnerships with EMC, Microsoft, IBM, Dolby, and a number of other large and small companies. Our work varies with each of our clients depending on the contract. I work on the team that focuses on our contract with EMC. EMC is one of the world's largest (if not the largest) high-end storage distributors. Our job is to work with the account teams to help them to know how much of what they have sold to who, whether it is currently being supported by them, and whether or not they are up to licensing terms according to the clients usage. EMC has made many acquisitions since its inception and then adding those databases to their already old legacy system can make for quite the process in figuring out what EMC's client currently has and is licensed to use. Once all the data is found, everything is verified against EMC sales orders. Our processes have improved, but even so, it can be quite the mess to get everything strait and verified so that we are providing accurate data.

#### **Project Overview**

We deal with many different "model families" or lines of products and each has its own process to "map" or organize the data and then to verify what we have organized, and here we are dealing with DataDomains. DataDomain systems are for very efficient data backup utilizing specialized target-based data-deduplication solutions. This project automates the pulling of dial back data for DataDomain serial numbers from a system called Elysium. We use Elysium and EMC sales orders in tandem to verify data we get from another database that holds general installation info called Installed Base. Previously the process to pull the DataDomain information has been very slow because doing each serial number individually takes a lot of time and Elysium itself takes a while to load the proper information because it

has to pull a lot of information from the previous dial-back (automatic report from the machine as to status of enclosures, drives, software, and other components). In automating this process, it will still take time to complete (though not nearly as much), the person can go and work on something else (we often have an open computer or two from people that are out of the office to run programs/macros like this), and the data will be organized in a concise and easy to read manner.

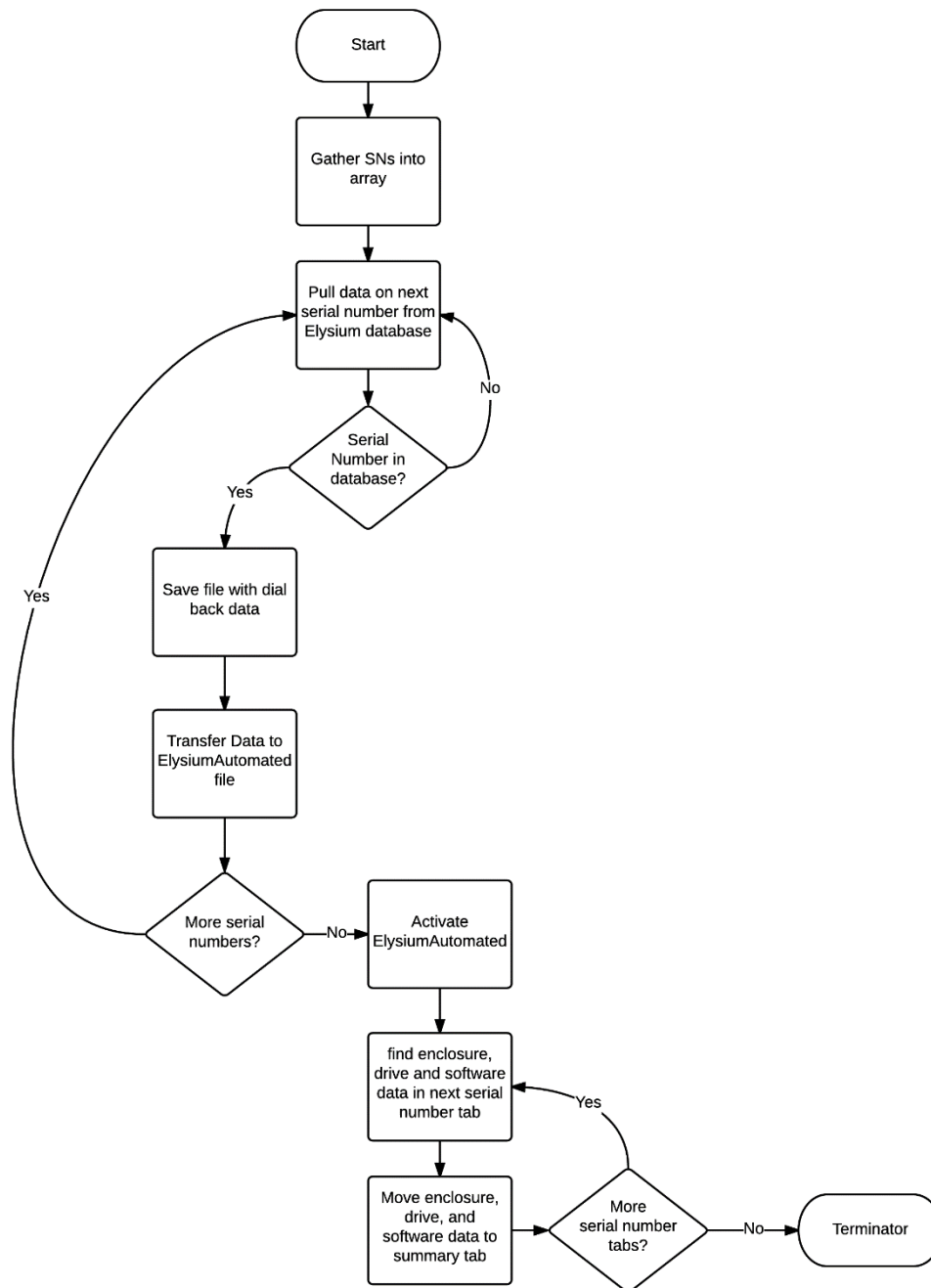
## **Implementation Documentation**

### **Role of project in the business**

This automation can be used in any project that contains DataDomain arrays. Many of the projects that I have done have DataDomain arrays, I know that I have also had a few projects where there were many DataDomain arrays and it took much too long to gather all of the information I need on them in order to properly be able to start to organize and verify the data that I had gathered. As DataDomain was an acquisition, many times we will come across sales orders that EMC does not have in their system and in this case it is very important that we have access to Elysium's information because it is our only way to verify the data from Installed Base (which has proven to be unreliable). The database accessed by the macro is behind a VPN, and access to which I cannot give out, so this project will not function as normal if the VPN is not activated.

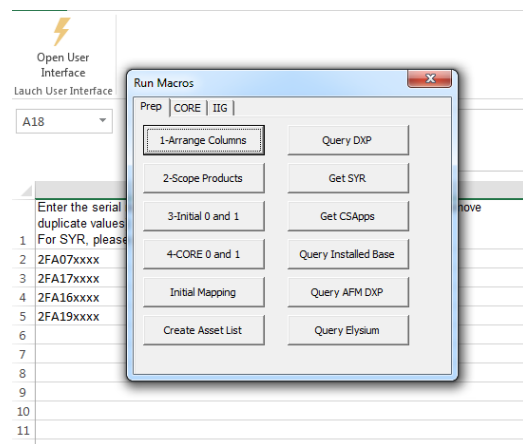
### **How I accomplished what I did**

The following is a high-level flow chart of the processes of the project. Of course it could be much more detailed as to how it does what it does, but this will help you to understand generally how the macro accomplishes what it does.



## Components

The program functions based off of a number of initial used inputs and then goes on from there (I have not created a list of the components although each is described in the following section along with its use and purpose). The user enters all of the DataDomain serial numbers onto a tab called “Serial Numbers for Macros” (this tab will exist in the general template that all download at the start of the project – I have merely built onto a previously existing template). A special tab has been created called “Macros” and when clicked the user can select the “Open User Interface” button, which will allow them to run whichever macros they need on the data they have (I have left all of the buttons so you can see the various macros that have been created, though only “Elysium Automated” that I have created is active, as the other macros are proprietary and I cannot share). When the user starts the macro they are prompted to select a save folder where all files will be kept, this is used as a kind of “home” directory for the duration of the macro. They have the option to cancel or continue as they need to. Once the proper “home” location is selected, the macro will create an array of all of the serial numbers that have been entered into the “Serial numbers for Macros” tab so that continual reference does not need to be made and it can just use the array instead.



Using Professor Allen’s agent class, each array is searched for from the Elysium search page and the most recent dial back is selected. If the serial number is not found in the database (which does occur) the a flag is set to “No data found” and a page is still created in ElysiumAutomated (the summary workbook) so that the user knows that this serial number was not just skipped or overlooked, but that it did not return any data. For serial numbers that do return data, the link will take the agent to a new

[illegible]

page that contains the text file with all of the information for that DataDomain's most recent dial back, and it is a lot of information. The agent then uses a modified importPage function (which will be discussed more in depth later) to


create a CSV of all of the information in the home directory and then that data is then copied to a tab by the name of the serial number in the ElysiumAutomated file. This is done as a loop for each of the serial numbers so that there will be as many tabs as there are serial numbers entered by the user in the beginning, and the data for each in its respective tab.

Once all of the data has been retrieved, a sub-procedure will create the summary tab which is what will be used by the employee. The sub-procedure will go to the first tab and look for certain key words that specify the beginning and ending of the data that we need for verification. The imported page can provide over 25,000 lines of data for a single array (serial number) and we only need maybe 100 of them for our verification processes. The macro will go and find this important information and paste it in the first open space on the Summary tab, once all of the data for a serial number has been pasted into the Summary tab (in column B) the serial number will be pasted into column A for an easy reference. This is done

4	
5	<script type="text/javascript" src="/Elysium/jsp/plugins/jquery.cookie.js"></script>
6	<script type="text/javascript" src="/Elysium/jsp/plugins/modernizr.mq.js"></script>
7	<script type="text/javascript" src="/Elysium/jsp/plugins/jquery.joyride-2.1.js"></script>
8	
9	

◀ ▶	Summary	2FA16xxxx	2FA07xxxx	2FA17xxxx	2FA19xxxx	+
-----	---------	-----------	-----------	-----------	-----------	---

workbook. Once all of



the necessary information has been transferred to the Summary tab, it is then copied and pasted into the Installed Base file (the template file used for the data organization) as the first tab for an easy reference during the organization and verification processes.

## Outcomes/outputs

The data in the Summary tab that we use for verification purposes is information about the enclosures (shelves that hold the drives and have their own serial numbers we need to make sure that we have), the drives themselves (which drives go to which enclosures, drive size etc.), and the currently running software (names and licenses). Using the sales order and this data, we can see what they originally purchased and what is currently installed and running and create a report according to whether they are in line or that discrepancies have been found.

The ElysiumAutomated file is created in the process of the macro, although it can grow to be very large with the massive amount of data that it stores, so if the user desires, they can delete it as the summary tab has been transferred to the Installed base file for easy access.

A		B						
1	Serial Numb...	Information						
2	2FA16xxxx	Enclosure Show Summary						
3	2FA16xxxx							
4	2FA16xxxx	Enclosure	Model No.	Serial No.	State	OEM Name	OEM Value	Capacity
5	2FA16xxxx							
6	2FA16xxxx	1	DD990	2FA1726029	Online	4 Slots		
7	2FA16xxxx	2	ES30	APM00122305140	Online	15 Slots		
8	2FA16xxxx	3	ES30	APM00122305147	Online	15 Slots		
9	2FA16xxxx	4	ES30	APM00122305142	Online	15 Slots		
10	2FA16xxxx	5	ES30	APM00122305145	Online	15 Slots		
11	2FA16xxxx	6	ES30	APM00122305144	Online	15 Slots		
12	2FA16xxxx							
13	2FA16xxxx	6 enclosures present.						
14	2FA16xxxx	Disk Show Hardware						
15	2FA16xxxx							
16	2FA16xxxx	Disk	Slot	Manufacturer/Model	Firmware	Serial No.	Capacity	Type
17	2FA16xxxx	(enc/disk)						
18	2FA16xxxx							
19	2FA16xxxx	1.1	1	HITACHI HUC10606_CLAR600	C330	PVHZ0YYB	558.79 GiB	SAS
20	2FA16xxxx	1.2	2	HITACHI HUC10606_CLAR600	C330	PVHSYUSB	558.79 GiB	SAS
21	2FA16xxxx	1.3	3	HITACHI HUC10606_CLAR600	C330	PVJ07ASB	558.79 GiB	SAS
22	2FA16xxxx	1.4	4	HITACHI HUC10606_CLAR600	C330	PVJ06T3B	558.79 GiB	SAS
23	2FA16xxxx	2.1	0	HUA723020ALA640	MK70A870	YFHKGG2G	1.81 TiB	SATA

## Learning and conceptual difficulties

I have had some previous experience with VBA, and then I learned a lot through this class with all of the homework and projects, but this was my first in depth and professional project that I have done. Previous it has always been either school related or personal projects so I had not paid so much attention to formatting details and comments in the code. These were things that kind of caught me off guard as to their importance in the project development process.

## Reworking the agent class

In order for the macro to work, I had to go through and recode some of the agent class' sub-procedures. This was a slow and detailed process. I had an idea of what it did, but I had to figure out how it did it and then how to change it so that it would do what I wanted it to do. Basically, the importPage procedure would download an HTML file use its data and kill it before the user could do anything with it. I changed it so that the procedure would download the entire page's data to a CSV file that was placed in a file of the user's selection (via an extra parameter that I added to be passed in) and then held on to so that the macro could get the necessary information from it and then delete the file so that the user doesn't have to deal with it later. The changed agent procedures can be found in the class modules of the Installed Base file containing the project code.

### **Integration into the existing template**

As the project is to be integrated into an already existing template, I had to figure out how to make sure that it didn't mess with any of the other macros and also I wanted it to make use of some of the pre-existing features such as the allfuntions class module (it mostly contains small functions such as for finding the last row of a column, the full name of a workbook given a keyword, and so on). I also wanted to make sure that the formatting would fit in with the pre-existing macros and that it could be integrated into the user interface for easy access. This was all done for business continuity, it is much easier for a user to have all the macros that they need in a single place than creating a new ribbon section for each one.

### **Controlling IE**

Controlling internet explorer with the agent class did prove to be difficult at certain points. Some things were relatively easy such as finding certain fields and entering information and clicking buttons, but once I needed to obtain certain information or manage a page in a certain way, things

became much more difficult. Thus, I had to comb through the agent procedures and change it according to what I needed done (as discussed previously).

### **Finding valuable information in a very large amount of dial back data**

Each dial back returns a massive amount of data, as it should. It is basically a health report back to the main system as to how the array is doing, if anything is wrong it will be reported there. A single report can provide over 25,000 lines of data about the array while we only need about 100 (give or take depending on the size) that contain information on the enclosures, drives, and software. So figuring out a way to grab just that information and leaving the rest was a bit of a puzzle because I had to make sure that the rules I found would be true for all successful dial backs. I was able to find certain key words that would identify the start and end rows of each set of data, and then I would use that information to make sure that I would grab only the useful information in our verification processes.

### **Future iterations**

This project itself may be completed, but this is only its first iteration. I will be working at Anglepoint over the summer so I am sure that I will maintain the code, but not only that I have some ideas to improve it and I am sure that I will receive some good suggestions for implementation once this program gets pushed live to the company.

### **Add labels**

One idea that I have is to add labels to the different sets of data. This will not be incredibly hard, it will just take a little extra work. I want to make it so that the user can filter on the serial number and then if they desire they can filter and just look at either the enclosure, drive, or software data. This is just another way to help streamline the process as the information that they need directly will be immediately available through applied filters.



### **Separate pull for shelf serial numbers**

Another idea that I have is to automatically have the macro create another data pull from the Install Base database when it finds enclosure serial numbers that we do not already have in our data. This would be a more intensive addition, requiring parsing through what we have already separated out, though it could be helpful because sometimes when we run the SQL queries for the initial data, it does not return quite everything and this could help to obtain some of that missing data. Also, each shelf holds only one type of drive, so the summary could be further simplified by getting rid of the drive section and placing the drive size next to each shelf. Then each serial number would be composed of the shelves and drive sizes followed by the installed software making it very easy to use and read.

### **Assistance**

As far as assistance with coding goes I did not have much help. I did consult with a few people at work who had previously written macros about methodology for figuring something out and working through bugs, but their expertise did not lie where most of my questions were- controlling Internet Explorer via VBA. I also had a few minute conversation with Professor Allen in respect to the importPage sub-procedure, we were cut off prematurely, but it helped me to understand a little better how the agent class dealt with importing data from a web page.

### **Guidance on formatting**

I did however receive guidance as far as formatting was concerned. I made sure to keep the Senior Consultant (in charge of this branch of development) in the loop as to where I was on my project and how everything was looking. We talked about it and he felt that it would be better for my formatting to fit along with what had already been produced by the company. This wasn't a limiting factor on what was accomplished and how it was done, merely formatting guidance as to how to gather

the necessary information, start the macro, and create the outputs to be used in the data organization processes.

## **Conclusion**

This project has provided me with a great learning opportunity and I know that I will continually be able to apply the skill I have learned as I maintain and make additions to this project and even as I move on to other projects. I know that I will have these opportunities because I am going to continue to work here at Anglepoint for at least the next year. I am excited to see what else I will be able to accomplish by using and building upon the skills that I have gained in this class.