Office 365 Administration Tool Bag

by

Tyler Sparr

Submitted to
the Faculty of the School of Information Technology
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Technology

© Copyright Tyler Sparr

The author grants to the School of Information Technology permission
to reproduce and distribute copies of this document in whole or in part.

Tyler Sparr

Mark Stockman, Faculty Advisor

04/29/15

04/29/15

University of Cincinnati
College of
Education, Criminal Justice, and Human Services

April 2015
Table of Contents

Section                                                                 Page

Abstract .................................................................................................................. 1
Introduction ............................................................................................................ 2
Project Concept ..................................................................................................... 3
Methodology .......................................................................................................... 4
Design Objectives .................................................................................................. 4
User Profile .......................................................................................................... 6
System Design ....................................................................................................... 7
Use Case Diagram .................................................................................................. 8
Security .................................................................................................................. 8
Timeline ............................................................................................................... 9
Competition .......................................................................................................... 10
Budget .................................................................................................................. 11
Problems and Issues ............................................................................................. 11
Future Recommendations ....................................................................................... 12
Conclusion ............................................................................................................ 13
References ............................................................................................................ 14
List of Illustrations

Figure 1 – O365 User Creation Form ................................................................. 3
Figure 2 – O365 Admin Center ........................................................................ 4
Figure 3 – User Creation Tool .......................................................................... 5
Table 1 – User Profile ..................................................................................... 6
Figure 4 – System Design ................................................................................ 7
Figure 5 – Use Case Diagram .......................................................................... 8
Figure 6 – Gantt Chart ..................................................................................... 9
Figure 7 – O365 Command Geographic Reporting ......................................... 10
Table 2 – Budget ............................................................................................ 11
Abstract

Office 365 is currently one of the leading providers for hosted email with nearly 60 million subscribers; however, most advanced administration tasks have to be done with PowerShell or take much longer through the web interface. These tools aim to streamline and simplify Office 365 administration, as well as empower non-technical staff such as HR and Finance. Using PowerShell as the backend, these tool will provide a convenient GUI interface in which to perform user management, create new users, and other various administrative duties with special emphasis on bulk operations. This will allow businesses to save time and money, while also allowing admins to delegate simple tasks.
Introduction

Office 365, as with many cloud-based solutions, is squarely aimed at those organizations that either do not have the in-house technical expertise to maintain their own email servers or the time. By allowing these organizations to offload some of the administration tasks to Microsoft, it can lead to huge cost-savings. There are still many different tasks that will need to be done by the administrator and not all of them are exactly quick tasks to complete, as well as many that require extensive PowerShell knowledge.

The solution to this problem is a few custom-built GUI tools that push the PowerShell to the background allowing the focus to be on the task rather than syntax or obscure commands. What this means is that not only can administrators perform tasks much faster and in a repeatable manner, but also that other departments who may not be very technical at all can participate in small administrative tasks. This takes pressure off the IT department and allows them to focus on other projects. The less time spent performing routine tasks and the more repeatable the process, the better the time and cost-savings for the business.

The remainder of this final report outlines in detail how the project was completed. The report includes the following sections: project concept, methodology, budget, timeline, problems encountered, and future recommendations.
Project Concept

While most admins know the value of scripting and automation, there is a large percentage, particularly in a small to medium business (SMB), which does not have the time to set it up in their environment. Admins in an SMB tend to utilize cloud services more, but are often reliant on the web interfaces for these services (Cohen). Currently, those web interfaces are slow and inefficient. Performing multiple tasks can be a matter of hours if not longer. My hope is that I have made some very valuable tools for these admins to allow them cut this time to seconds or minutes which will in turn let them focus in on other problems within their own business.

As an example of the inefficiencies of administering Office 365, please see Figure 1 below. The form must be filled out individually for each user account and after filling it out, there is a 15-30 second delay before the next user can be started processing. This makes the whole process to make one account take nearly as much time as thirty accounts can be made using my tool.

![Figure 1 - 0365 User Creation Form](image-url)
Methodology

Using PowerShell as the backend and its ability to call .Net assemblies, I created a fully native GUI tool for Office 365 administration. All coding is being done using the built in integrated scripting environment. SAPIEN PowerShell Studio 2015 is handling all coding for the GUI. This program helps to follow standard methods of coding and guidelines more accurately. All aspects of the code are native to PowerShell with nothing extra needed.

Design Objectives

The tools must be easy and intuitive to use. A big goal was to allow a layman to open the tool and perform administrative tasks with very little training. With ease of use of the tools being such a priority, precautions must be taken against errors being made. It will be nearly impossible to perform any sort of sanity checking, so logging is a big concern. Logs must be accurate and be able to be easily reviewed.

When you first log in to the Office 365 administrative interface on the web, you are greeted with an absurd amount of options as well as five separate other administrative functions.

Figure 2 – O365 Admin Center
Someone who has never had to administer the service before may not know where to begin or what the difference between the Office 365 admin center and the Exchange admin center is. That’s where the simple design of my tool is really effective as seen below:

![Figure 3 – User Creation Tool](image-url)
# User Profile

## User Profile Form

<table>
<thead>
<tr>
<th>Application:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>User needs to make changes to Office 365 users and does not wish to or does not know how to use PowerShell.</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Potential Users</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.) Office 365 Administrators</td>
</tr>
<tr>
<td>2.) Jr. Admins</td>
</tr>
<tr>
<td>3.) HR</td>
</tr>
<tr>
<td>4.) Finance</td>
</tr>
<tr>
<td>5.) Management Staff</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Software and Interface Experience:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Users should already have a general idea of how Office 365 works and what various fields are referring to. No working knowledge of PowerShell is needed.</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Experience with Similar Applications:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>User has basic computer skills and is able to fill out online forms accurately. Previous Excel and/or Access experience would be useful.</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Task Experience:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>The users may have no previous experience completing these tasks, or they may have a lot of experience. The goal is to streamline and speed up the performance of these tasks, as well as empower those users who aren’t as technically inclined. A high degree of accuracy at filling out forms is expected.</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Frequency of Use:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Depending on the environment, this tool may be used as often as daily. Expected usage pattern is once or twice a week.</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Key Interface Design Requirements that the Profile Suggests:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.) <strong>Interface must be simple and easy to navigate</strong></td>
</tr>
<tr>
<td>2.) <strong>Buttons and forms must succinctly convey what they will do</strong></td>
</tr>
<tr>
<td>3.) <strong>Separation of elements (new and existing users)</strong></td>
</tr>
<tr>
<td>4.) <strong>Real-Time updates</strong></td>
</tr>
<tr>
<td>5.) <strong>Fast</strong></td>
</tr>
</tbody>
</table>

*Table 1 - User Profile*
System Design

Each administrative computer (Windows only) connects over an HTTPS tunnel to Microsoft’s servers which allows them to run remote PowerShell commands against Microsoft’s servers. The HTTPS tunnel ensures that the commands are secure and, because it goes over the Internet, administration can be done from any Windows computer with Internet access. Logs are stored locally on the administrative machines should any investigation be needed. Please see illustration below:

Figure 4 - System Design
Overall, the project is as secure as it can be while still offering the ability to store passwords. By using the Windows Data API model to store encrypted passwords on the device, it ensures that only the same user from the same computer can use the stored credentials. This does not protect from a shared user account on the administrative workstation, but is as secure as it can be while maintaining functionality. Also, all traffic is sent over an encrypted HTTPS tunnel making the weakest point in the chain Microsoft’s certificates. Presuming these aren’t compromised and there is no SSL inspection proxy on the local network, this data cannot be sniffed.
Timeline

Figure 6 seen below is the final timeline for the project. The project spanned from August 2014 through April 2015 and was completed as part of the graduation requirements for the Bachelors of Science in Information Technology. All steps were complete as of April 30, 2015.
**Competition**

Office 365 is a newer service, so aren’t many products to fill this niche. The largest is 365 Command which includes many great features such as management, reporting, and support for everything you can already do in the web portal. Its main drawback is that until recently, they did not offer a free trial. They have since added a trial option, but I was unable to contact the sales team in time to demo this product. As such, all information was taken from their website (365 Command).

During the duration of the project, 365 Command has updated their pricing structure to be very competitive. Currently it is $20 a month for up to 50 users and an additional $0.30 per user. Special pricing is available for over 2,000 users. When the original evaluation was done, these prices were much higher. They also offer innovative reporting options as seen below in Figure 7. Now it’s hard to recommend my solution over theirs unless the evaluating company is incredibly budget conscious.

![Figure 7 – O365 Command Geographical Reporting](image-url)
Budget

The final budget is seen below in Table 2. The actual cost of this project does not include the coding hours, but they are included to show what a comparable project would cost. The main budgeted items were the license for PowerShell Studio 2015 and the monthly costs for the Office 365 service.

<table>
<thead>
<tr>
<th>Item</th>
<th>Quantity</th>
<th>Cost per Unit</th>
<th>Total Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coding Hours</td>
<td>150</td>
<td>$65</td>
<td>$9,750</td>
</tr>
<tr>
<td>PowerShell Studio 2015</td>
<td>1</td>
<td>$389</td>
<td>$389</td>
</tr>
<tr>
<td>Office 365 Business Essentials</td>
<td>7</td>
<td>$6.96</td>
<td>$48.72</td>
</tr>
<tr>
<td><strong>Total:</strong></td>
<td></td>
<td></td>
<td><strong>$10,187.72</strong></td>
</tr>
</tbody>
</table>

Table 2 - Budget

Problems and Issues

As with any large project, problems are to be expected and this project was no different. The problem that caused the most trouble was coding a multi-tabbed GUI. After much struggling, the choice was made to abandon this idea. Instead, multiple tools were created. This allowed for each tool to be more specialized without sacrificing functionality.

Another large problem was coding the GUI itself. Rather than wasting time working with a new language, a choice was made to purchase a license for SAPIEN PowerShell Studio 2015. This allowed for more focus to be put on the functionality rather than the coding.
**Future Recommendations**

There were various choices made during this project that when looking back may not have been the best choices. If the project were to start over, there’s a good chance I would not have chosen pure PowerShell as the code base. I chose it because it’s what I know, but that doesn’t mean it was the best choice. I probably would have also chosen to not build a full GUI and instead just build prompts into PowerShell scripts. This would allow less technical workers to follow along, while also allowing full IT staff to bypass the prompts via switches. If I had more time to perfect absolutely everything, I would probably build much more functionality into the tool. Right now, it really just focuses on the most common tasks, but there are others that I’ve had to perform a regular basis that are very specific to my company. I was intending to build a generalist tool, so I did not include those functions.

As far as improvements go, I’ve had some feedback that the tools look too similar. I’m sure this would be an easy change to make if I knew more .NET, but as I’m just using PowerShell studio to build it, I’m kind of limited. The themes are based on what typical Windows executable files look like. My plans are to continue to build and maintain this tool as a pet project and continue to use it in my business environment. I have no plans to market it or release it to the community as a whole because I don’t wish to actively maintain it. Also, with 365 Command’s recent pricing structure change, it’s hard to actively recommend my product to any but the most budget minded of businesses.
**Conclusion**

The biggest lesson learned throughout this project is that sometimes just because you have a hammer, doesn’t mean everything is a nail. This means that not everything needs to use the same processes and tools that you already know. It’s ok to look out there for other options. I coded in PowerShell because I know PowerShell. It’s definitely enhanced my knowledge of the language itself though and I’ve gained a deeper understanding of the Office 365 services, both pros and cons.
References

