Daniel Christianson daniel.s.christianson@gmail.com

Senior IT Support Technician • Aspiring Engineer • (612) 702-0265 • Burnsville/Minneapolis, MN

SUMMARY

IT Support Technician of 7 years excited to transition into a more challenging role. My education background was in graphics and media arts, but I found I enjoy the tech industry just as much, if not more. I've built up a wide breadth of skills over the years and I enjoy learning new ways to solve problems with programming.

TECHNICAL SKILLS

Software: Webstorm, Visual Studio, IIS, Notepad++, Sketch, Photoshop, Illustrator, After Effects, Active Directory, Azure, MEM, Intune, WDS MDT, VPN/RSA/MFA, Citrix, Symantec, MS Office, Acronis, Configuration Manager (SCCM)
Operating Systems: Windows 10, Windows Server 2012 R2, Mac OS, iOS, Android, some Linux
Databases: MySQL, Microsoft Access, Fox Pro
Programming/Scripting: JavaScript, React, Phaser, Git, Python, PowerShell, VBA, Fox Pro
GitHub: https://github.com/daniel-christianson **LinkedIn:** https://www.linkedin.com/in/daniel-s-christianson/
Personal Projects: https://danielchristianson.art/

EDUCATION

Normandale Community College: Associate of Fine Arts Degree
Minneapolis College of Art and Design: Animation major
Art Institute International MN: Media Arts & Animation major
Code School: 13 Online Courses *(before they merged with Pluralsight https://www.pluralsight.com/codeschool)*
Pluralsight: 10 Online Courses *(https://www.pluralsight.com/)*
Course Topics: Azure AD, Azure Intune, JavaScript, ES6, React, Angular, Node.js, Express.js, Git, Active Directory, PowerShell, Python, C#, HTML, CSS, Web Design

WORK EXPERIENCE

Accenture SUS / Amadeus / Navitaire — July 2017 – Present (3 years)

Corporate Office – 333 South 7th St. Suite 1700 Minneapolis, MN — *IT Support Level 3*

Responsibilities

- **Helpdesk:** Service Now helpdesk environment with strict SLAs, included on-call rotation.
- **After Hours Support:** on-call rotation, calls were rare.
- **Laptop/Asset Support:** imaging SME, procurement, deployment, inventory, troubleshoot, repair, decommission.
- **Active Directory:** new hires, roll-offs, add/remove AD groups, add/remove computers, reset passwords.
- **Configuration Manager (SCCM):** compliance reports, compliance emails, security updates.
- **RSA/VPN/MFA/Symantec Support:** assign tokens, troubleshoot connection issues, reset PINs.
- **Office Network Printer Support:** 3 local printers/fax/scanners, 1 local multimedia printer.
- **Audio/Video Conference Room Support:** 10 local conference rooms.

Notable Contributions

Laptop Compliance Email Tool

Technology Used: PowerShell, VBA Excel

Summary: Created a tool that analyzed the output of a compliance report to send emails to relevant users. This saved multiple hours of work per week for IS team members who previously had to go through the report manually and send emails one at a time.

Details:

- The first part of the tool was written in VBA to iterate through the compliance report, which was in Excel spreadsheet format. It located users whose laptops were being flagged for security compliance issues and saved their details to a CSV file depending on their location, issue, etc.
- The second part of the tool was written in PowerShell to evoke Microsoft Outlook through the Windows COM object. The tool read the details in the CSV files, kept track of whether the user had been notified previously, then sent an email with instructions for the user to take action. If the user was notified over 3 times a manager would be copied.

Active Directory Automation

Technology Used: PowerShell

Summary: Worked on various tools/scripts to automate common AD tasks.

Examples:

- **"Remove AD User Tool."** Takes a list of usernames from a CSV file and checks to see if they belong to a specified AD group. If so, the user is removed from the group and a log is created.
- **"Password Generator Script."** Generates a password that excludes characters like: O, 0, i, l, 1.
- **"New AD User Tool."** A customized user creation tool with a GUI.
- **"Delete AD Computer Tool."** Deletes a specified computer and creates a log file with BitLocker recovery key and local admin password saved to a secure network share.

Laptop Imaging Automation

Technology Used: PowerShell, VB

Summary: Customized Microsoft Deployment Toolkit (MDT) to automate laptop imaging while still meeting Accenture security standards.

Customizations:

- "Remote Imaging via USB drive." To allow remote imaging via USB drive during the COVID-19 pandemic, I read and modified several of the built-in VB scripts for MDT to suppress the "join domain" function. I then added a custom VB script that created a pop-up window on the desktop, effectively pausing the image process and allowing the support agent to connect to VPN before continuing.
- "Active Directory Computer Account Scripts." One script moved the computer account to the correct OU in AD based on its default gateway. Another script retrieved the user's email address from AD and assigned it to the computer's description to ensure Accenture compliance. A third script grabbed the laptop's serial number from BIOS and saved it to AD as a custom attribute so support agents could troubleshoot the laptop easier.
- "Custom Start Menu and Taskbar Script." Modified the default Windows setup using PowerShell commands and XML files. Found a way to add the default OneDrive and Teams icons to the start menu, which was challenging since those programs install to the user profile after they log on for the first time.
- "WDS Backup / Replication Tool." I created this tool so it was easy to backup MDT files and replicate changes I made to all our office locations around the world (MSP, SLC, MNL, LON, and SYD). After copying updates to a given server, it also ran a PowerShell command to refresh and restart the WDS service to ensure any changes pushed were then live.

Lifetouch National School Studios — March 2013 – July 2017 (4 years)

Production Lab – 7800 Picture Drive Bloomington, MN — *IT Support Level 1*

Responsibilities

- **Helpdesk:** our team handled IT and network needs of entire building (large quantity of tickets), standard SLAs.
- **After Hours Support:** on-call/onsite rotation, calls were common and often required onsite assistance.
- **Desktop/Laptop/Asset Support:** procurement, deployment, inventory, troubleshooting, repair, decommission.
- **Application Support:** upwards of 50 in-house applications, each with unique install/troubleshooting instructions.
- **Database Corrections:** old Fox Pro DBFs and MySQL, regularly queried and corrected data to fix application issues.
- **Network Support:** two server/network rooms, troubleshot network connections, ran LAN lines to patch panels, etc.
- **Server Monitoring:** Windows Servers 2008/2012, troubleshoot apache tomcats, IIS restarts, search log files, etc.
- **Active Directory:** troubleshot AD groups/permissions but escalated when changes were needed.
- **Windows XP to Windows 7 migrations (and exchange server to O365):** facilitated end user migrations.
- **Office Network Printer Support:** around 100 local office printers of varying brands and models.
- **Industrial Multimedia Printer Support:** around 30 local printers (HP and Noritsu).
- **Rimage Support (CD/DVD printing hardware):** rimage SME, setup, backup, troubleshooting, repair.
- **Audio/Video Conference Room Support:** 2-3 conference rooms.

Notable Contributions

Fox Pro Report-Generating Program

Technology Used: Fox Pro, VBA Excel

Summary: I rewrote and expanded on an old set of scripts to create a fully automated program and eliminate a manual 2 hour process. I also created a GUI so it could be maintained without me.

Details:

- The scripts were used to generate several reports. First the data was pulled from multiple CSV files and DBF tables. Then it was imported into Excel where formatting changes were made to neatly distribute the data over several pages of the spreadsheet. Finally the reports were saved to a network share for production teams to access.
- To generate all the necessary reports a support agent spent around 4 hours manually running the scripts over and over, changing parameters each time for different results.
- My program used functions and passed the parameters automatically to generate and save the reports without any human intervention.
- I also created a GUI version of the program so custom reports could still be generated if needed.
- Part of the challenge was finding an easy way to update some of the backend data due to the fact that it changed each year as production labs took on new work and processes were assigned to different locations. I accomplished this by including an option in the GUI that opened up the relevant DBF table so it could be modified. After saving the modified table, the program automatically updated the Excel templates accordingly.

PERSONAL PROJECTS

Find Aspect Ratio Tool

Technology Used: Python, JavaScript, PowerShell

Summary: As a fun exercise after taking an intro to Python course, I created a tool to find an aspect ratio given any two pixel dimensions, then I translated it to JavaScript and PowerShell to explore the different syntaxes.

Details:

- First it divides the two pixel dimensions to get a quotient, then it multiplies the quotient by *n* (where *n* is a whole number starting at 1). It loops incrementing *n* by 1 until the product is a whole number. The resulting aspect ratio is *n:product*.
- I included a conditional for aspect ratio 8:5 as it's usually written as 16:10.

Systematically Random Bingo Card

Technology Used: JavaScript

Summary: A silly web app that lets you divide your bingo card values into 3 categories of likely occurrence (high, medium, or low chance). Each time you refresh the page it systematically scrambles the card.

Details:

- I created this because my brother and I wanted to compete in a "product reveal bingo" game where you create a custom bingo card of events that are likely (or not likely) to occur during an online product reveal. Since some events are more likely to occur than others, and some spaces on the bingo card have a higher chance to create a bingo, I thought it might be fun if the card's "shuffle" function took that into account.
- The card values start in three separate arrays (most likely array, least likely array, etc.). One large array is then created by taking the values of all the original arrays and systematically passing them randomized index numbers for the new array.
- In the end, I think a fully randomized card might be a little more fun. I enjoyed the challenge though.

Phaser Game Engine Projects

Technology Used: JavaScript, Phaser.js, Matter.js, Tiled

Summary: Used the open source JavaScript game engine (Phaser) to work on a couple video game projects. One side-scrolling platformer, one top-down RPG-type game.

Details:

- Started with tutorials.
- Read through a lot of API documentation to create custom content.
- Added custom enemy objects to the game that moved at random increments of time and caused damage to the player when collision was detected. Enemies would be removed from memory if the player's weapon touched them. If an enemy was removed, a score variable would increase.
- Manipulated tile transparency with timeouts and sensors. When the player entered a building the overworld would fade to black and the building's roof would disappear. Originally, I tried to automate the tile transparency by separating roof and overworld tiles into different arrays based on custom object properties. This proved more difficult than expected and I ended up creating manual tile arrays with Phaser's built in support of the Tiled program.