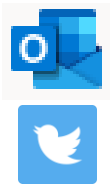# PowerShell App Deployment Toolkit

CHRIS THOMAS

DESKTOP ENGINEER - INGHAM ISD

CTHOMAS@INGHAMISD.ORG

@AUTOMATEMYSTUFF

Session Description

Even if you're just starting to learn PowerShell, you can use this powerful toolkit and it's array of functions to automate and self-document your application installations or workstation tweaks. Sure, SCCM can take an MSI and make your life easier, but what if that teacher wants the shortcut moved or the uninstaller removed from the start menu. PADT can help make your life even easier by enabling you to have better interaction with your users, simplify your deployment tasks and standardize your deployment methodology.

I challenge everyone to replace command prompt with PowerShell and not to rely on aliases in the beginning.

◦ Right-click taskbar

◦ Select Properties

◦ Select Navigation tab

◦ Check "*Replace Command Prompt with Windows PowerShell...*"

◦ Click OK

How good you think you are — Dunning-Kruger effect — Imposter Syndrome — How good you are
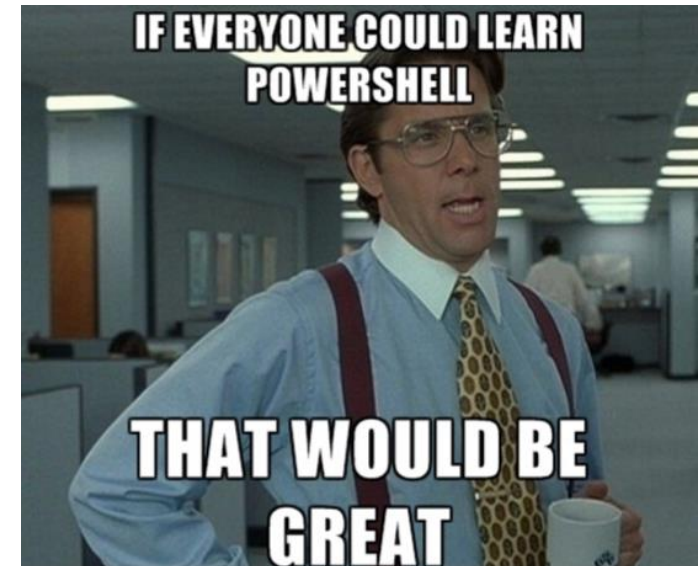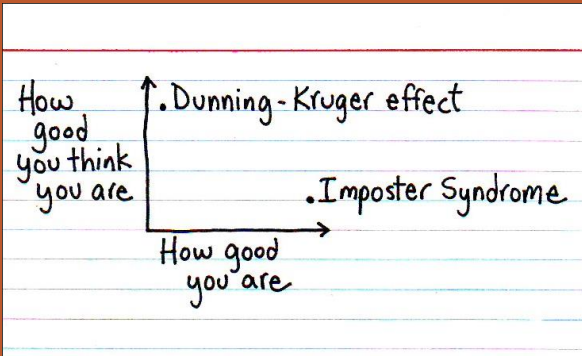
Who Am I?

- o 20 Years In K12 Technology
  - o Intern, Tech, Coordinator, Engineer

- o Lifelong Learner
  - o /r/sysadmin, /r/k12sysadmin, RSS feeds, Twitter

- o Relentlessly Inquisitive
  - o Let's Ask The WinAdmins Chat (currently Discord)

- o Problem Solver
  - o Professional Googler

- o Voracious Reader
  - o docs.microsoft.com

- o Community Minded
  - o MAEDS, MISCUG, WMiSMUG, MMS

- o #ImpostorSyndromeBeDamned
  - o Own what you know. Be confident, not arrogant. Ask questions. *Be wrong*. But don't be silent, and don't think you don't belong in the group.

Past Presentations

2013 MAEDS Fall Conference
Attended 'Marketing Yourself' by Kris Young and Kevin Galbraith
"*Our Name, Reputation and Skill Sets Need to be KNOWN*"

2014 MAEDS Spring PD Day
First time presenting and it's a 'ConfigMgr panel' for a half day session.

2014 MAEDS Fall Conference
Presented 'PADT and SCCM'

2015 MAEDS Spring PD Day
Presented 'ConfigMgr panel' for a half day session.

2015 MAEDS Fall Conference
Presented 'Automate ALL THE THINGS with PowerShell App Deployment Toolkit'

2017 MAEDS Fall Conference
Presented 'PowerShell – Intro session for those that have been too afraid to take the plunge'

2018 MAEDS Fall Conference
Presented 'PowerShell - Intermediate Session for Those That Overcame Their Fears and Took the Plunge'

2019 MACUL Pre Con Workshop
Presented 'Office 365 Admin Training'

2019 MISCUG (April)
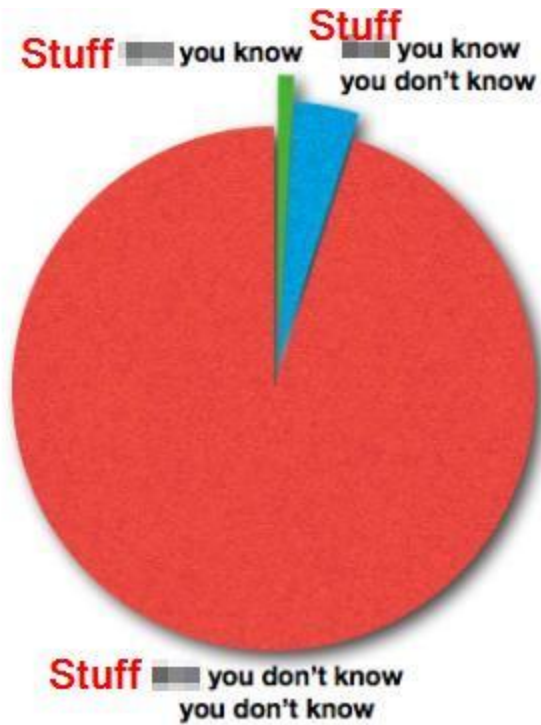Presented 'MISCUG PowerShell App Deployment Toolkit'

Be The Master

"Teaching does not always feel rewarding. It doesn't need to be. It is a repayment of something that was done for you. It is not a good thing that you do; it is an obligation that you have."

https://donjones.com/2017/10/19/become-the-master-or-go-away/
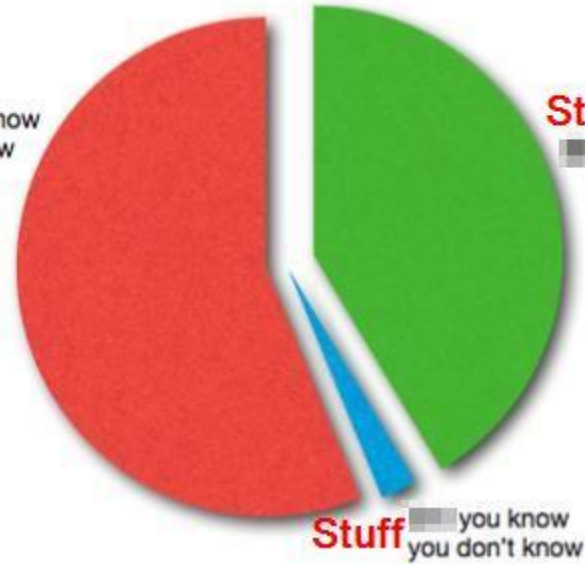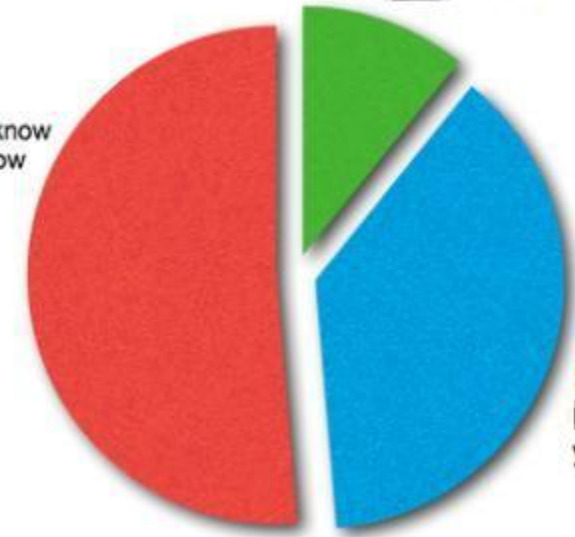
Wisdom and the Art of Being *Not Dangerous*

No One Knows What the F*** They're Doing (or "The 3 Types of Knowledge")
Steve Schwartz • May 13, 2010
https://www.bridge-global.com/blog/3-types-of-knowledge/

Don Jones: "If you're not willing to play a little bit you'll probably not be successful at PowerShell."

Jeffrey Snover: "I'm a Distinguished Engineer, I'm the Lead Architect with Windows Server and System Center Datacenter, and I invented the dang thing and still there's a struggle to it and that's normal.

Windows PowerShell Unplugged with Jeffrey Snover & Don Jones
TechEd North America 2014 • (1hr 16min • quotes @ 3:15)
https://channel9.msdn.com/Events/TechEd/NorthAmerica/2014/DCIM-B318

Vocabulary – 01/02

Shell: the command interpreter that is used to pass commands to the operating system

ISE: the Integrated Scripting Environment is an application where you can run, test, and debug scripts in a single GUI with tab completion, syntax coloring, selective execution, and context-sensitive help

Cmdlet: a task-oriented command that is typically used to return a .NET Framework object to the next command in the pipeline

Variable: a name given to stored information, e.g. $users, $iWishYouHadAStrongPassword, $x

Parameter: input values or arguments used by the cmdlet or script to make it more dynamic

- **Named**:

```
PS C:\>
PS C:\> Get-ChildItem -Path $env:USERPROFILE\AppData\Local\Temp -Filter *.exe
```

- **Positional**:

```
PS C:\>
PS C:\> Get-ChildItem $env:USERPROFILE\AppData\Local\Temp *.exe
```

- **Switch**:

```
PS C:\>
PS C:\> Get-ChildItem -Path $env:USERPROFILE\AppData\Local\Temp -Filter *.exe -Recurse
```

Object: a representation of something with methods to take actions against it and properties to access information stored within it

Vocabulary – 02/02

Module: a set of related Windows PowerShell functionalities, grouped together as a convenient unit (usually saved in a single directory)

- **Script Module**: a file (.psm1) that contains PowerShell code for functions, variables and more
- **Binary Module**: a .NET Framework assembly (.dll) that contains compiled code
- **Dynamic Module**: a module that only exists in memory (Import-PSSession)

Pipeline (|): a method to send the results of the preceding command as input to the next command

$_: "the current object in the pipeline", or "this", or $PSItem

PSSession: a persistent connection to a local or remote computer that is created, managed and closed by the user

PSDrive: a virtual drive that provides direct access to a data store (file system, registry, certificate store, SCCM)

Function: a command or series of commands grouped to run together

Alias: shortcut to a command, cmdlet or function

Tab Completion: the ability to complete cmdlet names, parameters, file paths, file names, etc with the use of the tab key

It is better to KNOW HOW TO LEARN than to know.
-Dr. Seuss

Remember Microsoft Compiled HTML Help Files?

C ▶ H ▶ M

putty.chm
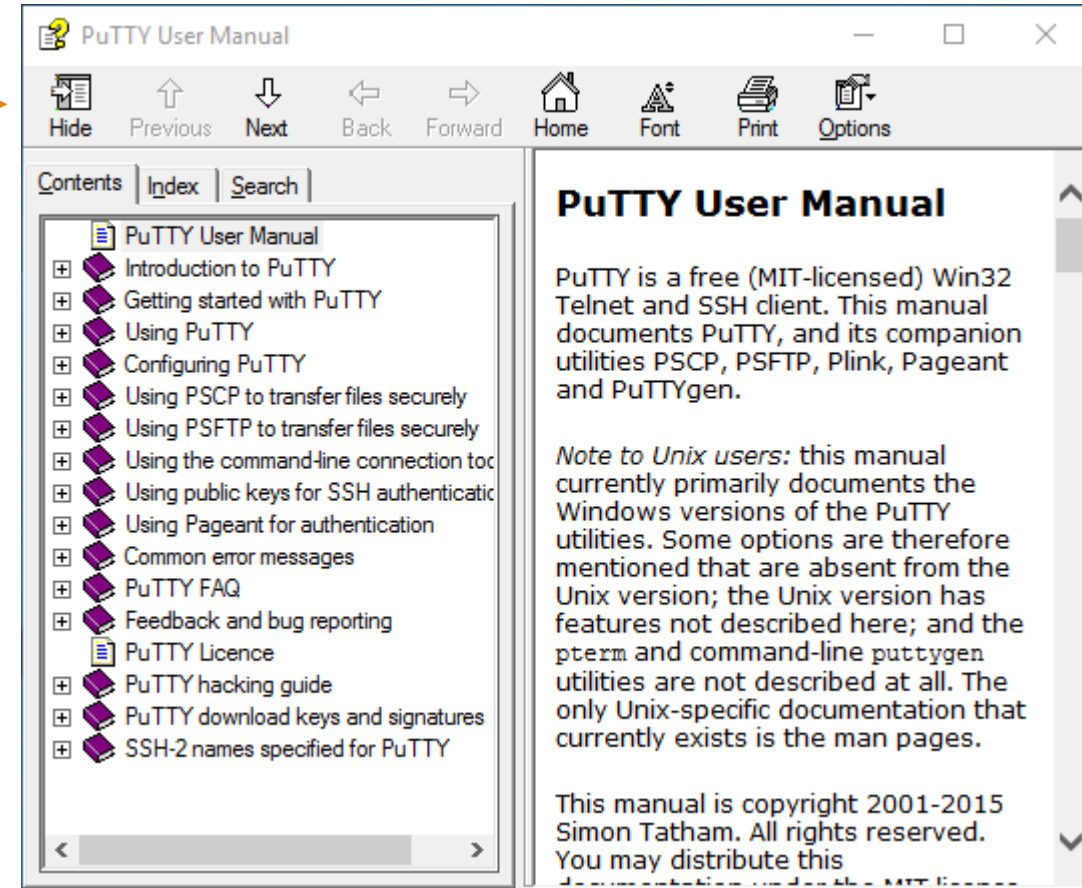
GET-

C	H	M
O	E	E O
M	L	M D
M	P	B U
A		E L
N		R E
D		

Get-Command

ALIAS: gcm

SYNOPSIS: Gets all commands.

PS C:\> Get-Command -Module *<moduleName>*

PS C:\> Get-Command -Verb Get

PS C:\> Get-Command -Verb Get -Noun P*

PS C:\> Get-Command Get-P*

PS C:\> Get-Command Get-P* -Module Microsoft.PowerShell.Utility

PS C:\> Get-Command Get-P* -Module Microsoft.PowerShell.Management

Get-Help

ALIAS: help, man

SYNOPSIS: Displays information about Windows PowerShell commands and concepts.

PS C:\> Get-Help *<cmdlet>* -Examples

PS C:\> Get-Help *<cmdlet>* -Full

PS C:\> Get-Help *<cmdlet>* -ShowWindow

PS C:\> Get-Help *<cmdlet>* -Online

PS C:\> Get-Help *<cmdlet>* | Clip

Cheat Mode For Slide Prep

PS C:\> Get-Help <cmdlet> | Select-Object -ExpandProperty Synopsis | Clip

Get-Member

ALIAS: gm

SYNOPSIS: Gets the properties and methods of objects.

PS C:\> *<cmdlet>* | Get-Member

Caveat

PS C:\> Get-ADUser cthomas | Get-Member

vs.

PS C:\> Get-ADUser cthomas -Properties * | Get-Member

Get-Module

ALIAS: gmo

SYNOPSIS: Gets the modules that have been imported or that can be imported into the current session.

PS C:\> Get-Module

PS C:\> Get-Module -ListAvailable

PS C:\> $session_DC1 = New-PSSession -ComputerName V-DC1
PS C:\> Get-Module -PSSession $session_DC1 -ListAvailable

Why PowerShell App Deployment Toolkit?

- Consistent Deployments
- Self-Documenting
- Pre-Built Logging
- Pre-Built User Interface
- Custom Branding
- Pre-Built Functions and Variables
- Extensible With Custom Functions
- SCCM Integration
- Over 10,000 Lines Of Code, FREE!
- Responsive Developers
- Good Documentation
- Active Release Schedule
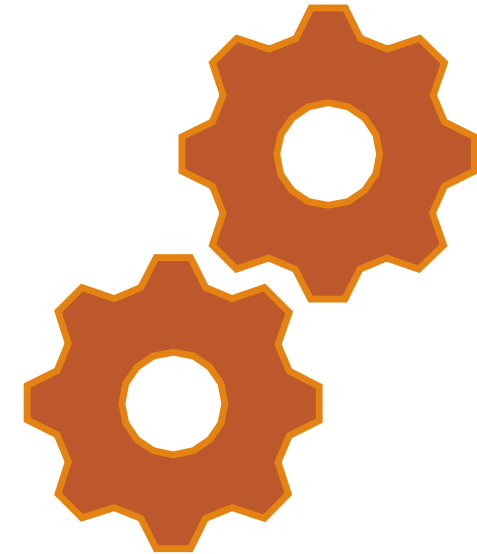
PowerShell App Deployment Toolkit

PowerShell App Deployment Toolkit Main Page
http://psappdeploytoolkit.com/

PSAppDeployToolkit ~~3.7.0~~ _3.8.0_ Direct Download
https://github.com/PSAppDeployToolkit/PSAppDeployToolkit/rele
ases/download/~~3.7.0~~ _3.8.0_/PSAppDeployToolkit_v~~3.7.0~~ _3.8.0_.zip

Discussion Boards
https://discourse.psappdeploytoolkit.com/

Authors
Seán Lillis
• twitter.com/seanels
Dan Cunningham
• twitter.com/sintaxasn

PowerShell App Deployment Toolkit Shoutouts

Become the Hero of the Day - Master ConfigMgr 2012 R2 with a Limited Budget and Free Community Tools
System Center Universe 2014 • (47min • mentioned @ 36:00)
Kent Agerlund
• twitter.com/agerlund
• blog.coretech.dk/author/kea
• https://www.youtube.com/watch?v=9hDFqcPjjSA

Microsoft System Center Configuration Manager Community Jewels
TechEd North America 2014 • (1hr 6min • mentioned @ 18:20)
Jörgen Nilsson
• twitter.com/ccmexec
• ccmexec.com
• channel9.msdn.com/Events/TechEd/NorthAmerica/2014/PCIT-B320#fbid=

New Features in AdminStudio 2016 SP2

This section lists the new features that are included in AdminStudio 2016 SP2:

- Support for PowerShell Wrapped Packages

27th May 2018

PSAppDeployToolkit now integrated in to Flexera AdminStudio

The PowerShell App Deployment Toolkit remains a free, open source, community developed tool. Due to popular demand, the PSADT is now integrated and distributed with Flexera Admin Studio.

Why we love it

This is a true timesaver, plain and simple, especially for those who don't have a lot of PowerShell experience. In addition, it provides greater control of your installation environment with functions you don't get with SCCM or other application distribution solutions function set!

Without PowerShell wrapping, your options for preparing an application for deployment are: repackage the vendor installer into a format for silent delivery mode, create a transform or run the installer silently.

By wrapping an existing Windows installer (.msi) or complex installation package (.exe) in a PowerShell script, you can automatically perform deployment tasks both before and after installation.

The advantages of using PowerShell wrapping instead of repackaging or transforms to modify an installer include:

- The original package is not modified
- A consistent UI experience to your users for all of your organization's installers
- Provides a more powerful set of functionality than can be provided using transforms

A Single Wrapper Regardless of Format

AdminStudio 2016 SP2 provides full support for converting the following deployment formats into a PowerShell wrapped package:

- Windows Installer packages .msi
- InstallShield InstallScript .exe files
- InstallShield Basic MSI installers that are compressed into a setup.exe file
- InstallShield Suite Installer .exe files
- Wise Package Studio .exe files
- Other executable file types that can be uncompressed by 7-ZIP

141 votes

Vote

Animated GIFs as icons in Software Center

It would be pretty 🔥 if software center supported animated GIFs for App icons

Riley Childs shared this idea · November 21, 2018 · Flag idea as inappropriate...

🐦 Tweet

Add a comment...

📎 Attach a File

Your email address

Sign in with: f G

Post comment

bdam commented · January 24, 2019 21:33 · Flag as inappropriate
This is the one thing keeping my organization from being able to adopt modern management in our environment and is really hurting our cloud adoption. (Try not to think about that too hard or it'll hurt.) Why even bother developing the product if this isn't included?

Latest release

PowerShell App Deployment Toolkit 3.8.0

🏷 3.8.0
-○- b231ed4

👤 sintaxasn released this 5 days ago

Version 3.8.0 [09/23/2019]

- Added ability to change the dialog banner height, with a configurable maximum
- Added ability to change the dialog banner and icon filename through the configuration file
- Added support for animated gifs in the dialog banner
- Added a dynamic dialog banner height with a configurable maximum to the dialogs
- Added support for flattening folder structure within Copy-File with -Flatten switch
- Added architecture independent logic for detecting msi's when filename ends in x64.msi or x86.msi
- Added logging to Remove-Folder function so it is also traceable in the log file when successful
- Changed ProcessObjects parameter in Get-RunningProcesses function to mandatory=$false so Show-InstallationWelcome without use ... does not ...
- Changed encoding of AppDeployToolkitConfig.xml to UTF-8 (was incorrectly UTF16)
- Fixed issue where ExecuteProcess MsiExecWaitTime used ticks instead of seconds
- Fixed ... where date ... removing illegal characters and locale-specific time / date formats
- Fixed ... certain processes using .NET ... in Show-WelcomePrompt
- Fixed issue where Write ... creating ... format ... for CMTrace ... non-English cultures
- Fixed issue ... non-standard quote characters in the Execute-MSP function
- Fixed issue where Show-WelcomePrompt hangs when ForceCloseAppsCountdown is passed to Show-InstallationWelcomePrompt and application(s) ... fixed ... apps are not running when timeout is reached
- Fixed issue where Set-PinnedApplication did not pin to / unpin from taskbar on Windows 10 because Microsoft removed the taskbar verbs
- Fixed issue where Get-InstalledApplication only supported Basic Latin unicode characters by removing logic which "removes any control characters which might interfere with logging and creating file path names from these variables", as the creation of file name is only used in Execute-MSI function and the logic there already replaces invalid filename characters
- Fixed issue where Get-LoggedOnUser failed to detect if user is member of the Administrators group, when Administrator group has an orphaned account listed
- Fixed issue where active setup execution fails when StubExePath contains ps1 file in Set-ActiveSetup function

ENDORSEMENT

20

PowerShell App Deployment Toolkit Resources

Installing Applications with PowerShell App Deployment Toolkit - Part 1
https://replicajunction.github.io/2015/04/07/installing-applications-with-psadt-part1/
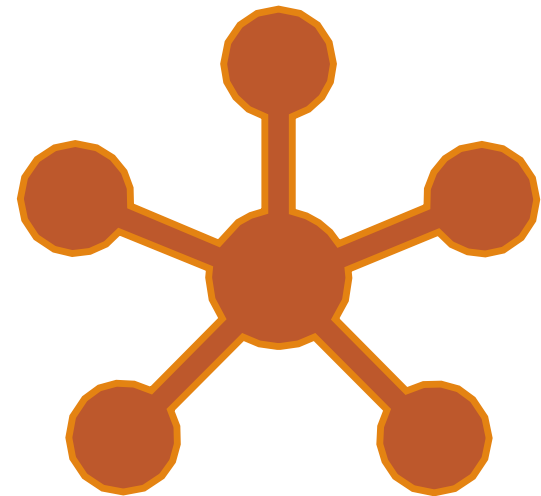
Powershell App Deployment Toolkit: Getting Started
https://www.youtube.com/watch?v=9bNsrP8iri4

Java Deployment with SCCM 2012 and PowerShell App Deployment Toolkit
http://jengeek.blogspot.com/2014/07/java-deployment-with-sccm-2012-and.html

PSAPPDEPLOYTOOLKIT: GET THE BENEFITS WITHOUT WRITING A SINGLE LINE OF POWERSHELL
https://dcunningham.net/psappdeploytoolkit-get-the-benefits-without-writing-a-single-line-of-powershell/

App Packaging Resources

Unattended/Silent Installation Switches for Windows Apps
http://unattended.sourceforge.net/installers.php

Window Development System Error Codes
https://msdn.microsoft.com/en-us/library/windows/desktop/ms681381.aspx

net helpmsg <errorcode>
http://blog.configmgrftw.com/error-codes/

Why the ConfigMgr Package model won't go anywhere
http://www.david-obrien.net/2014/04/applications-vs-packages-configmgr-package-model-wont-go-anywhere/

PowerShell App Deployment Toolkit Functions

Refresh-Desktop
Refresh-SessionEnvironmentVariables
Register-DLL
Unregister-DLL
Block-AppExecution
Close-InstallationProgress
Convert-RegistryPath
ConvertTo-NTAccountOrSID
Copy-File
Disable-TerminalServerInstallMode
Enable-TerminalServerInstallMode
Execute-MSI
Execute-MSP
Execute-Process
Execute-ProcessAsUser
Exit-Script
Get-DeferHistory
Get-FileVersion
Get-FreeDiskSpace
Get-HardwarePlatform
Get-IniValue
Get-InstalledApplication
Get-LoggedOnUser

Get-MsiExitCodeMessage
Get-MsiTableProperty
Get-ObjectProperty
Get-PEFileArchitecture
Get-PendingReboot
Get-RegistryKey
Get-RunningProcesses
Get-ScheduledTask
Get-ServiceStartMode
Get-UniversalDate
Get-UserProfiles
Get-WindowTitle
Install-MSUpdates
Install-SCCMSoftwareUpdates
Invoke-HKCURegistrySettingsForAllUsers
Invoke-ObjectMethod
Invoke-RegisterOrUnregisterDLL
Invoke-SCCMTask
New-Folder
New-MsiTransform
New-Shortcut
New-ZipFile

Remove-File
Remove-Folder
Remove-MSIApplications
Remove-RegistryKey
Resolve-Error
Send-Keys
Set-ActiveSetup
Set-DeferHistory
Set-IniValue
Set-MsiProperty
Set-PinnedApplication
Set-RegistryKey
Set-ServiceStartMode
Show-BalloonTip
Show-DialogBox
Show-InstallationProgress
Show-InstallationPrompt
Show-InstallationRestartPrompt
Show-InstallationWelcome
Show-WelcomePrompt
Start-ServiceAndDependencies
Stop-ServiceAndDependencies
Test-Battery

Test-IsMutexAvailable
Test-MSUpdates
Test-NetworkConnection
Test-PowerPoint
Test-RegistryValue
Test-ServiceExists
Unblock-AppExecution
Update-Desktop
Update-GroupPolicy
Update-SessionEnvironmentVariables
Write-FunctionHeaderOrFooter
Write-Log

PowerShell App Deployment Toolkit Variables

Toolkit Name
$appDeployToolkitName
$appDeployMainScriptFriendlyName
Script Info
$appDeployMainScriptVersion
$appDeployMainScriptMinimumConfigVersion
$appDeployMainScriptDate
$appDeployMainScriptParameters

Datetime and Culture
$currentDateTime
$currentTime
$currentDate
$currentTimeZoneBias
$culture
$currentLanguage
$currentUILanguage

Environment Variables
$envHost
$envShellFolders
$envAllUsersProfile
$envAppData
$envArchitecture
$envCommonProgramFiles
$envCommonProgramFilesX86
$envCommonDesktop
$envCommonDocuments
$envCommonStartMenuPrograms
$envCommonStartMenu
$envCommonStartUp
$envCommonTemplates
$envComputerName
$envComputerNameFQDN
$envHomeDrive
$envHomePath

$envHomeShare
$envLocalAppData
$envLogicalDrives
$envProgramFiles
$envProgramFilesX86
$envProgramData
$envPublic
$envSystemDrive
$envSystemRAM
$envSystemRoot
$envTemp
$envUserCookies
$envUserDesktop
$envUserFavorites
$envUserInternetCache
$envUserInternetHistory
$envUserMyDocuments
$envUserName
$envUserProfile
$envUserSendTo
$envUserStartMenu
$envUserStartMenuPrograms
$envUserStartUp
$envSystem32Directory
$envWinDir

Domain Membership
$IsMachinePartOfDomain
$envMachineWorkgroup
$envMachineADDomain
$envLogonServer
$MachineDomainController
$envMachineDNSDomain
$envUserDNSDomain
$envUserDomain

Operating System
$envOS
$envOSName
$envOSServicePack
$envOSVersion
$envOSVersionMajor
$envOSVersionMinor
$envOSVersionBuild
$envOSVersionRevision
$envOSProductType
$IsServerOS
$IsDomainControllerOS
$IsWorkStationOS
$envOSProductTypeName
$Is64Bit
$envOSArchitecture

Current Process Architecture
$Is64BitProcess
$psArchitecture

PowerShell And CLR (.NET) Versions
$envPSVersionTable
$envPSVersion
$envPSVersionMajor
$envPSVersionMinor
$envPSVersionBuild
$envPSVersionRevision
$envCLRVersion
$envCLRVersionMajor
$envCLRVersionMinor
$envCLRVersionBuild
$envCLRVersionRevision
Permissions/Accounts
$CurrentProcessToken
$CurrentProcessSID

$ProcessNTAccount
$ProcessNTAccountSID
$IsAdmin
$IsLocalSystemAccount
$IsLocalServiceAccount
$IsNetworkServiceAccount
$IsServiceAccount
$IsProcessUserInteractive
$LocalSystemNTAccount
$SessionZero

Script Name and Script Paths
$scriptPath
$scriptName
$scriptFileName
$scriptRoot
$invokingScript
$scriptParentPath

App Deploy Script Dependency Files
$appDeployLogoIcon
$appDeployLogoBanner
$appDeployConfigFile
$appDeployToolkitDotSourceExtensions
$xmlConfigFile
$configConfigVersion
$configConfigDate

Script Directories
$dirFiles
$dirSupportFiles
$dirAppDeployTemp

Script Naming Convention
$appVendor
$appName

$appVersion
$appLang
$appRevision
$appArch
$installTitle
$installName

Executables
$exeWusa
$exeMsiexec
$exeSchTasks
RegEx Patterns
$MSIProductCodeRegExPattern

Registry Keys
$regKeyApplications
$regKeyLotusNotes
$regKeyAppExecution
$regKeyDeferHistory

COM Objects
$Shell
$ShellApp

Log File
$logName
$logTempFolder
$configToolkitLogDir
$DisableScriptLogging
$RevertScriptLogging

Script Parameters
$deployAppScriptParameters
$appDeployMainScriptParameters
$appDeployExtScriptParameters

Logged On Users
$LoggedOnUserSessions
$usersLoggedOn
$CurrentLoggedOnUserSession
$CurrentConsoleUserSession
$RunAsActiveUser

Miscellaneous
$dpiPixels
$runningTaskSequence
$IsTaskSchedulerHealthy
$invalidFileNameChars
$useDefaultMsi

```powershell
#*===============================================
#*  PRE-INSTALLATION
If ($deploymentType -ne "uninstall") { $installPhase = "Pre-Installation"
    #*===============================================
        # Perform pre-installation tasks here
    #*===============================================
#*  INSTALLATION
$installPhase = "Installation"
    #*===============================================
        # Perform installation tasks here
    #*===============================================
#*  POST-INSTALLATION
$installPhase = "Post-Installation"
    #*===============================================
        # Perform post-installation tasks here
    #*===============================================
#*  PRE-UNINSTALLATION
} ElseIf ($deploymentType -eq "uninstall") { $installPhase = "Pre-Uninstallation"
    #*===============================================
        # Perform pre-uninstallation tasks here
    #*===============================================
#*  UNINSTALLATION
$installPhase = "Uninstallation"
    #*===============================================
        # Perform uninstallation tasks here
    #*===============================================
#*  POST-UNINSTALLATION
$installPhase = "Post-Uninstallation"
    #*===============================================
        # Perform post-uninstallation tasks here
```

Recycled Imagery - Don't Judge Me … I Was An AS400 Admin At The Time

```
PS C:\Users\chthomas> Get-WmiObject win32_product | Where-Object -Property "Name" -like "*adobe*" | Select-Object -Property Name, IdentifyingNumber

Name                                                             IdentifyingNumber
----                                                             -----------------
Adobe Reader XI (11.0.07)  MUI                                   {AC76BA86-7AD7-FFFF-7B44-AB0000000001}
Adobe Flash Player 14 ActiveX                                    {C4B95D2E-BDE6-412D-AF7B-EC43A298C55B}
Adobe Flash Player 14 Plugin                                     {3FC9A6DE-C105-4576-8F63-656FFB1BF8EB}


PS C:\Users\chthomas> gwmi win32_product | ? {$_.name -like "*adobe*"} | select name, identifyingnumber

name                                                             identifyingnumber
----                                                             -----------------
Adobe Reader XI (11.0.07)  MUI                                   {AC76BA86-7AD7-FFFF-7B44-AB0000000001}
Adobe Flash Player 14 ActiveX                                    {C4B95D2E-BDE6-412D-AF7B-EC43A298C55B}
Adobe Flash Player 14 Plugin                                     {3FC9A6DE-C105-4576-8F63-656FFB1BF8EB}


PS C:\Users\chthomas>
```
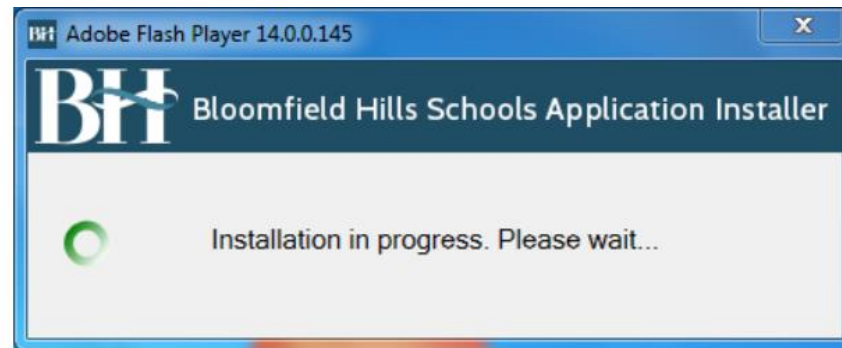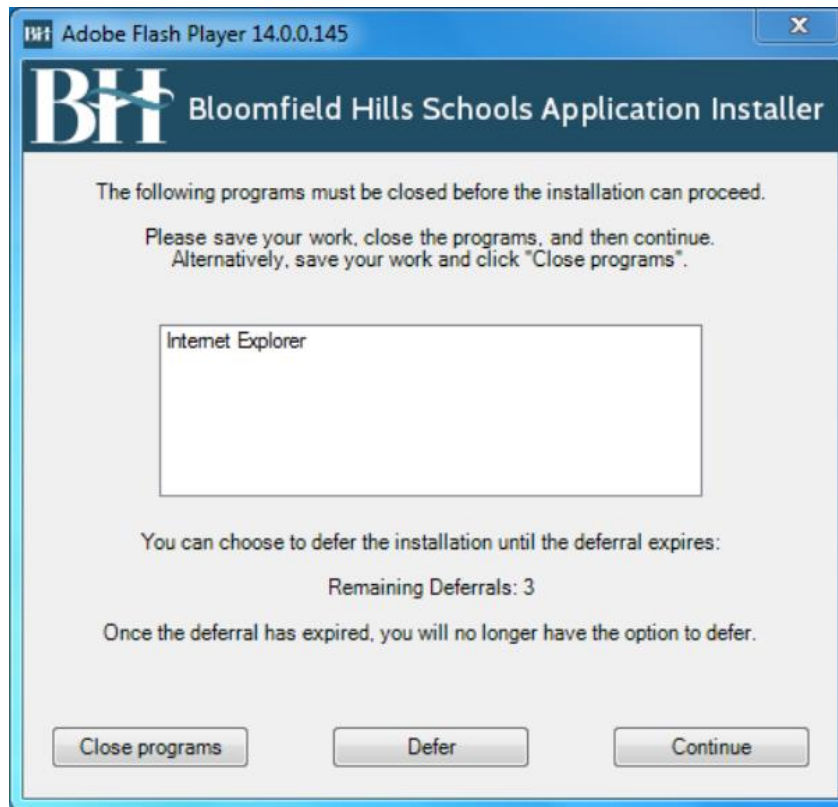
PowerShell Way
Get-WmiObject win32_product | Where-Object -Property "Name" -like "*adobe*" | Select-Object -Property Name, IdentifyingNumber
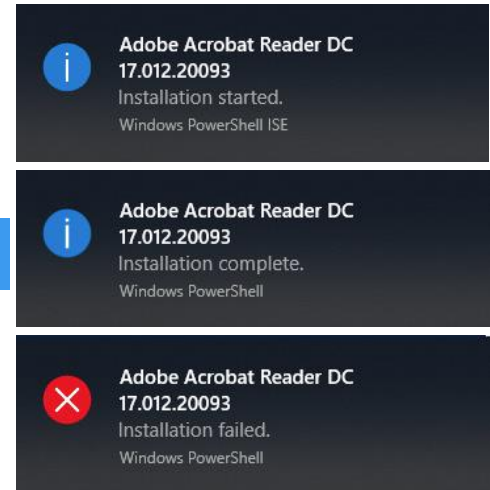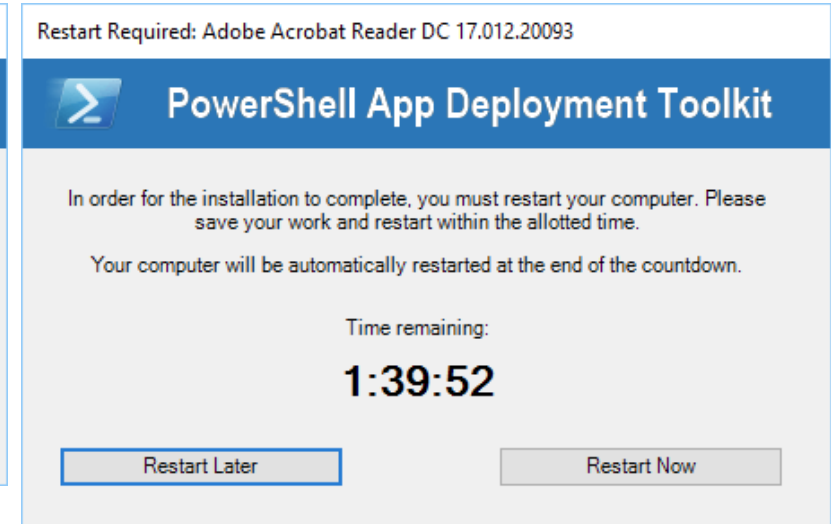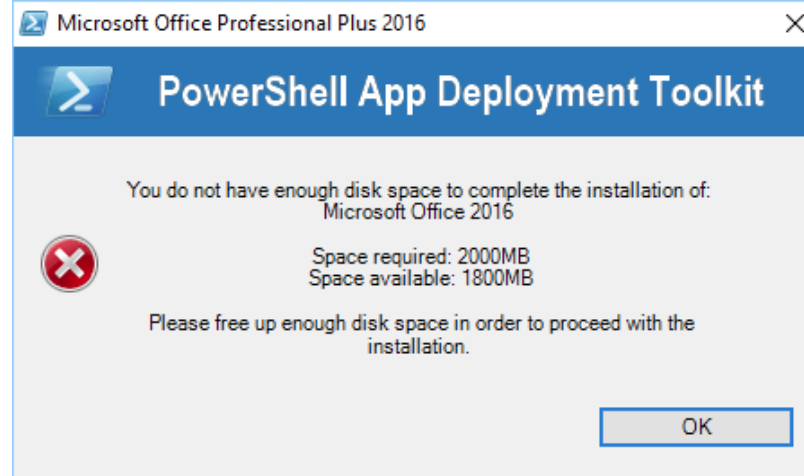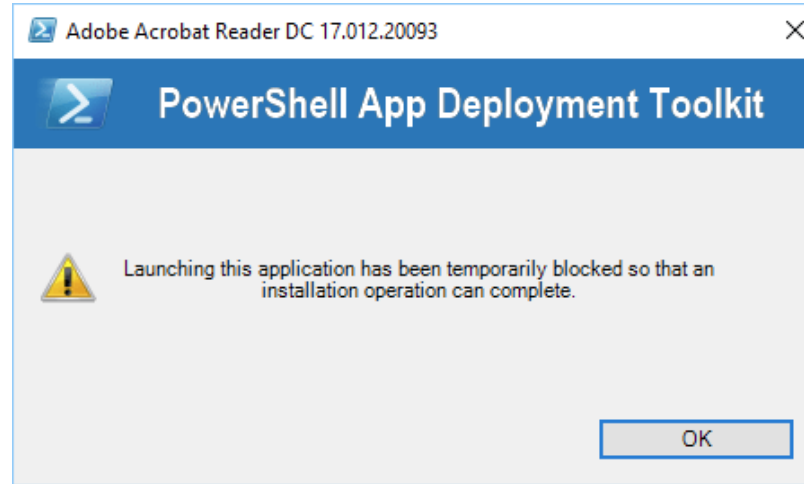
LazyMode Way
gwmi win32_product | ? {$_.name -like "*adobe*"} | select name, identifyingnumber
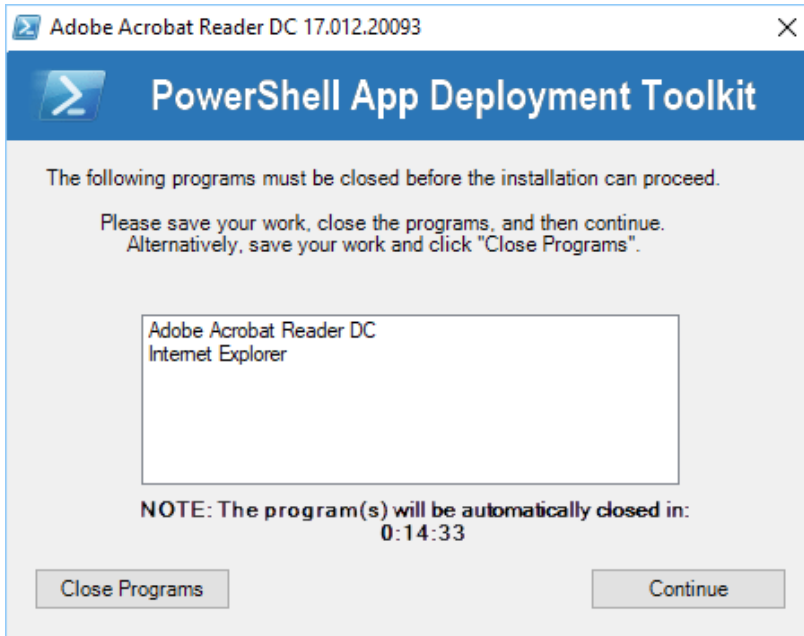
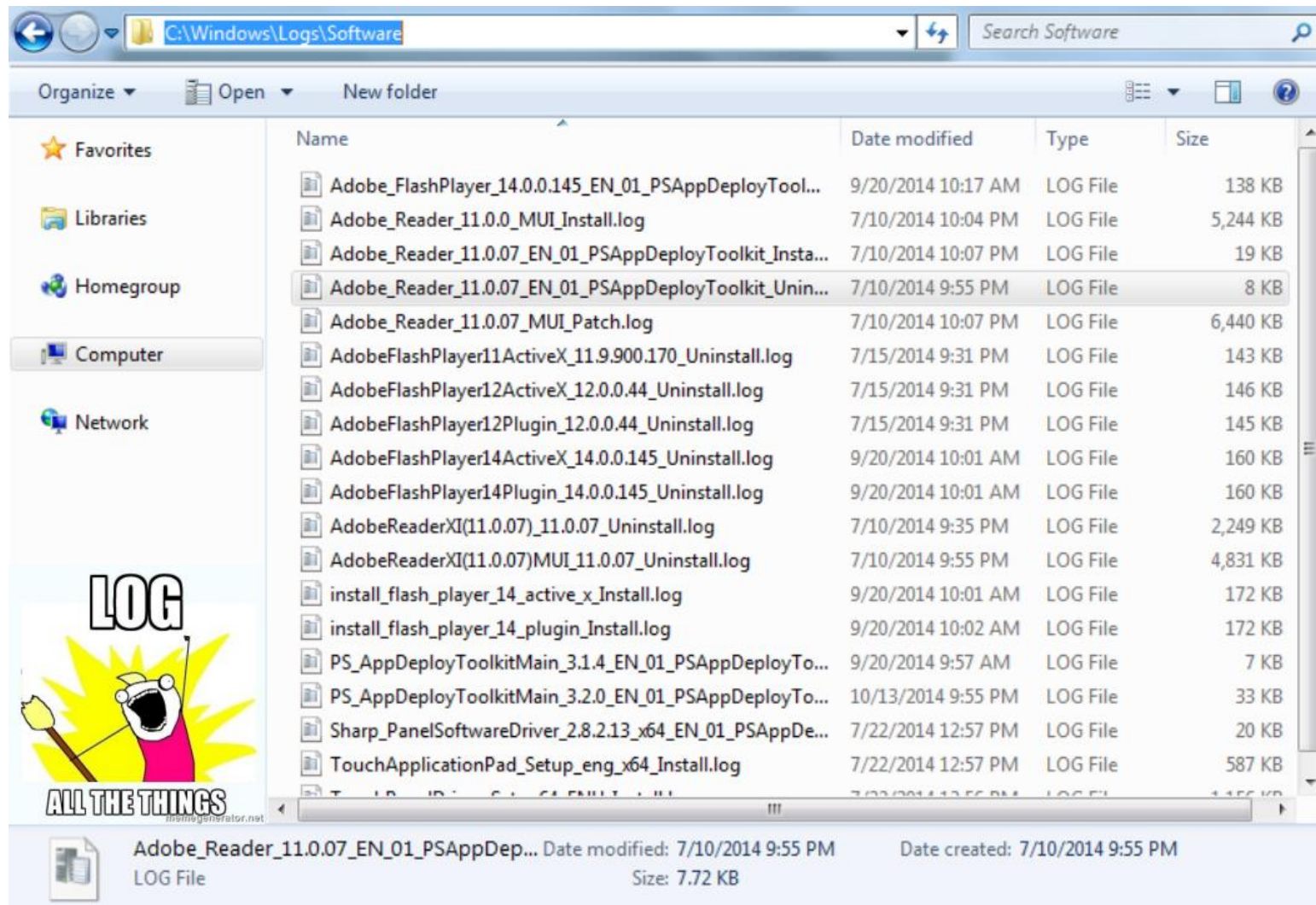# Import-Module For Tab Completion While Staging Applications

- Open… .*AppDeployToolkit\AppDeployToolkitMain.ps1*

- Save As… .*AppDeployToolkit\AppDeployToolkitMain.psm1*

- Import-Module to profile
  (you have a profile, right?)
  .\Documents\WindowsPowerShell\Microsoft.PowerShellISE_profile.ps1

- Get-Command -Module AppDeployToolkitMain

- Get-Help <cmdlet/function> -Full / -Examples

Examples

```bat
@echo off
Title Installing Itunes

REM Install iTunes x64 with pre-reqs
start /wait msiexec /i "%~dp0AppleApplicationSupport.msi" /q /norestart alluser=1
start /wait msiexec /i "%~dp0AppleMobileDeviceSupport64.msi" /q
start /wait msiexec /i "%~dp0AppleSoftwareUpdate.msi" /q
start /wait msiexec /i "%~dp0Bonjour64.msi" /q
start /wait msiexec /i "%~dp0iTunes64.msi" /q

REM Removes Desktop Shortcut - Windows 7
if exist "C:\Users\Public\Desktop\iTunes.lnk" del "C:\Users\Public\Desktop\iTunes.lnk"

REM Removes Auto Launch from registry
reg delete "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run" /v "iTunesHelper" /f

REM Copy Storytelling Alice to Local Machine
xcopy /E /Y /V %~dp0*.* "C:\Program Files (x86)\Storytelling Alice\"

REM Copy Alice Shortcut to All Users Program Menu
if exist "C:\Program Files (x86)\Storytelling Alice\Storytelling Alice.lnk" copy "C:\Program Files (x86)\Storytelling Alice\Storytelling Alice.lnk" "C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Storytelling Alice.lnk"

REM MSP Example ~~
@Echo off

msiexec.exe /i %~dp0Exceptions\AcrobatProfessional10.0-EFG\AcroPro.msi EULA_ACCEPT=YES REGISTRATION_SUPPRESS=YES SUITEMODE=1 INSTALLLEVEL=101 AS_DISABLE_LEGACY_COLOR=1 IGNOREAAM=1 TRANSFORMS=en_US.mst /qn
msiexec.exe /i %~dp0Build\Adobe_CS6_Design_and_Web_Premium_x64.msi /quiet REBOOT=ReallySuppress
msiexec.exe /p PATCH=%~dp0Exceptions\AcrobatProfessional10.0-EFG\AcrobatUpd1011.msp /qn
msiexec.exe /i %~dp0Exceptions\AdobePDFSettings11-mul\AdobePDFSettings11-mul.msi ADOBE_SETUP=1 /qn
%~dp0Exceptions\AdobeHelp\InstallAdobeHelp.exe appVersion=4.0.244  appId=chc  pubId=4875E02D9FB21EE389F73B8D1702B320485DF8CE.1  installerArg1=-silent  installerArg2=-eulaAccepted  installerArg3=-programMenu  installerArg4=AdobeHelp.a

REM Install Google Chrome
msiexec /qn /norestart /i "%~dp0GoogleChromeStandaloneEnterprise.msi"

REM Copy Master Preferences to Local Machine
if exist "c:\Program Files\Google\Chrome\Application\master_preferences" copy /y "%~dp0master_preferences" "C:\Program Files\Google\Chrome\Application\master_preferences"
if exist "c:\Program Files (x86)\Google\Chrome\Application\master_preferences" copy /y "%~dp0master_preferences" "C:\Program Files (x86)\Google\Chrome\Application\master_preferences"

REM Removes Desktop Shortcut - Windows 7
if exist "C:\Users\Public\Desktop\Google Chrome.lnk" del "C:\Users\Public\Desktop\Google Chrome.lnk"

REM Disable Auto Update
reg add HKLM\Software\Policies\Google\Update /f /v AutoUpdateCheckPeriodMinutes /d 0

REM Stop and Disable Google Update Service
sc stop gupdate
sc config gupdate start= disabled

REM Delete Google Update Scheduled Tasks
schtasks /delete /TN "GoogleUpdateTaskMachineCore" /F
schtasks /delete /TN "GoogleUpdateTaskMachineUA" /F

REM Return exit code to SCCM
exit /B %EXIT_CODE%
```

```
1    @echo off
2
3    REM This hasn't been parameterized yet, so replace the appropriate folders/files below.
4    REM Also, the jre folder must exist inside the source folder with the scripts.
5    REM I'm doing msi parameters along with regedits since the parameters didn't work consistently between versions for me...
6
7    taskkill /F /IM iexplorer.exe
8    taskkill /F /IM iexplore.exe
9    taskkill /F /IM firefox.exe
10   taskkill /F /IM chrome.exe
11   taskkill /F /IM javaw.exe
12   taskkill /F /IM jqs.exe
13   taskkill /F /IM jusched.exe
14
15   REM Call the uninstall script
16   cscript %~dp0Java_Uninstall.vbs
17
18   REM Install JRE x86
19   msiexec.exe /i %~dp0\jre1.7.0_51_x86\jre1.7.0_51.msi /passive /norestart AUTOUPDATECHECK=0 JAVAUPDATE=0 IEXPLORER=1 /l*v c:\windows\temp\Java7x86.log
20
21   REM Installing JRE x64
22   if exist %ProgramFiles(x86)%
23   msiexec.exe /i %~dp0\jre1.7.0_51_x64\jre1.7.0_51.msi /passive /norestart AUTOUPDATECHECK=0 JAVAUPDATE=0 IEXPLORER=1 /l*v c:\windows\temp\Java7x64.log
24
25   REM Disable automatic updates
26   reg delete "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run" /v SunJavaUpdateSched /f
27   reg add "HKEY_LOCAL_MACHINE\Software\JavaSoft\Java Update\Policy" /v EnableJavaUpdate /t REG_DWORD /d 0 /f
28
29   REM Start Menu shortcuts
30   rmdir "%PROGRAMDATA%\Microsoft\Windows\Start Menu\Programs\Java" /S /Q
31
32   REM Return the exit code to SCCM
33   exit /B %EXIT_CODE%
```

```vbscript
'#################################################################
'# Written by Ivan Dretvic (http://ivan.dretvic.com)
'# Creates all users Greenshot start menu short cut
'#################################################################

'This script requires a variable of %ProgramFiles% (or the location of the root folder of Greenshot software).
'This was the simplest method to create appropriate shortcut with minimal code.
On error resume next
if WScript.Arguments.Count > 0 then
    dim shell, lnkPath, link, pfpath
    Set shell = WScript.CreateObject("WScript.shell")
    lnkPath = shell.SpecialFolders("AllUsersStartMenu")
    Set link = shell.CreateShortcut(lnkPath & "\Kompozer.lnk")
    link.Description = "Kompozer - Easy web authoring"
    pfpath = WScript.Arguments.item(0)
    link.TargetPath = pfpath & "\Kompozer\kompozer.exe"
    link.WorkingDirectory = pfpath & "\Kompozer"
    link.IconLocation = pfpath & "\Kompozer\kompozer.exe,0"
    link.Save
    set shell = nothing
    set lnkpath = nothing
    set link = nothing
    set pfpath = nothing
end if
```

My Old Way To Create Start Menu Shortcuts

Firefox 38.3.0 ESR

Execute-Process, Copy-File, New-Folder, Uninstalling non-MSI Applications

```
1  ## <Perform Installation tasks here>
2  Execute-Process -Path "$dirFiles\Firefox Setup 38.3.0esr.exe" -Parameters "/INI=$dirFiles\config.ini"
3
4  ## <Perform Post-Installation tasks here>
5  #Execute-Process -Path "$envProgramFilesX86\Mozilla Maintenance Service\uninstall.exe" -Parameters "/S"
6  Copy-File -Path $dirFiles\cck2\defaults\pref\autoconfig.js -Destination "$envProgramFilesX86\Mozilla Firefox\defaults\pref"
7  Copy-File -Path $dirFiles\override.ini -Destination "$envProgramFilesX86\Mozilla Firefox\browser"
8
9  #Copy The CCK2 Stuff
10 Copy-File -Path $dirFiles\cck2\cck2.cfg -Destination "$envProgramFilesX86\Mozilla Firefox"
11 New-Folder -Path "$envProgramFilesX86\Mozilla Firefox\cck2\"
12 Copy-File -Path $dirFiles\cck2\cck2\*.* -Destination "$envProgramFilesX86\Mozilla Firefox\cck2\"
13 New-Folder -Path "$envProgramFilesX86\Mozilla Firefox\cck2\modules\"
14 Copy-File -Path $dirFiles\cck2\cck2\modules\*.* -Destination "$envProgramFilesX86\Mozilla Firefox\cck2\modules\"
```

```
1  # <Perform Uninstallation tasks here>
2  Execute-Process -Path "$envProgramFilesX86\Mozilla Firefox\uninstall\helper.exe" -Parameters '/S'
3  Get-ChildItem -Path "$envProgramFilesX86\Mozilla Firefox" -Include * | Remove-Item -Recurse
```

CODE: https://pastebin.com/6QxKiYp3

Lego Mindstorms EV3 1.1.1

Execute-Process, Execute-MSI, Post-Install Cleanup (1 installer requires 11 uninstallers for full cleanup)

```
1    ## <Perform Installation tasks here>
2    Execute-Process -Path "$dirFiles\setup.exe" -Parameters "/r:n /qn /AcceptLicenses yes"
```

Get-WmiObject -Class Win32_Product -Filter "InstallDate = '20190318'"

```
1    ## <Perform Pre-Uninstallation tasks here>
2    ## Uninstall LEGO MINDSTORMS NXT x64 Driver
3    Execute-MSI -Action Uninstall -Path "{A0831C28-A6FA-49A3-86AE-B5AE3C9EE19C}"
4    ## Uninstall LEGO MINDSTORMS EV3
5    #Execute-MSI -Action Uninstall -Path "LEGO_SW.{5B0CB826-E499-4E6B-94F0-75B6327ED934}"
6    ## Uninstall LEGO MINDSTORMS EV3 Home Edition
7    Execute-MSI -Action Uninstall -Path "{01D821CA-B361-45E2-8748-904AFEDC1DBD}"
8    ## Uninstall LEGO MINDSTORMS EV3 Uninstaller
9    Execute-MSI -Action Uninstall -Path "{5F3092B9-4240-4037-A287-BF6F9A2996BC}"
10   ## Uninstall LEGO MINDSTORMS EV3 Home Content
11   Execute-MSI -Action Uninstall -Path "{6AAF31BC-3005-429B-90D8-1C4A18DEE73A}"
12   ## Uninstall LEGO MINDSTORMS EV3 Home English Support
13   Execute-MSI -Action Uninstall -Path "{C4CF0D3D-8724-4B20-ABB0-4C73BDEA0F63}"
14   ## Uninstall VI VC2008MSMs x64
15   Execute-MSI -Action Uninstall -Path "{07E00E94-7A78-40FA-9BEF-71C190E98041}"
16   ## Uninstall NI EulaDepot
17   Execute-MSI -Action Uninstall -Path "{87F60C46-07E2-46B4-B872-680DE4184C0A}"
18   ## Uninstall NI .NET Framework 4
19   Execute-MSI -Action Uninstall -Path "{A3A11F6C-E573-4D1C-A9D4-701D7551544B}"
20   ## Uninstall NI Uninstaller
21   Execute-MSI -Action Uninstall -Path "{C7743231-5899-418D-8CA5-22B0F654D894}"
22   ## Uninstall NI VC2008MSMs x86
23   Execute-MSI -Action Uninstall -Path "{E84997A1-4D6F-4C0B-B60D-F85B360D2666}"
24   ## Uninstall NI MDF Support
25   Execute-MSI -Action Uninstall -Path "{FA35D849-889D-4454-9532-6BE2008D2CDF}"
```

CODE: https://pastebin.com/ygAZPpcX

Sharp Panel Pen Drivers & Software – 1/2

Execute-Process, Execute-MSI, Show-InstallationPrompt (Required User Interaction), Remove-File
Install Certificate
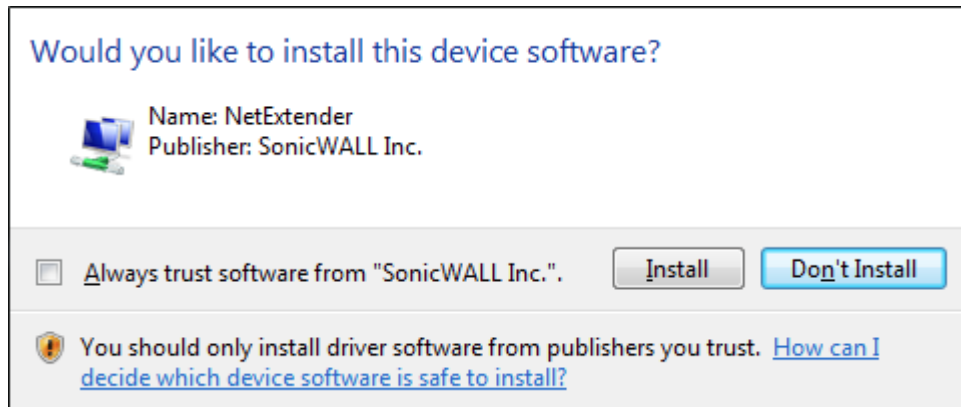
InstallShield Record Installation and Uninstallation
http://unattended.sourceforge.net/installers.php

RECORD: /r Copy: c:\windows\setup.iss
USE: /s /sms install.iss/uninstall.iss

CODE: http://pastebin.com/ZTxViEuj

Sharp Panel Pen Drivers & Software – 2/2

```
1    # Import SHARP certificate for driver installation
2    Execute-Process -FilePath "$dirSupportFiles\certutil.exe" -Arguments "-addstore -f `"TrustedPublisher`" $dirSupportFiles\sharp.cer"
3
4    # Install 64-bit Sharp Touch Panel Driver, Pen Software, Touch Application Pad Software and Pen Software Update
5    if ((Get-WmiObject -Class Win32_OperatingSystem | Select-Object OSArchitecture) -like "*64-bit*"){
6        Execute-MSI -Action Install -Path "$dirFiles\TouchPanelDriver_Setup64_ENU.msi"
7        Execute-Process -FilePath "$dirFiles\PenSoftware_Setup_x64.exe" -Arguments "/s /sms /fl$dirFiles\PenSoftware_Setup_x64_Install.iss"
8        Execute-MSI -Action Install -Path "$dirFiles\TouchApplicationPad_Setup_eng_x64.msi"
9
10       Show-InstallationPrompt -Message "This next installer could not be completely automated. You will need to proceed through the prompts to complete the installation.
         `n`nPlease click: `n`n- Next `n- Next `n- Install `n- Finish" -PersistPrompt -Icon Exclamation -ButtonMiddleText "OK"
11
12       Execute-Process -FilePath "$dirFiles\PenSoftware_(Ver.2.8.2.13)_Updater.exe" -Arguments "/s /sms /fl$dirFiles\PenSoftware_(Ver.2.8.2.13)_Updater_Install.iss"
13   }
```

```
1    # Perform uninstallation tasks here
2
3    # Prompt the user to close the following applications if they are running:
4    Show-InstallationWelcome -CloseApps "sharppen" -AllowDefer -DeferTimes 3
5    # Show Progress Message (with a message to indicate the application is being uninstalled)
6    Show-InstallationProgress -StatusMessage "Uninstalling Application $installTitle. Please Wait..."
7
8    # Uninstall x64 Touch Panel Driver
9    Execute-MSI -Action Uninstall -Path "{7626BF3E-02BF-4896-8E71-4342BD67ED8E}"
10
11   # Uninstall x64 Pen Software (pre-update)
12   # Execute-Process -FilePath "$dirFiles\PenSoftware_Setup_x64.exe" -Arguments "/s /sms /fl$dirFiles\PenSoftware_Setup_x64_Uninstall.iss"
13
14   # Uninstall x64 Touch Application Pad Software
15   Execute-MSI -Action Uninstall -Path "{D08BFAFA-2832-41C0-A092-4D30DC626168}"
16
17   # Uninstall x64 Pen Software Update
18   Execute-MSI -Action Uninstall -Path "{0164330E-769E-4F9A-B070-9256CF175659}"
19
20   # Uninstall x64 Pen Software with Update
21   Show-InstallationPrompt -Message "This next uninstaller could not be completely automated. You will need to proceed through the prompts to complete the uninstallation.
         `n`nPlease click: `n`n- Remove (radio button) `n- Next `n- Yes `n- Finish" -PersistPrompt -Icon Exclamation -ButtonMiddleText "OK"
22
23   Execute-Process -FilePath "$dirFiles\PenSoftware_(Ver.2.8.2.13)_Updater.exe" -Arguments "/s /sms /fl$dirFiles\PenSoftware_(Ver.2.8.2.13)_Updater_Uninstall.iss"
```

CODE: http://pastebin.com/ZTxViEuj

Wacom Bamboo Tablet 5.3.5-3

Post-Install Cleanup (if installer provides no opt-out for plug-ins)

```
1    # Perform installation tasks here
2    Execute-Process -FilePath "$dirFiles\setup.exe" -Arguments "/s"
```

Get-WmiObject -Class Win32_Product -Filter "InstallDate = '20190318'"

```
1    #Remove Facebook Plugins
2    Execute-Process -FilePath "$envProgramFiles\TabletPlugins\fbWTPUninstall.exe" -Arguments "/S"
3    Execute-Process -FilePath "$envProgramFilesX86\TabletPlugins\fbWTPUninstall.exe" -Arguments "/S"
```
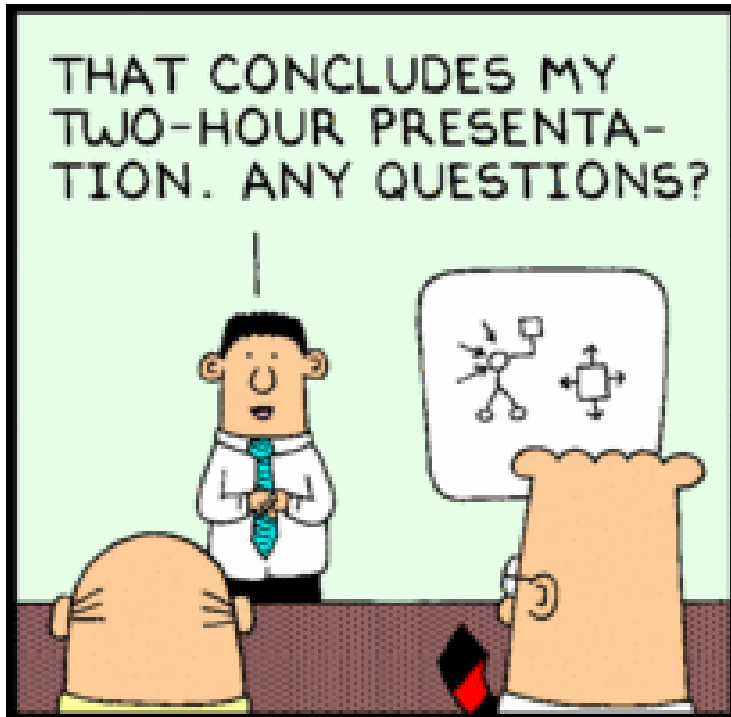
CODE: https://pastebin.com/6QxKiYp3

Cisco IP Communicator - BringToFront

Non-Standard Usage - Workstation Tweaks

```
1  ## Show Welcome Message, close Internet Explorer if required, allow up to 3 deferrals, verify there is enough disk space to complete
2  Show-InstallationWelcome -CloseApps 'communicatork9' -AllowDefer -DeferTimes 3 -CheckDiskSpace -PersistPrompt
3
4  ## <Perform Installation tasks here>
5  Set-RegistryKey -Key 'HKEY_CURRENT_USER\SOFTWARE\Cisco Systems, Inc.\Communicator' -Name 'BringToFront' -Value '1' -Type String
6
7  ## <Perform Post-Installation tasks here>
8  Execute-Process -Path "C:\Program Files (x86)\Cisco Systems\Cisco IP Communicator\communicatork9.exe" -NoWait
```

CODE: https://gist.github.com/chrisATautomatemystuff/57c79632c1682f32af826affa6fcf42c

The End