PowerShell jXchange Management Console Guide

PowerShellJMC

Release 2019

**PowerShellJMC**

Guide to the PowershellJMC

Guide

[Summary 1](#_Toc7444772)

[Prerequisites 1](#_Toc7444773)

[Get-Help 1](#_Toc7444774)

[Setup, Installation, and Updates 1](#_Toc7444775)

[Setup Artifactory 1](#_Toc7444776)

[Installation and/or Update 1](#_Toc7444777)

[Uninstall 1](#_Toc7444778)

[Cleanup 1](#_Toc7444779)

[Cmdlet List 1](#_Toc7444780)

[PowerShell ISE 1](#_Toc7444781)

[Authentication 1](#_Toc7444782)

[New-JMCConnection 1](#_Toc7444783)

[Processes 1](#_Toc7444784)

[Basic Get-SGInstitution Usage 1](#_Toc7444785)

[Basic Add-SGInsitution Usage 1](#_Toc7444786)

[Advanced Add-SGInsitution Usage with Export-Excel and Import-Excel 1](#_Toc7444787)

[Basic Add-SGUser Usage 1](#_Toc7444788)

[Basic Add-SGHostedProviderUser Usage 1](#_Toc7444789)

[Add-SGProvider 1](#_Toc7444790)

[Export-SGInstitution 1](#_Toc7444791)

[Import-SGInstitution 1](#_Toc7444792)

[Working with Providers - New-SGProviderSilverlake 1](#_Toc7444793)

[Excel Spreadsheet Building 1](#_Toc7444794)

[Basic Add-IMSCustomClaim Usage 1](#_Toc7444795)

# Summary

This document is best viewed in web view, click on its icon in the bottom left part of word. At my resolution 170% looks nice.



To use any of the PowerShell, double click on the box with the PowerShell scripts in them to copy the script.

This document will attempt to discus how to use the PowerShellJMC cmdlets. This document assumes some basic knowledge of PowerShell.

Tool advertisement is located [here](https://tools.jhacorp.com/powershelljmc). Please use chrome when using the tools website.

This can be used with another set of cmdlets, [PowerShellWSTT](https://tools.jhacorp.com/powershellwstt).

Cmdlets are **Verb-Noun** structured, they should be intuitively named.

Safe JMC cmdlets to play with at first are the cmdlets that start in Get-\*, New-\*, and Export-\*. These currently do not do anything to the system.

## Prerequisites

* Suggested Training:

* + [Learn PowerShell in a month of lunches](https://www.youtube.com/watch?v=6CRTahGYnws&list=PL6D474E721138865A) by Don Jones.
  + This one is from [Microsoft](https://www.youtube.com/watch?v=wUnYLfXm81s&list=PLsrZV8shpwjMXYBmmGodMMQV86xsSz1si).
  + This set is from [Mike Shepard](https://teams.microsoft.com/l/entity/com.microsoft.teamspace.tab.wiki/tab::0865db5d-130a-4a8b-8ad8-8fd7015fad91?context=%7B%22subEntityId%22%3A%22%7B%5C%22pageId%5C%22%3A2%2C%5C%22origin%5C%22%3A2%7D%22%2C%22channelId%22%3A%2219%3Abd88a415aad84a3cae41d34c589993e8%40thread.skype%22%7D&tenantId=53049b77-3e8f-4792-977f-0a3e5f23891b).
  + PowerShell is about Discovery not Memory.
* [Windows Management Framework 5.1](https://www.microsoft.com/en-us/download/details.aspx?id=54616)
* ‘Windows PowerShell ISE’ is recommended to be used instead of the ‘Windows PowerShell’, the IntelliSense works best in the ISE.
* Install-Module -Name ImportExcel -Force

### Get-Help

Rule of Thumb – use Get-Help before going anywhere else.

There is a possible solution to what you are trying to do in the help itself before going elsewhere.

Also think about where do I want to start, and what is my end goal.

All cmdlets contain their own help. Here is an example of one.

|  |
| --- |
| **Script Snippet** |
| #show help in output window below  Get-Help Get-SGInstitution -Full  #show help in a popup window  Get-Help Get-SGInstitution -ShowWindow |

## Setup, Installation, and Updates

The PowerShellJMC is located in a PowerShellGet repository. PowerShellGet is only located in PowerShell 5.1 framework. The repository helps with distribution and version control. The repository location is accessible anywhere in Jack Henry, Eden, jha.dev, Personal Laptops, Hosting. The repository is kind of like an App Store for PowerShell.

### Setup Artifactory

Setup Access to the Artifactory Repository and Register it. Use [this script](https://tools.jhacorp.com/artifactory-root-cert-install) on the server or workstation that will be using the cmdlets. This script will install the root certs of the Artifactory and then register the repository.

|  |
| --- |
| **Script Snippet** |
| #temp fix the cert error  add-type @"  using System.Net;  using System.Security.Cryptography.X509Certificates;  public class TrustAllCertsPolicy : ICertificatePolicy {  public bool CheckValidationResult(  ServicePoint srvPoint, X509Certificate certificate,  WebRequest request, int certificateProblem) {  return true;  }  }  "@  [System.Net.ServicePointManager]::CertificatePolicy = New-Object TrustAllCertsPolicy    #download certs  c:  cd\  md -Path c:\temp -ErrorAction SilentlyContinue  Invoke-WebRequest -Uri https://artifactory.jkhy.com:443/techserv-eis-packages/ArtifactoryCerts/ETSMMOMNPKICA02-CA.cer -OutFile C:\temp\ETSMMOMNPKICA02-ArtifactoryRoot.cer  Invoke-WebRequest -Uri https://artifactory.jkhy.com:443/techserv-eis-packages/ArtifactoryCerts/ETSMMOMNPKIOR02-CA.cer -OutFile C:\temp\ETSMMOMNPKIOR02-ArtifactoryRoot.cer  #new way, not as good, not always supported  #Import-Certificate -FilePath C:\temp\ETSMMOMNPKICA02-ArtifactoryRoot.cer -CertStoreLocation Cert:\LocalMachine\Root  #Import-Certificate -FilePath C:\temp\ETSMMOMNPKIOR02-ArtifactoryRoot.cer -CertStoreLocation Cert:\LocalMachine\Root    #Legacy  C:\Windows\System32\certutil.exe -addstore "CA" "C:\temp\ETSMMOMNPKICA02-ArtifactoryRoot.cer"  C:\Windows\System32\certutil.exe -addstore "Root" "C:\temp\ETSMMOMNPKIOR02-ArtifactoryRoot.cer"  #Register the Repository  Register-PSRepository -Name EisArtifactory -SourceLocation https://artifactory.jkhy.com/api/nuget/techserv-eis-powershellget -InstallationPolicy Trusted |

### Installation and/or Update

This command will install or update the module that contains all of the released cmdlets. You might be prompted to update PowerShellGet, make sure to click **Yes**. If this command has been used, there is no need to use Import-Module unless you want to see all of the cmdlets.

|  |
| --- |
| **Script Snippet** |
| Install-Module -Name PowerShellJMC -Repository EisArtifactory -Force |

|  |
| --- |
| **CAUTION** |
| Because this cmdlet is a dll, you will need to “Restart” the PowerShell ISE to get access to the new cmdlets. There is no command to unload the dll after it is loaded. |

### Uninstall

This command will uninstall the module.

|  |
| --- |
| **Script Snippet** |
| Uninstall-Module -Name PowerShellJMC -AllVersions -Force |

### Cleanup

This command will delete all the old versions of the module.

|  |
| --- |
| **Script Snippet** |
| Get-InstalledModule -name PowerShellJMC -AllVersions | ? {$\_.Version -ne (Get-InstalledModule -name PowerShellJMC).Version} | Uninstall-Module |

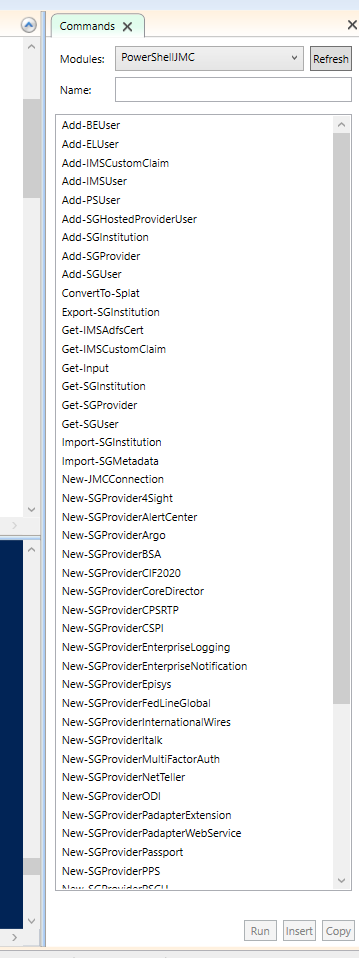
## Cmdlet List

Here is a list of cmdlets that is currently accessible, new ones might be added at any time upon request. Requests need to be made to [info-EISTools@jackhenry.com](mailto:info-EISTools@jackhenry.com). Most of these are self-explanatory.

* Add-IMSUser
* Add-PSUser
* Add-ELUser
* Add-BEUser
* New-SGProviderBSA
* New-SGProviderYellowHammer
* New-SGProviderPassport
* New-SGProviderPPS
* New-SGProviderCPSRTP
* New-SGProviderPSCU
* New-SGProviderSynergy
* New-SGProviderODI
* New-SGProviderCSPI
* New-SGProviderEnterpriseLogging
* New-SGProviderAlertCenter
* New-SGProviderArgo
* New-SGProviderItalk
* New-SGProviderInternationalWires
* New-SGProviderFedLineGlobal
* New-SGProviderSymantecMFA
* Get-IMSCustomClaim
* Add-IMSCustomClaim
* Add-SGInstitution
* Add-SGHostedProviderUser
* Get-SGUser
* Get-SGProvider
* Get-Input
* ConvertTo-Splat
* Import-SGInstitution
* Export-SGInstitution
* New-JMCConnection
* New-SGProviderPadapterExtension
* New-SGProviderPadapterWebService
* New-SGProviderCoreDirector
* New-SGProvider4Sight
* New-SGProviderEpisys
* New-SGProviderSynapsys
* New-SGProviderNetTeller
* New-SGProviderMultiFactorAuth
* Get-IMSAdfsCert
* New-SGProviderCIF2020
* Remove-SGProvider
* Set-IMSAdfsCert
* Set-JxAdfsAccountAuth
* Test-IAdapterPort
* Test-AdfsRP
* Remove-SGInstitution
* New-SGProviderEnterpriseNotification
* New-SGProviderSilverlake
* Add-SGUser
* Add-SGProvider
* Get-SGInstitution
* Import-SGMetadata

### PowerShell ISE

The Commands tab is helpful when trying to see all the supported cmdlets. Locate it inside the PowerShell ISE. This will help you discover the cmdlets. Clicking a cmdlet will let you see all of its properties. Most of the properties have been named the same as the GUI JMC.



## Authentication

All the Add, Get, Import, Export, and Remove cmdlets authenticate by default to <https://localhost> with the (IWA) Integrated Windows Authentication by passing your logged in user to the endpoint just like the JMC GUI does. A New-JMCConnection can be used to specify a server and/or a different logon credentials.

### New-JMCConnection

Here are just a few ways to use the JMC Connection cmdlet.

This is an example of what we use in QA to connect allow us to connect from jhacorp.com to qa.edn. We do not need this much.

|  |
| --- |
| **Script Snippet 1** |
| #help  Get-Help New-JMCConnection -ShowWindow  #Create a credential object  $cred = Get-Credential -UserName "user@domain.com" -Message "Enter an account to access the jXChange Server:"  #Create's a connection to another server and/or different domain user credential.  $connection = New-JMCConnection -JMCServerName jX2017padapter.qa.edn -JMCUserName $cred.UserName -JMCPassword $cred.Password  #Cause all specified cmdlets to use $connection  $PSDefaultParameterValues = @{‘Add-SGInstitution:JMCConnection’=$connection; ‘New-SGProviderSilverlake:JMCConnection’=$connection; ‘Add-SGProvider:JMCConnection’=$connection; ‘Add-SGUser:JMCConnection’=$connection; ‘New-SGProviderCIF2020:JMCConnection’=$connection; ‘Add-SGHostedProviderUser:JMCConnection’=$connection; 'Get-SGInstitution:JMCConnection'=$connection; 'New-SGProviderPadapterWebService:JMCConnection'=$connection ; 'Remove-SGProvider:JMCConnection'=$connection; 'Add-IMSCustomClaim:JMCConnection'=$connection ; 'Export-SGInstitution:JMCConnection'=$connection; 'Get-SGProvider:JMCConnection'=$connection }  #Call a cmdlet and the connection will be automatically passed to it because it was specified in the $PSDefaultParameterValues, this makes your scripts cleaner.  Get-SGInstitution |

If you are in the same domain there is no need to specify a user.

|  |
| --- |
| **Script Snippet 2** |
| #help  Get-Help New-JMCConnection -ShowWindow  #Create's a connection to another server.  $connection = New-JMCConnection -JMCServerName jX2017padapter.qa.edn  #Cause all specified cmdlets to use $connection  $PSDefaultParameterValues = @{‘Add-SGInstitution:JMCConnection’=$connection; ‘New-SGProviderSilverlake:JMCConnection’=$connection; ‘Add-SGProvider:JMCConnection’=$connection; ‘Add-SGUser:JMCConnection’=$connection; ‘New-SGProviderCIF2020:JMCConnection’=$connection; ‘Add-SGHostedProviderUser:JMCConnection’=$connection; 'Get-SGInstitution:JMCConnection'=$connection; 'New-SGProviderPadapterWebService:JMCConnection'=$connection ; 'Remove-SGProvider:JMCConnection'=$connection; 'Add-IMSCustomClaim:JMCConnection'=$connection ; 'Export-SGInstitution:JMCConnection'=$connection; 'Get-SGProvider:JMCConnection'=$connection }  #Call a cmdlet and the connection will be automatically passed to it because it was specified in the $PSDefaultParameterValues, this makes your scripts cleaner.  Get-SGInstitution |

## Processes

For each feature in the JMC GUI there is a corresponding cmdlet. Example, Get-SGInstitution does the same thing as the first screen in the JMC, it gives you the list of all the institutions. Some of the cmdlets can be setup in template form because none of the values change like on the MFA provider. Other cmdlets can use an excel sheet to changes the values of the properties.

|  |
| --- |
| **TIP** |
| Remember, and cmdlet’s parameters can be filled out from an excel file if the column names match up to the cmdlet’s parameter -parmname names. |

### Basic Get-SGInstitution Usage

Here are a few ways to use this cmdlet. It produces an initial list of institutions like the Manage Institutions.

|  |
| --- |
| **Script Snippet** |
| #help  Get-Help Get-SGInstitution -ShowWindow  #Show all the institutions in the output window  Get-SGInstitution  #Show all the institutions in a GUI grid to pick from  Get-SGInstitution | Out-GridView  #Show all the institutions in a GUI grid to pick from and store it in a variable to be used later  $institution = Get-SGInstitution | Out-GridView -PassThru |

Next, Let’s take it a step further. Let’s produce a usable list of institution objects like the “View by Provider” button in the JMC.

|  |
| --- |
| **Script Snippet** |
| #help  Get-Help Get-SGInstitution -ShowWindow  #Get FI by Provider Type  Get-SGInstitution -FilterByProvider "pAdapter\_Extension"  #Get FI by Provider Type "SilverLake"  Get-SGInstitution -FilterByProvider SilverLake  #Get FI by Provider Type "SilverLake" AND give me only the ABA’s with 555555555  Get-SGInstitution -ABA 555555555 -FilterByProvider SilverLake |

### Basic Add-SGInsitution Usage

This will add an institution.

|  |
| --- |
| **Script Snippet** |
| #help  Get-Help Add-SGInstitution -ShowWindow  #Add ServiceGateway FI  Add-SGInstitution -Name TestingJMCPowerShell -Environment PROD -ABA 123456788 -Description "myBank, City - State" -City Monett -State MO -Country US |

### Advanced Add-SGInsitution Usage with Export-Excel and Import-Excel

Here the goal is to create an excel file that can be used to create many institutions with a spreadsheet. And cmdlet can be used this way to pipe data into it from other objects or excel files, just make sure that the properties match up.

|  |
| --- |
| **Script Snippet** |
| #help  Get-Help Add-SGInstitution -ShowWindow  #(optional) Add a temporary ServiceGateway FI or use an existing one  Add-SGInstitution -Name TestingJMCPowerShell -Environment PROD -ABA 123456788 -Description "myBank, City - State" -City Monett -State MO -Country US  #This will form the base for our bulk Add-SGInsitution  Get-SGInstitution -Name TestingJMCPowerShell | Export-Excel -Path C:\temp\TempFiExport.xlsx  #Cleanup - Delete the columns that are not needed in the excel file: Mod\_User, Mod\_DateTime, InformationMessages, HasInformationMessages, JMCConnection, CurrentPSTransaction, CommandOrigin.  #Add some row data  #Save the excel file  #Use the excel file to feed data to the Add-SGInstitution cmdlet to create institutions in SG.  Import-Excel -Path C:\temp\TempFiExport.xlsx | Add-SGInstitution    #Next, Report on what was done: Use the excel file to feed data to the Add-SGInstitution cmdlet to create institutions in SG.  Import-Excel -Path C:\temp\TempFiExport.xlsx | Add-SGInstitution | Export-Csv C:\temp\exportedUserReport.csv |

### Basic Add-SGUser Usage

This will add a user if it doesn’t exist and then add the selected operations.

|  |
| --- |
| **Script Snippet** |
| #help  Get-Help Add-SGUser -ShowWindow  #Add 1 user and check all operations  Add-SGUSer -ABA 123456788 -Environment PROD -UserName user@domain.com  #Add 1 user and check selected operations  Add-SGUSer -ABA 123456788 -Environment PROD -UserName user@domain.com -Operations MFAOOBValidate, MFAQnAPolInq, Ping  #Add 1 user and check entire group of operations  Add-SGUSer -ABA 123456788 -Environment PROD -UserName user@domain.com -OperationGroups ACH |

### Basic Add-SGHostedProviderUser Usage

This will add a user if it doesn’t exist and then add the selected operations.

|  |
| --- |
| **Script Snippet** |
| #help  Get-Help Add-SGProvider -ShowWindow  #Add 1 user and check all operations  Add-SGHostedProviderUser -UserName user@domain.com -ProviderTypeName iPay  #Add 1 user and check selected operations  Add-SGHostedProviderUser -UserName user@domain.com -ProviderTypeName iPay -Operations MFAOOBValidate, MFAQnAPolInq, Ping  #Add 1 user and check entire group of operations  Add-SGHostedProviderUser -UserName user@domain.com -ProviderTypeName iPay -OperationGroups ACH |

### Add-SGProvider

Adds a provider to an institution on a local or remote ServiceGateway server.

|  |
| --- |
| **Script Snippet** |
| #help  Get-Help Get-SGProvider -ShowWindow  #create a provider and then send it to Add-SGProvider  New-SGProviderMultiFactorAuth -Name bob -Environment PROD -ABA 012345678 -ProviderHostName b -CallerCredential b -CallerId b -InstanceId b -LegacyInstanceId b -LegacyCallerId b -LegacyCallerCredential b -CertFilePath b -GroupKey b -LicenseKey b -Target b -TargetBackup b -CertPassword b -NCPLogFile b | Add-SGProvider  #create a provider and then send it to Add-SGProvider with a variable  $mfa = New-SGProviderMultiFactorAuth -Name bob -Environment PROD -ABA 012345678 -ProviderHostName b -CallerCredential b -CallerId b -InstanceId b -LegacyInstanceId b -LegacyCallerId b -LegacyCallerCredential b -CertFilePath b -GroupKey b -LicenseKey b -Target b -TargetBackup b -CertPassword b -NCPLogFile b  $mfa | Add-SGProvider  #create a provider and then send it to Add-SGProvider with as a parameter  $mfa = New-SGProviderMultiFactorAuth -Name bob -Environment PROD -ABA 012345678 -ProviderName name -ProviderDescription desc -ProviderHostName b -CallerCredential b -CallerId b -InstanceId b -LegacyInstanceId b -LegacyCallerId b -LegacyCallerCredential b -CertFilePath b -GroupKey b -LicenseKey b -Target b -TargetBackup b -CertPassword b -NCPLogFile b  Add-SGProvider -Provider $mfa  #This is how to create a report of your work and look at the results  $slReport = New-SGProviderSilverlake -Name Matthew -Environment PROD -ABA 123456788 -ProviderHostName 10.1.1.1 -ProviderPort 40001 -ProviderSecurityKey bob -ProviderInstitutionId 20 | Add-SGProvider  $slReport | Export-Csv c:\temp\Report-SL-Work.csv |

### Export-SGInstitution

Export institutions from a local or remote ServiceGateway server.

|  |
| --- |
| **Script Snippet** |
| #help  Get-Help Export-SGInstitution -ShowWindow  #Exports all institutions  Get-SGInstitution | Export-SGInstitution -password myCrazyPassword -path C:\temp\Exported  #Exports selected institutions  Get-SGInstitution | Out-GridView -PassThru | Export-SGInstitution -Password myPassword -Path c:\temp\testFolderExport |

### Import-SGInstitution

Import institutions from a local or remote ServiceGateway server.

|  |
| --- |
| **Script Snippet** |
| #help  Get-Help Import-SGInstitution -ShowWindow  #one liner to pick a single export file, this gives the user a filter and can select one or more things to import  dir C:\Temp\testFolderExport\\* | Out-GridView -PassThru | Import-SGInstitution -Password myPassword  #CAUTION: this imports ALL the exported xml files from another server  dir C:\Temp\testFolderExport\\* | Import-SGInstitution -Password myCrazyPassword |

### Working with Providers - New-SGProviderSilverlake

Creates a new ServiceGateway SilverLake Provider object. All providers work the same way, it doesn’t matter what the different parameters are named.

|  |
| --- |
| **Script Snippet** |
| #help  Get-Help New-SGProviderSilverlake -ShowWindow  #create and add the Silverlake provider  New-SGProviderSilverlake -Name Matthew -Environment PROD -ABA 123456788 -ProviderHostName 10.1.1.1 -ProviderPort 40001 -ProviderSecurityKey bob -ProviderInstitutionId 20  #Once it has been created now it needs to be added to the the institution in servicegateway. It needs to be sent | into Add-SGProvider  New-SGProviderSilverlake -Name Matthew -Environment PROD -ABA 123456788 -ProviderHostName 10.1.1.1 -ProviderPort 40001 -ProviderSecurityKey bob -ProviderInstitutionId 20 | Add-SGProvider |

### Excel Spreadsheet Building

All cmdlets can be exported to excel to form all the columns needed to create mass installs. As long as the column/properties are different then spreadsheet’s can be combined excluding “-Name, -ABA, -Environment”.

|  |
| --- |
| **Script Snippet – Add-SGInstitution Export Example** |
| #help  Get-Help Add-SGInstitution -ShowWindow  #To export this Fi to a CSV file to create an excel spreadsheet, use the following 2 lines  $myFi = Add-SGInstitution -Name TestingJMCPowerShell -Environment PROD -ABA 123456788 -Description "myBank, City - State" -City Monett -State MO -Country US  $myFi | Export-Csv c:\temp\Bulk-FI-Template.csv |

|  |
| --- |
| **Script Snippet – New-SGProviderSilverlake Export Example** |
| #help  Get-Help New-SGProviderSilverlake -ShowWindow  #To export this provider to a CSV file to create an excel spreadsheet, use the following 2 lines  $myPrvObj = New-SGProviderSilverlake -Name Matthew -Environment PROD -ABA 123456788 -ProviderHostName 10.1.1.1 -ProviderPort 40001 -ProviderSecurityKey bob -ProviderInstitutionId 20  $myPrvObj.Provider | Export-Csv -Path .\Bulk-SLprovider-Template.csv |

|  |
| --- |
| **Script Snippet – Add-SGUser Export Example** |
| #help  Get-Help Add-SGUser -ShowWindow  #To export this user to a CSV file to create an excel spreadsheet, use the following 2 lines  $myUser = Add-SGUSer -ABA 123456788 -Environment PROD -UserName user@domain.com -OperationGroups ACH -Operations MFAOOBValidate  $myUser | Export-Csv c:\temp\Bulk-User-Template.csv |

|  |
| --- |
| **Script Snippet – Add-SGHostedProviderUser Export Example** |
| #help  Get-Help Add-SGHostedProviderUser -ShowWindow  #To export this user to a CSV file to create an excel spreadsheet, use the following 2 lines  $myUser = Add-SGHostedProviderUser -UserName user@domain.com -ProviderTypeName iPay -OperationGroups ACH -Operations MFAOOBValidate  $myUser | Export-Csv c:\temp\Bulk-User-Template.csv |

### Basic Add-IMSCustomClaim Usage

This will add a user if it doesn’t exist and then add the Custom Claim to it.

|  |
| --- |
| **Script Snippet** |
| #help  Get-Help Add-IMSCustomClaim -ShowWindow  #this cmdlet can be called 4 ways all hinging on the -Type  Add-IMSCustomClaim -UserName user@domain.com -Type Role -Value "Bob"  Add-IMSCustomClaim -UserName user@domain.com -Type OrganizationType -Value "Jack Henry"  Add-IMSCustomClaim -UserName user@domain.com -Type ValidatedConsumerName -Value "Bob"  Add-IMSCustomClaim -UserName user@domain.com -Type ValidatedConsumerProduct -Value "Bob" |

### Basic Test-IAdapterPort Usage

This will take 3 parameters to test an iAdapter Port.

|  |
| --- |
| **Script Snippet** |
| #help  Get-Help Add-IAdapterPort -ShowWindow  #this cmdlet can be called 2 ways, useSSL defaults to False  Test-IAdapterPort -ServerName techserv.jhacorp.com -Port 40820  Test-IAdapterPort -ServerName techserv.jhacorp.com -Port 40820 -UseSSL $true |

### Advanced Test-IAdapterPort Usage

This will take all the parameter from Get-SGProvider to test an iAdapter Port.

|  |
| --- |
| **Script Snippet** |
| #help  Get-Help Add-IAdapterPort -ShowWindow  #this cmdlet can be called a few ways using other cmdlets  #this one gets all the institutions and asks for their providers, then sends them in to be tested if they are Silverlake or CIF2020.  Get-SGInstitution | Get-SGProvider -WarningAction SilentlyContinue | Test-IAdapterPort  #this one gets only the SilverLake institutions and asks for their SL provider, then sends them in to be tested if they are Silverlake.  Get-SGInstitution -FilterByProvider Silverlake | Get-SGProvider | Test-IAdapterPort  #this one gets only the CIF2020 institutions and asks for their 2020 provider, then sends them in to be tested if they are CIF2020.  Get-SGInstitution -FilterByProvider CIF2020 | Get-SGProvider | Test-IAdapterPort |