

Lessons with the PnP PowerShell

Woo Hoo!



Todd Klindt
15 Year SharePoint MVP

Writer, speaker, podcaster,
consultant at Sympraxis Consulting

todd.klindt@sympraxisconsulting.com
[@toddklindt](https://twitter.com/toddklindt)
www.toddklindt.com/sef2022



Agenda

- PowerShell Love
- Microsoft's Modules a bit (very little)
- Talk about PnP
- Look at some fun scripts



Environments

Why PowerShell Anyway?

- Repeatability
 - Consistency
 - Free Documentation
-
- Be the toolmaker, not the tool user



VS Code

- Not just for developers
- Free download
- PowerShell Extension
- Not a bad place
- Source Control
- Github & Azure Dev Ops
- In the browser
 - <https://vscode.dev>
 - <https://github1s.com/pnp/powershell>

The screenshot shows the Visual Studio Code interface. The main area displays a PowerShell script named `TKTools.psm1`. The script contains a function `Rename-TKDownloadVideo` which takes parameters for a video name and episode number, and renames the file based on whether it contains spaces or periods. Below the editor, the terminal window shows the PowerShell version (7.2.0) and a command being run: `PS C:\Users\todd_> Write-Host "¡Viva Las Vegas!"`, resulting in the output `¡Viva Las Vegas!`.

```
> TKTools.psm1 1
D: > Code > TKTools > > TKTools.psm1 > Rename-TKDownloadVideo {}

function Rename-TKDownloadVideo {
    [CmdletBinding(SupportsShouldProcess)] # to do: Add "confirm"
    param(
        [Parameter(ValueFromPipeline=$true)]
        [string]$VideoName,
        [switch]$IsEpisode,
        [switch]$UseSpace
    )

    Begin {}

    Process {
        # Write-Host $VideoName
        $filename = Get-ChildItem $VideoName -ErrorAction SilentlyContinue
        $isfile = (Test-Path $VideoName -PathType Leaf)

        if (($null -ne $filename) -and $isfile ) {

            # Whichever is greater, spaces or periods
            if (((($filename.BaseName.Split(" ")).count -gt ($filename.BaseName.Split(".").count)) -or ($useSpace)) {
                Write-Verbose "Using space"
                $nameparts = $filename.BaseName.Split(" ")
            } else {
                Write-Verbose "Using Period"
                $nameparts = $filename.BaseName.Split(".")
            }
            Write-Verbose "Parts = $($nameparts.count)"
        }
    }
}

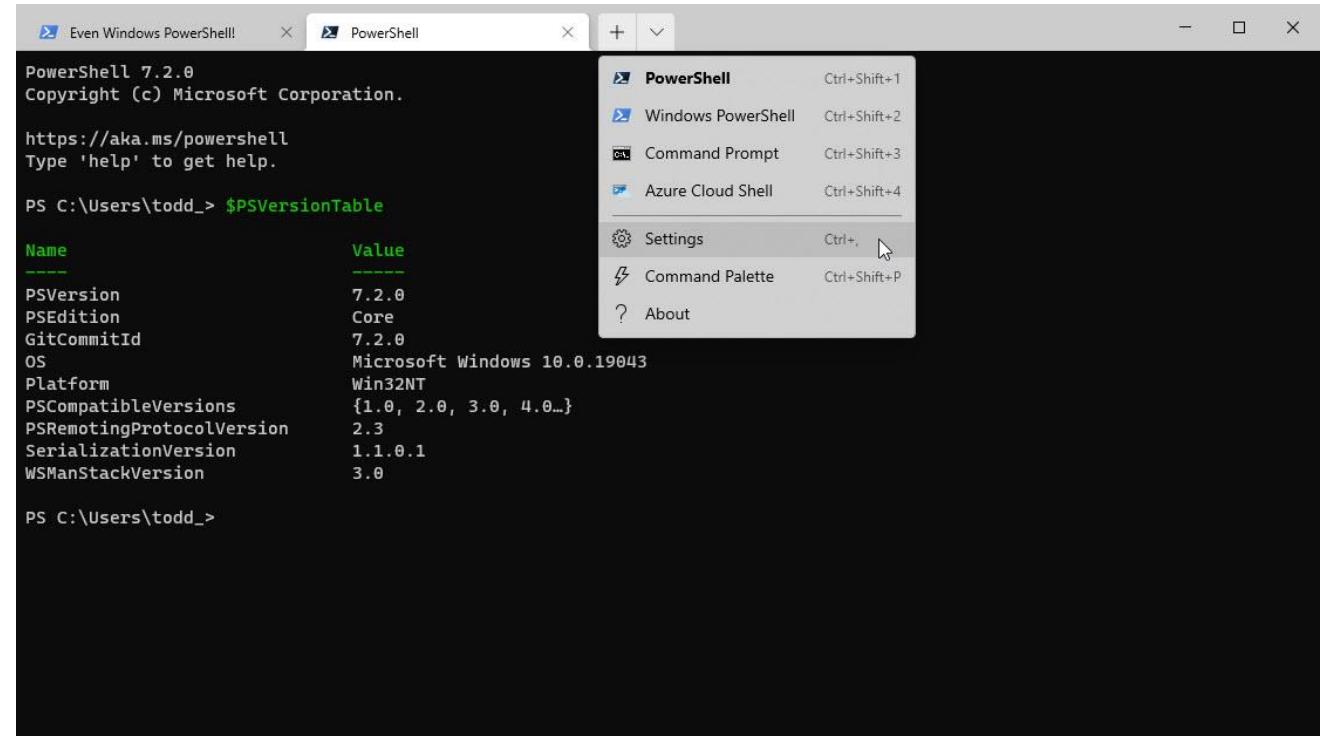
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL
PowerShell 7.2.0
Copyright (c) Microsoft Corporation.

https://aka.ms/powershell
Type 'help' to get help.

PS C:\Users\todd_> Write-Host "¡Viva Las Vegas!"
¡Viva Las Vegas!
PS C:\Users\todd_>
```

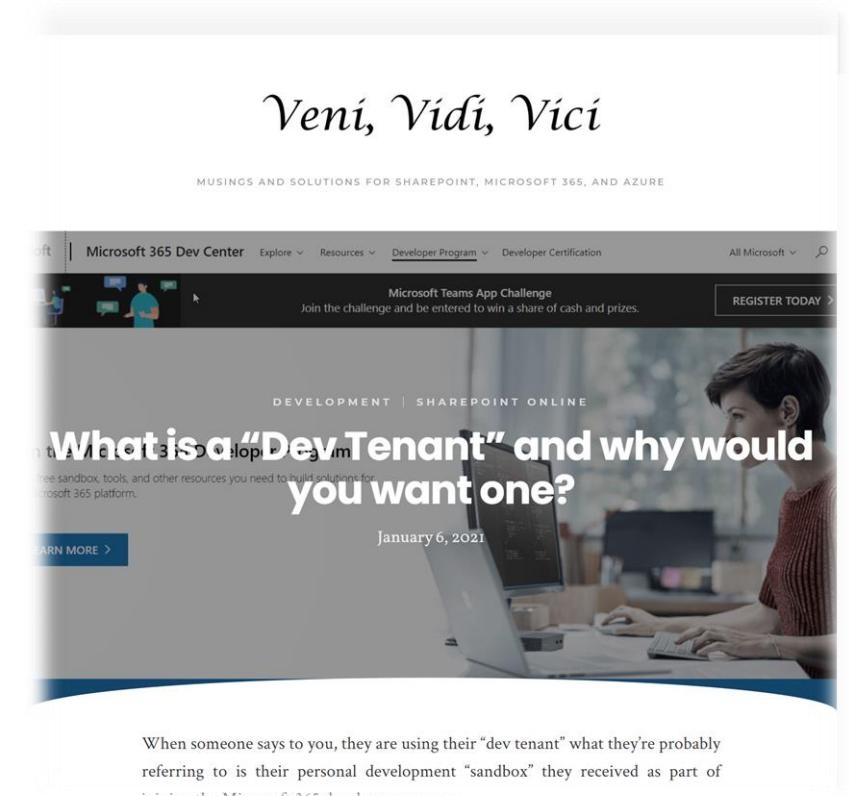
Windows Terminal

- The Future
- Windows Store
- Tabs
- <https://github.com/Microsoft/Terminal>



Get a Dev Tenant

- Great place to noodle around
- Free
- 100% less chance you'll get sacked
- <https://www.toddklindt.com/GetDevTenant>



When someone says to you, they are using their “dev tenant” what they’re probably referring to is their personal development “sandbox” they received as part of joining the Microsoft 365 developer program.

Let's talk about Passwords

- You will need your M365 username and password a lot
- Good and bad options
- Annoying but secure
 - \$Credentials = Get-Credential
- Less annoying and way, way less secure
 - \$username = admin@company.onmicrosoft.com
 - \$password = "RightHereInPlainText"
 - \$secure = \$password | ConvertTo-SecureString -AsPlainText –Force
 - \$Credentials = New-Object System.Management.Automation.PSCredential (\$username, \$secure)



Windows Credential Manager

- Built into Windows
- Let Microsoft worry about them securely
- Add-PnPStoredCredential
- Get-PnPStoredCredential

The screenshot displays two windows side-by-side. On the left is a PowerShell window titled 'Even Windows PowerShell!' showing command-line interactions:

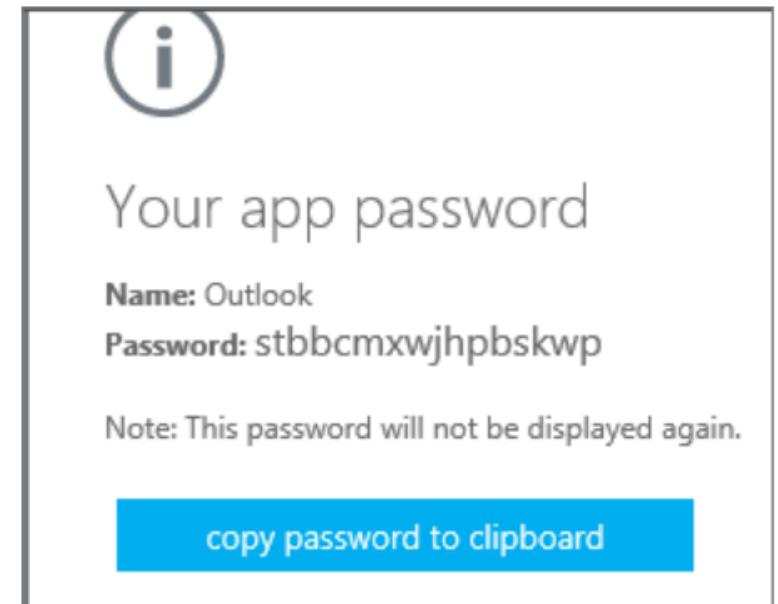
```
PS C:\Users\todd_> $Creds = Get-PnPStoredCredential -Name https://m365x995492-admin.sharepoint.com
PS C:\Users\todd_> Connect-SPOService -Url https://m365x995492-admin.sharepoint.com -Credential $Creds
PS C:\Users\todd_> Get-SPOSite
```

On the right is the Windows Credential Manager application window titled 'Credential Manager'. It shows a list of stored credentials under the 'Windows Credentials' tab. The columns are 'Url', 'Owner', and 'Storage Quota'. Most URLs begin with 'https://m365x995492.sharepoint.com/'. The 'Owner' column shows email addresses like 'admin@m365x995492.onmicrosoft.com' and 'adelev@m365x995492.onmicrosoft.com'. The 'Storage Quota' column shows values such as '26214400'.

Url	Owner	Storage Quota
https://m365x995492.sharepoint.com/search	admin@m365x995492.onmicrosoft.com	26214400
https://m365x995492.sharepoint.com/sites/STS3	admin@m365x995492.onmicrosoft.com	26214400
https://m365x995492.sharepoint.com/sites/RetailOperations	admin@m365x995492.onmicrosoft.com	26214400
https://m365x995492.sharepoint.com/sites/SalesAndMarketing	admin@m365x995492.onmicrosoft.com	26214400
https://m365x995492.sharepoint.com/sites/Petril	admin@m365x995492.onmicrosoft.com	26214400
https://m365x995492.sharepoint.com/sites/ContosoNews	admin@m365x995492.onmicrosoft.com	26214400
https://m365x995492.sharepoint.com/sites/livelycreditor8	admin@m365x995492.onmicrosoft.com	26214400
https://m365x995492-my.sharepoint.com/	admin@m365x995492.onmicrosoft.com	26214400
https://m365x995492.sharepoint.com/sites/ContosoBrand	admin@m365x995492.onmicrosoft.com	26214400
https://m365x995492.sharepoint.com/sites/USSales	admin@m365x995492.onmicrosoft.com	26214400
https://m365x995492.sharepoint.com/sites/Communications	admin@m365x995492.onmicrosoft.com	26214400
https://m365x995492.sharepoint.com/sites/Remoteliving	admin@m365x995492.onmicrosoft.com	26214400
https://m365x995492.sharepoint.com/sites/GlobalMarketing	admin@m365x995492.onmicrosoft.com	26214400
https://m365x995492.sharepoint.com/sites/operations	admin@m365x995492.onmicrosoft.com	26214400
https://m365x995492.sharepoint.com/sites/leadership-connection	admin@m365x995492.onmicrosoft.com	26214400
https://m365x995492.sharepoint.com/sites/benefits	admin@m365x995492.onmicrosoft.com	26214400
https://m365x995492.sharepoint.com/sites/delete21	adelev@m365x995492.onmicrosoft.com	26214400
https://m365x995492.sharepoint.com/sites/TestCommSite	adelev@m365x995492.onmicrosoft.com	26214400
https://m365x995492.sharepoint.com/sites/DigitalInitiativePublicRelations	adelev@m365x995492.onmicrosoft.com	26214400
https://m365x995492.sharepoint.com/sites/TestSite9	adelev@m365x995492.onmicrosoft.com	26214400
https://m365x995492.sharepoint.com/sites/contosoteam	adelev@m365x995492.onmicrosoft.com	26214400
https://m365x995492.sharepoint.com/sites/massivedepartment39	adelev@m365x995492.onmicrosoft.com	26214400
https://m365x995492.sharepoint.com/sites/SOCTeam	adelev@m365x995492.onmicrosoft.com	26214400
https://m365x995492.sharepoint.com/sites/office365adoption	adelev@m365x995492.onmicrosoft.com	26214400

A Quick Word about MFA

- Depends on your authentication provider
- Enable MFA at the tenant (AzureAD) level
- Create an app password for your account



Certificates

- Good for server to server authentication
- Better than storing username and password or secret
- Self signed cert for testing
- More on this later

Microsoft Modules

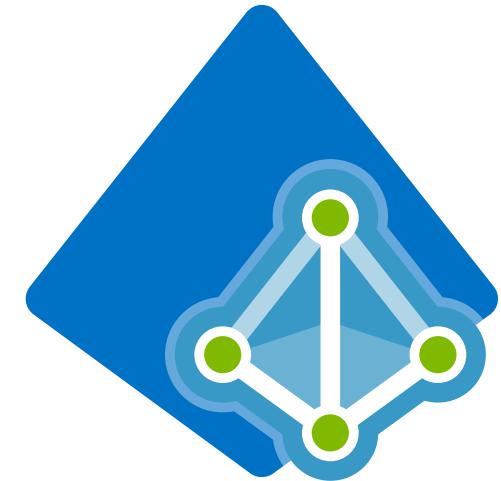
Connect to your Azure AD Tenant

- **MSOnline (Deprecated)**

```
$Credentials = Get-Credential  
Connect-MsolService -Credential $Credentials  
Get-MsolUser  
Get-Command -Module msonline
```

- **AzureAD (Net Deprecated Yet)**

```
$Credentials = Get-PnPStoredCredential -Name https://m365x995492-admin.sharepoint.com  
Connect-AzureAD -Credential $Credentials  
Get-AzureADUser  
Get-Command -Module AzureAD
```



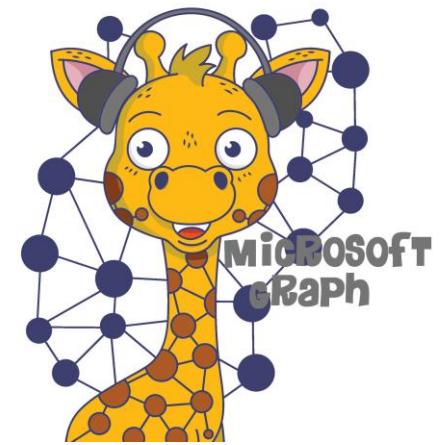
PowerShell with SharePoint Online

- Be prepared for disappointment
 - Getting better with age
- Allows basic manipulation of SharePoint Online
 - Users and groups
 - Tenants
 - Site Collections
 - Hub Sites
 - Multi-Geo
- `Install-Module -Name Microsoft.Online.SharePoint.PowerShell`



Graph SDK

- Got its fingers in everything
- Find project on Github
 - <https://github.com/microsoftgraph/msgraph-sdk-powershell>
- Not for the faint of heart
- Cmdlets mostly generated by AutoREST
- Future of AzureAD module
- More on this later



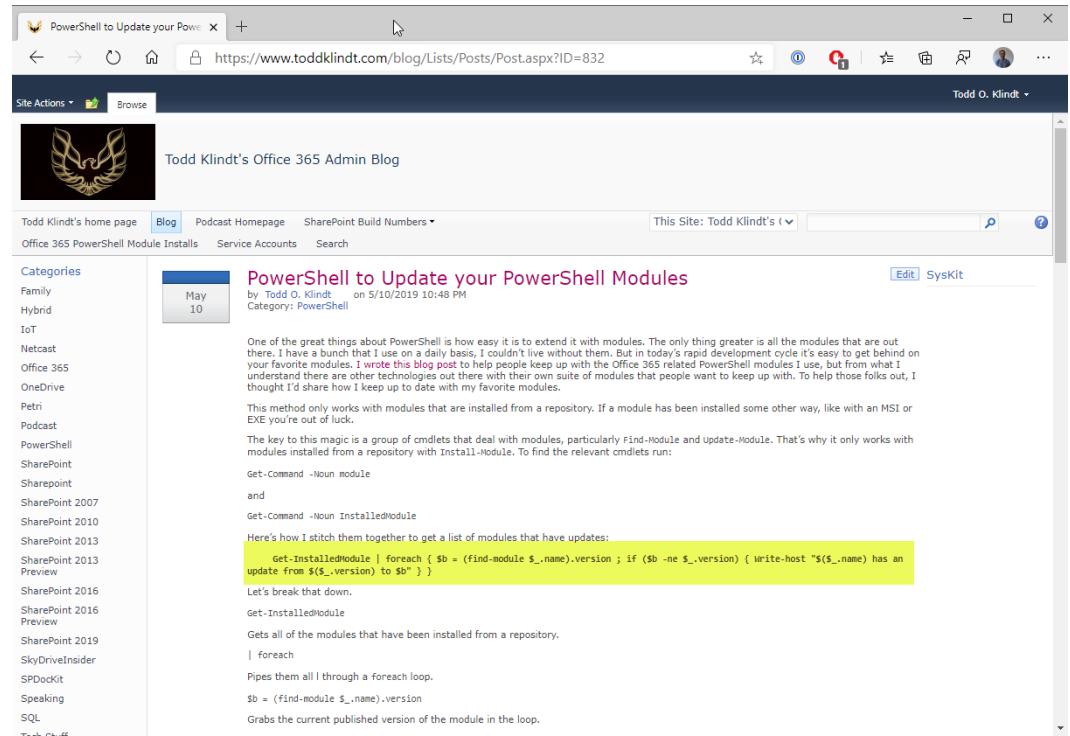
Office 365 / Microsoft 365 / Unified Groups

- No Groups module
- MSOL / Azure
- Exchange
- Teams
- PnP PowerShell
- Graph SDK



Updating the Modules

- Find-Module to get current version
- Get-InstalledModule
- Get-Module -ListAvailable
- Update-Module Microsoft.Online.SharePoint.PowerShell
- Update-Module msonline
- Update-Module AzureAD
- Update-Module PnP.PowerShell
- <https://www.toddklindt.com/PoshUpdateModules>



PnP PowerShell!!!

Patterns and Practices PowerShell (Phew!)

- GitHub
 - <https://github.com/pnp/powershell>
- Adds 620+ more cmdlets
- Scoped at Site Collection
- Install-Module PnP.PowerShell
- Get-Command -Module PnP.PowerShell
- Use SharePointPnPPowerShellOnline for on-prem SharePoint

CommandType	Name	Version	Source
Alias	Ensure-PnPFolder	2.24.18...	SharePointPnPPowershellonline
Alias	Execute-PnQuery	2.24.18...	SharePointPnPPowershellonline
Alias	Load-PnProvisioningTemplate	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPApp	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPClientsidePage	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPClientsidePagesection	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPClientsideText	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPClientsidewebPart	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPContentType	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPContentTypeToDocumentSet	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPContentTypeToList	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPCustomAction	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPDataRowstoProvisioningTemplate	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPDocumentSet	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPEventReceiver	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPField	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPFieldFromXml	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPFieldToContentType	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPFile	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPFileToProvisioningTemplate	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPFolder	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPHtmlPublishingPageLayout	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPIndexedProperty	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPJavaScriptBlock	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPJavaScriptLink	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPListFoldersToProvisioningTemplate	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPListItem	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPMasterPage	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPNavigationNode	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPOffice365GroupToSite	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPublishingImageRendition	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPPublishingPage	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPublishingPageLayout	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPRoleDefinition	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPsiteClassification	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPsiteCollectionAdmin	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPsiteCollectionAppCatalog	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPsiteDesign	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPsiteScript	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PNPSavedCredential	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPTaxonomyField	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPTenantCdnOrigin	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPTenantTheme	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPUserToGroup	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPView	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPWebhookSubscription	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPWebPartToWebPartPage	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPWebPartToWikiPage	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPWikiPage	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPWorkflowDefinition	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Add-PnPWorkflowSubscription	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Apply-PnPProvisioningTemplate	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Clear-PnPListItemAsRecord	2.24.18...	SharePointPnPPowershellonline
Cmdlet	Get-PnPWorkflow	2.24.18...	SharePointPnPPowershellonline

Favorites

- PnPFile
 - Add-PnPFile, Copy-PnPFile
 - Find-PnPFile, Get-PnPFile
 - Move-PnPFile, Remove-PnPFile
 - Rename-PnPFile
 - Set-PnPFileCheckedIn
 - Set-PnPFileCheckedOut
- PnPList
 - Add, Get, Set, Remove
- Get-PnPListItem
 - Set-PnPListItem
- Get-PnPProperty
- Set-PnPGroupPermissions
- Add-PnPView
- Get-PnPField
- Restore-PnPRecycleBinItem
- Office 365 Groups
 - New-PnPMicrosoft365Group, Remove-PnPMicrosoft365Group
 - Get-PnPMicrosoft365Group, Set-PnPMicrosoft365Group
 - Get-PnPMicrosoft365GroupMembers
 - Get-PnPMicrosoft365GroupOwners

PnP Authentication

```
Add-PnPSStoredCredential -Name "https://tenant.sharepoint.com" -  
Username yourname@tenant.onmicrosoft.com
```

```
Connect-PnPOnline -Url "https://tenant.sharepoint.com"
```

```
Connect-PnPOnline -Url "https://tenant.sharepoint.com/sites/hr"
```

```
$Credentials = Get-PnPSStoredCredential -Name  
"https://tenant.sharepoint.com"
```

```
Connect-SPOSERVICE -Url https://tenant-admin.sharepoint.com -  
Credentials $ Credentials
```

But my Boss HATES PnP PowerShell!

- Your boss is misinformed 😊
- Vesa Juvonen, Senior Program Manager from SharePoint Engineering and MCM for SharePoint, is one of the main project owners
- Is scanned the same as any PowerShell Gallery Module (not at all)
- Erwin van Hunen used to work at RenCore
 - Exceptions are approved by SharePoint Engineering team
- Signed with Microsoft's key starting November 2017
- Uses the same API as web parts and other SharePoint code
- It's Open Source
- Respects SharePoint security
 - Can be more secure, as it can be more fine grained
- PnP PowerShell hits the Office 365 API a billion times a month

PnP Vs. SPO

- SPO
- Tenant Wide tasks
- Basic tasks
- New functionality that hasn't made it into SharePoint Admin Center
- New functionality that hasn't made it into PnP yet
- PNP
- Everything SPO can do
- Tasks by site owners
- Tasks by users
- Site specific settings
- More specific tasks
- Want to be cooler
- Working on an object smaller than a site
- Working on an object other than a site

Azure Function

- Great serverless automation
- Can be trigger from Flow
 - Http
 - Azure Queue
- PowerShell 7 environment (PnP)
- Use Certificate authentication
- Can publish with VS Code
- Use Console to troubleshoot
- Turn on logging to save console



Multiple versions of the same Module?

- `Find-Module -AllowPrerelease -AllVersions`
- `Install-Module -RequiredVersion`
- `Install-Module -AllowClobber`
- `Import-Module -Name PnP.PowerShell -RequiredVersion 1.9.0 -Prefix Old`

Examples

(Show them how cool VS Code is)

Upload a File

```
$web = https://tenant.sharepoint.com/sites/hr
$folder = "Shared Documents"
Connect-PnPOnline -Url $web
Add-PnPFile -Path '.\Boot fairs with Graphic design.docx'
-Folder $folder
```

Add a Folder

```
Add-PnPFolder -Name "Folder 1" -Folder $folder
```

```
Add-PnPFile -Path '.\Building materials licences to budget  
for Storytelling.docx' -Folder '$folder\Folder 1'
```

Get Internal Shared Files

```
$docliblist = Get-PnPList -Includes DefaultViewUrl,IsSystemList | Where-Object -Property IsSystemList -EQ -Value $false | Where-Object -Property BaseType -EQ -Value "DocumentLibrary"
Foreach ($doclib in $docliblist)
{
    $doclist = Get-PnPListItem -List $DocLib
    foreach ($doc in $doclist) {
        if (((($doc.FieldValues).SharedWithUsers) -ne $null) {
            foreach ($user in (((($doc.FieldValues).SharedWithUsers))) {
                Write-Output "$((($doc.FieldValues).FileRef) - $($user.email))"
            }
        }
    }
}
```

Get Extended File Info

```
$docliblist = Get-PnPList -Includes DefaultViewUrl,IsSystemList | Where-Object -Property IsSystemList -EQ -Value $false | Where-Object -Property BaseType -EQ -Value "DocumentLibrary"

Foreach ($doclib in $docliblist)
{
    $doclibTitle = $doclib.Title
    $doclist = Get-PnPListItem -List $DocLib
    $doclist | ForEach-Object { Get-PnPProperty -ClientObject $_ -Property File, ContentType, ComplianceInfo}
    foreach ($doc in $doclist) {
        [pscustomobject]@{Library= $doclibTitle;Filename = ($doc.File).Name;ContentType = ($doc.ContentType).Name;Label = ($doc.ComplianceInfo).ComplianceTag}
    }
}
```

Bulk Undelete Files

```
Connect-PnPOnline -Url https://sadtenant.sharepoint.com/ -Credentials  
SadTenantAdmin  
$bin = Get-PnPRecycleBinItem | Where-Object -Property Leafname -Like -Value  
"*.jpg" | Where-Object -Property Dirname -Like -Value "Important Photos/Shared  
Documents/*" | Where-Object -Property DeletedByEmail -EQ -Value  
baduser@sadtenant.phooey  
$bin.count  
$bin | foreach -begin { $a = 0} -Process {Write-Host "$a - $($_.LeafName)" ; $_  
| Restore-PnPRecycleBinItem -Force ; $a++ } -End { Get-Date }  
($bin[20001..30000]) | foreach -begin { $a = 0} -Process {Write-Host "$a -  
$($_.LeafName)" ; $_  
| Restore-PnPRecycleBinItem -Force ; $a++ } -End { Get-Date }  
  
# https://www.toddklindt.com/PoshRestoreSPOFiles
```

New Site and Group

- New-SPOSite
 - No Group, No Team, No Bueno
 - Can be Groupified later
- New-PnPsite -Type TeamSite -Title "Modern Team Site" - Alias ModernTeamSite –IsPublic
 - Group, No Team
 - Can be Teamified later

The Whole Enchilada

- New-Team -DisplayName "Fancy Group" -
Description "Fancy Group made by PowerShell?" -
Alias FancyGroup -AccessType Public
 - There is no later!

“Save Site as Template”

- `Get-PnPSiteTemplate -Out customer.xml`
- `Add-PnPListFoldersToSiteTemplate -Path customer.xml -List 'Data Storage' -Recursive`
- `Invoke-PnPSiteTemplate -Path customer.xml -Handlers Lists, SiteSecurity`
- `Get-Command -Module PnP.PowerShell -Name *temp*`

Get All the Flows

Add-PowerAppsAccount

```
Get-AdminFlow | ForEach-Object { $ownername = (Get-MsolUser -ObjectId  
$_.CreatedBy.userId).DisplayName ; $owneremail = (Get-MsolUser -ObjectId  
$_.CreatedBy.userId).UserPrincipalName ; Write-Host $_.DisplayName, $ownername,  
$owneremail }
```

- Requires MSOL module and connection

Get Your Flows

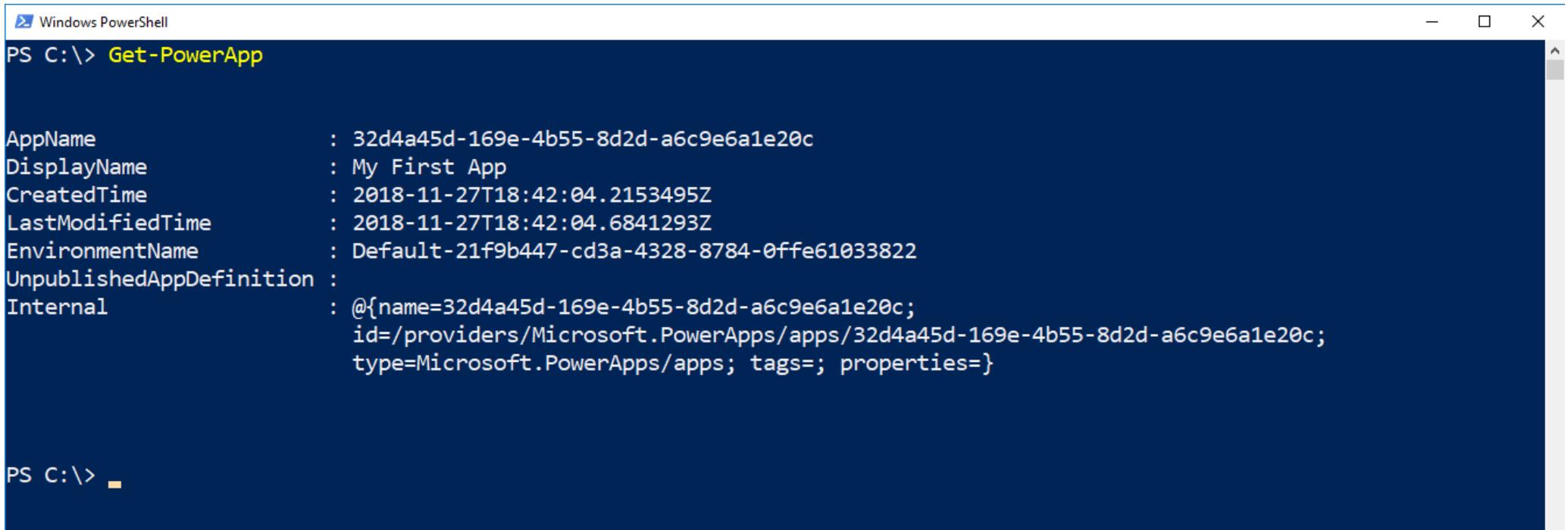
```
Windows PowerShell
PS C:\> Get-Flow

FlowName      : ddf63e93-3f0a-4a78-951e-46d2fa7eec63
Enabled       : True
DisplayName   : Send an email when a new item is created in SharePoint.
UserType      : Owner
CreatedTime   : 2018-11-27T18:25:32.6267962Z
LastModifiedTime : 2018-11-27T18:25:32.9141198Z
EnvironmentName : Default-21f9b447-cd3a-4328-8784-0ffe61033822
Internal      : @{name=ddf63e93-3f0a-4a78-951e-46d2fa7eec63; id=/providers/Microsoft.ProcessSimple/environments/Default-21f9b447-cd3a-4328-8784-0ffe61033822/flows/ddf63e93-3f0a-4a78-951e-46d2fa7eec63; type=Microsoft.ProcessSimple/environments/flows; properties=}

PS C:\>
```

And Power Apps Apps

- Get-PowerApp



A screenshot of a Windows PowerShell window titled "Windows PowerShell". The command "Get-PowerApp" is run at the prompt, and its output is displayed. The output shows various properties of a Power App, such as AppName, DisplayName, CreatedTime, LastModifiedTime, EnvironmentName, UnpublishedAppDefinition, and Internal. The Internal property contains a JSON object describing the app's definition.

```
PS C:\> Get-PowerApp

AppName          : 32d4a45d-169e-4b55-8d2d-a6c9e6a1e20c
DisplayName      : My First App
CreatedTime      : 2018-11-27T18:42:04.2153495Z
LastModifiedTime : 2018-11-27T18:42:04.6841293Z
EnvironmentName   : Default-21f9b447-cd3a-4328-8784-0ffe61033822
UnpublishedAppDefinition :
Internal         : @{name=32d4a45d-169e-4b55-8d2d-a6c9e6a1e20c;
                   id=/providers/Microsoft.PowerApps/apps/32d4a45d-169e-4b55-8d2d-a6c9e6a1e20c;
                   type=Microsoft.PowerApps/apps; tags=; properties=}

PS C:\>
```

Disable Flow Button

- SPO Method
 - Connect-SPOSERVICE -Url https://flowhater-admin.sharepoint.com
 - \$val = [Microsoft.Online.SharePoint.TenantAdministration.FlowsPolicy]::Disabled
 - Set-SPOSITE -Identity https://flowhater.sharepoint.com/sites/SadSite -DisableFlows \$val
- PnP Method
 - Connect-PnPOnline -Url https://flowhater.sharepoint.com/sites/SadSite -Credentials 'MeanSiteAdmin'
 - Set-PnPSite -DisableFlows:\$true

Fun Links to Click

- <https://www.toddklindt.com/posh>
- <https://www.sharepointdiary.com/>
- <https://office365itpros.com/>
- <https://aka.ms/m365pnp>
- <https://aka.ms/m365/devprogram>





Questions?



todd.klindt@sympraxisconsulting.com
@toddklindt
www.toddklindt.com
www.toddklindt.com/sef2022