

SharePoint Online Administration with PowerShell

Technical Track
Brendan Griffin

- ▶ **Brendan Griffin** — @brendankarl | brendan-griffin.com
 - Senior Premier Field Engineer @ Microsoft
 - Microsoft Certified Master – SharePoint 2010
 - SharePoint-er since 2004

Agenda

- ▶ The “Challenge”
- ▶ The Solution
- ▶ Drawbacks
- ▶ CSOM to the Rescue!

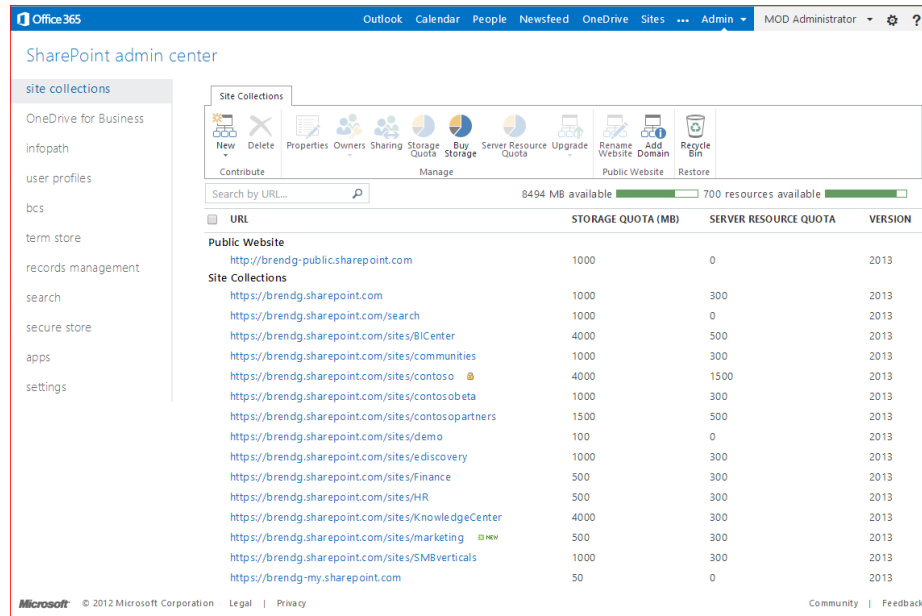
The Challenge

- ▶ SharePoint administrators rely heavily on PowerShell
- ▶ Office 365 is a \$1.5 billion a year business and growing
- ▶ What does the increased adoption of Office 365 mean for traditional SharePoint administrators?



The Challenge

- ▶ Does **No Server Access** == **No PowerShell**?
- ▶ Are we back in the dark ages of GUI based administration?



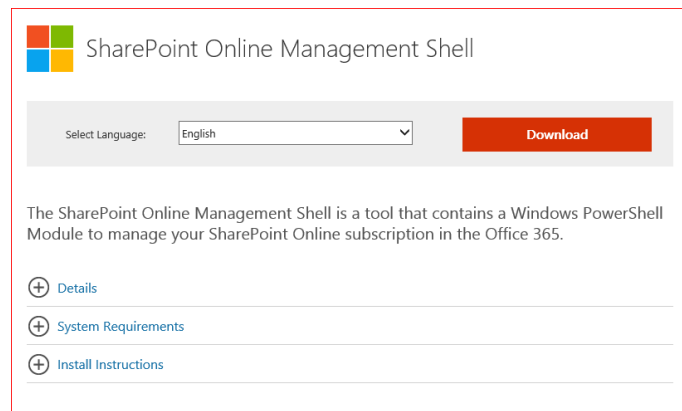
The screenshot displays the SharePoint Admin Center interface. The top navigation bar includes links for Outlook, Calendar, People, Newsfeed, OneDrive, Sites, and Admin. The left sidebar lists various site collections including OneDrive for Business, infopath, user profiles, bcs, term store, records management, search, secure store, apps, and settings. The main content area shows a table of site collections with columns for URL, Storage Quota (MB), Server Resource Quota, and Version. The table lists various site collections under the 'Public Website' and 'Site Collections' categories, including URLs like http://brendg-public.sharepoint.com and http://brendg.sharepoint.com/sites/communities. A progress bar at the top indicates 8494 MB available and 700 resources available.

URL	STORAGE QUOTA (MB)	SERVER RESOURCE QUOTA	VERSION
Public Website			
http://brendg-public.sharepoint.com	1000	0	2013
Site Collections			
https://brendg.sharepoint.com	1000	300	2013
https://brendg.sharepoint.com/search	1000	0	2013
https://brendg.sharepoint.com/sites/BICenter	4000	500	2013
https://brendg.sharepoint.com/sites/communities	1000	300	2013
https://brendg.sharepoint.com/sites/contoso	4000	1500	2013
https://brendg.sharepoint.com/sites/contosobeta	1000	300	2013
https://brendg.sharepoint.com/sites/contosopartners	1500	500	2013
https://brendg.sharepoint.com/sites/demo	100	0	2013
https://brendg.sharepoint.com/sites/ediscovery	1000	300	2013
https://brendg.sharepoint.com/sites/Finance	500	300	2013
https://brendg.sharepoint.com/sites/HR	500	300	2013
https://brendg.sharepoint.com/sites/KnowledgeCenter	4000	300	2013
https://brendg.sharepoint.com/sites/marketing	500	300	2013
https://brendg.sharepoint.com/sites/SMBverticals	1000	300	2013
http://brendg-my.sharepoint.com	50	0	2013



The Solution

- ▶ **Of course not!**
- ▶ A number of PowerShell Cmdlets are available for SharePoint Online
- ▶ Download the **SharePoint Online Management Shell** to get started [here](#)



SharePoint Online Management Shell

- ▶ So, I've installed it – what next?
- ▶ Let's take a closer look:



SharePoint Online Management Shell

- ▶ Focused on Site Collection and User administration tasks



SharePoint Online Management Shell

- Includes 30 Cmdlets, which is slightly less than On-Premises! [Link](#)

```
SharePoint Online Cmdlets Count.ps1 X
1 Import-Module Microsoft.Online.SharePoint.PowerShell -DisableNameChecking
2 $SPOL = Get-Module -Name "Microsoft.Online.SharePoint.PowerShell"
3 Write-Host "SharePoint Online has" $SPOL.ExportedCmdlets.Count "Cmdlets" -ForegroundColor Green

PS C:\> C:\Users\brendg\Desktop\SharePoint Online Cmdlets Count.ps1
SharePoint Online has 30 Cmdlets
```

```
SharePoint Cmdlets Count.ps1 X
1 asnp *SharePoint* -ErrorAction SilentlyContinue
2 $SP = Get-Command -Module Microsoft.SharePoint.PowerShell
3 Write-Host "SharePoint has" $SP.Count "Cmdlets" -ForegroundColor Green

PS C:\> C:\Users\SharePoint Cmdlets Count.ps1
SharePoint has 779 Cmdlets
```


O365 vs. On Premises Cmdlets



- ▶ Get-SPOSite
- ▶ Set-SPOSite
- ▶ New-SPOSite
- ▶ Remove-SPOSite
- ▶ New-SPOUser
- ▶ ...

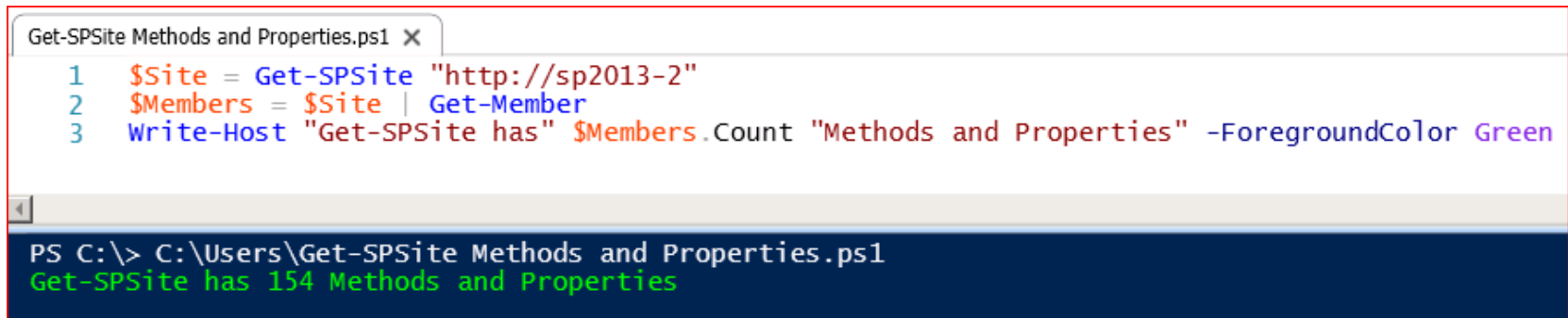


- ▶ Get-SPSite
- ▶ Set-SPSite
- ▶ New-SPSite
- ▶ Remove-SPSite
- ▶ New-SPUser
- ▶ ...

- ▶ **Spot the difference?**

O365 vs. On Premises Cmdlets

- ▶ “I will only need to make minor changes to my scripts” – *Brendan Griffin (2013)*
 - **Wrong!!!** – Cmdlets aren’t functionally equivalent
 - Have different Methods and Properties
 - In some cases behave differently

A screenshot of a PowerShell console window. The title bar reads "Get-SPSite Methods and Properties.ps1 X". The script content is as follows:

```
1 $Site = Get-SPSite "http://sp2013-2"  
2 $Members = $Site | Get-Member  
3 Write-Host "Get-SPSite has" $Members.Count "Methods and Properties" -ForegroundColor Green
```

The console output at the bottom shows the command being executed: `PS C:\> C:\Users\Get-SPSite Methods and Properties.ps1`, followed by the result: `Get-SPSite has 154 Methods and Properties` in green text.

O365 vs. On Premises Cmdlets

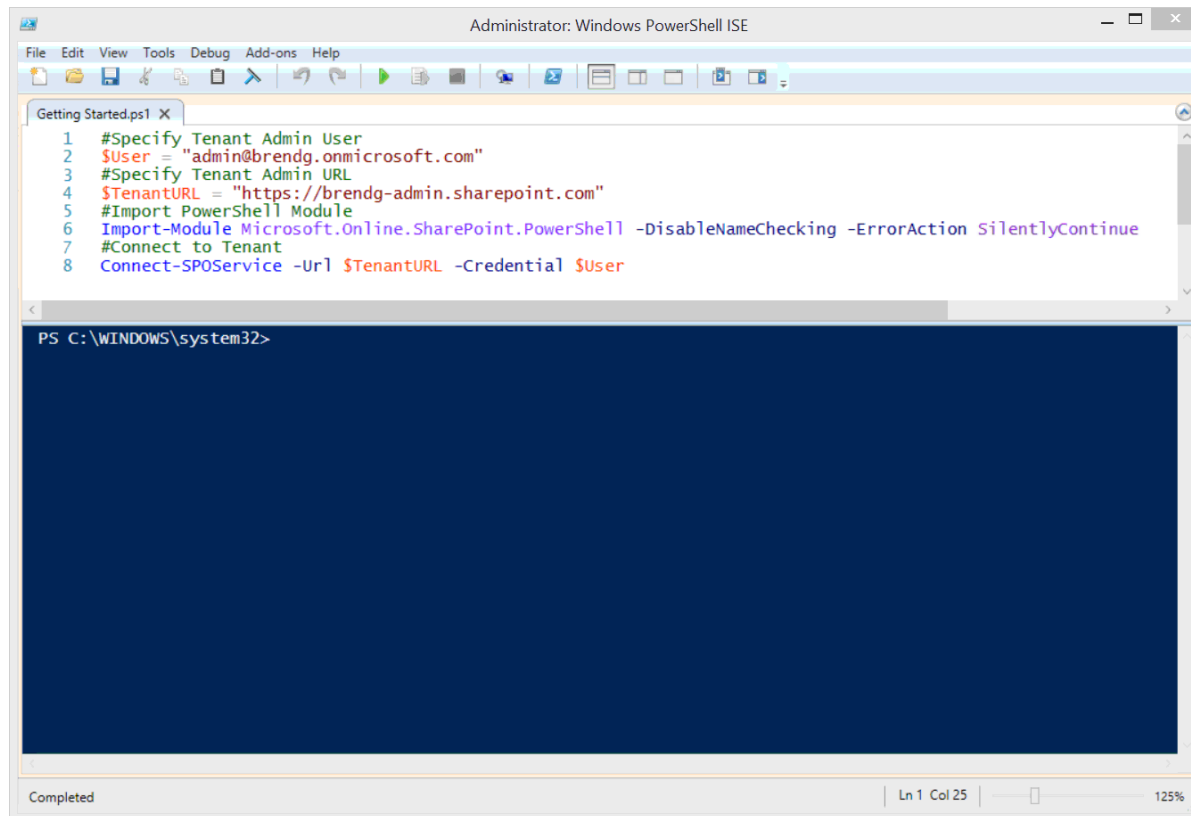
► Example:

- **Get-SPSite** returns all Site Collections
 - **Get-SPOSite** returns all Site Collections except My Sites
- In some cases limitations can be addressed using a different approach – more on this later...
- Let's take a look at some of the Cmdlets in action...



SPOL Management Shell – Getting Started

- ▶ Import the **Microsoft.Online.SharePoint.PowerShell** Module
- ▶ Use **Connect-SPOService** to connect to the Tenant

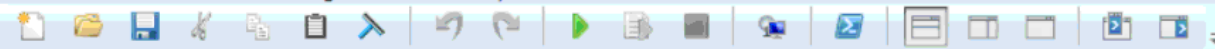


The screenshot shows the Windows PowerShell ISE interface. The title bar reads "Administrator: Windows PowerShell ISE". The menu bar includes File, Edit, View, Tools, Debug, Add-ons, and Help. The toolbar contains various icons for file operations and execution. The script editor shows a file named "Getting Started.ps1" with the following code:

```
1 #Specify Tenant Admin User
2 $User = "admin@brendg.onmicrosoft.com"
3 #Specify Tenant Admin URL
4 $TenantURL = "https://brendg-admin.sharepoint.com"
5 #Import PowerShell Module
6 Import-Module Microsoft.Online.SharePoint.PowerShell -DisableNameChecking -ErrorAction SilentlyContinue
7 #Connect to Tenant
8 Connect-SPOService -Url $TenantURL -Credential $User
```

The console window below the script editor shows the prompt "PS C:\WINDOWS\system32>" and is currently empty. The status bar at the bottom indicates "Completed", "Ln 1 Col 25", and a zoom level of "125%".

File Edit View Tools Debug Add-ons Help



Getting Started.ps1 X

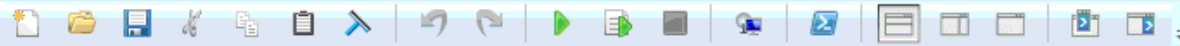
```
1 #Specify Tenant Admin User
2 $User = "admin@brendg.onmicrosoft.com"
3 #Specify Tenant Admin URL
4 $TenantURL = "https://brendg-admin.sharepoint.com"
5 #Import PowerShell Module
6 Import-Module Microsoft.Online.SharePoint.PowerShell -DisableNameChecking -ErrorAction SilentlyContinue
7 #Connect to Tenant
8 Connect-SPOService -Url $TenantURL -Credential $User
```

PS C:\WINDOWS\system32>

Completed

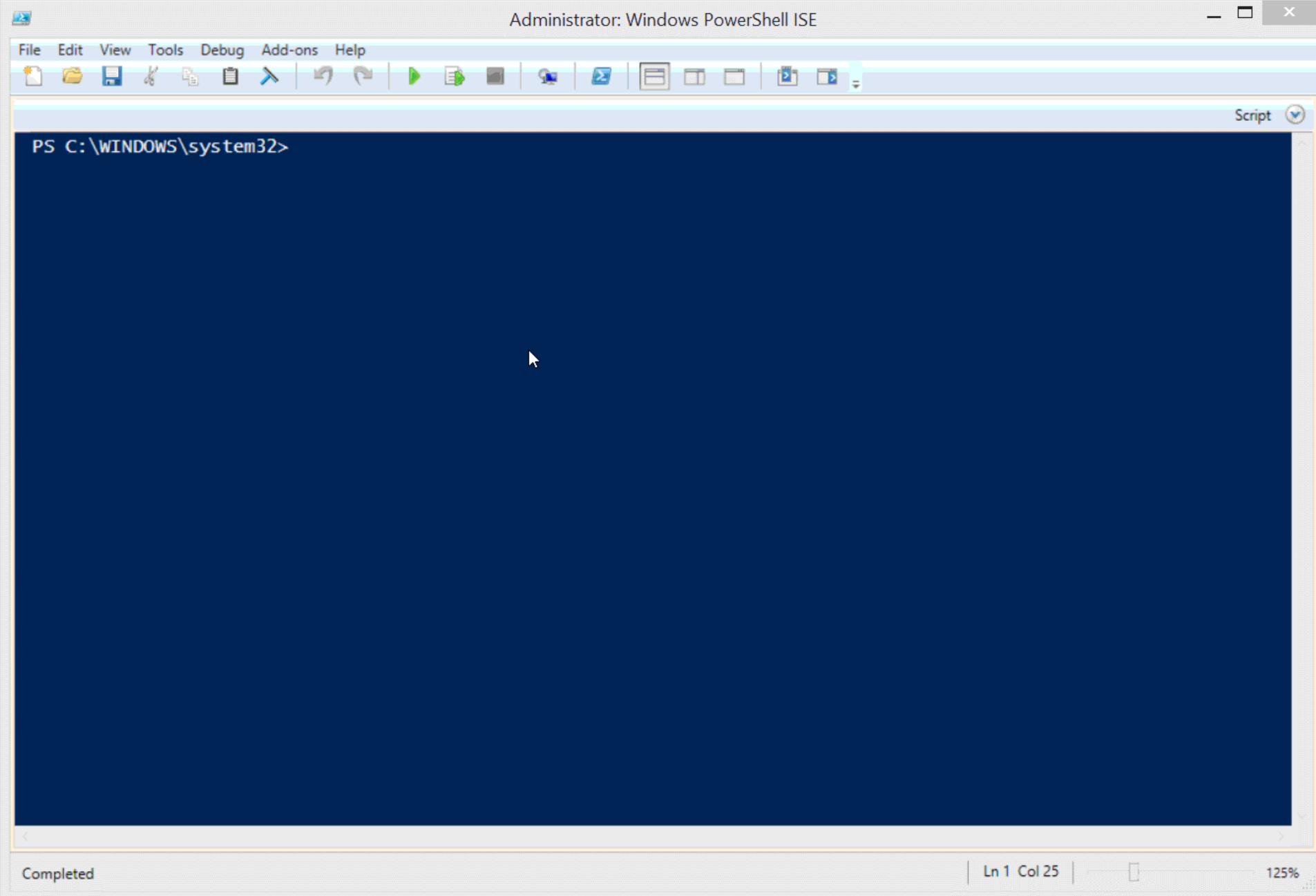
Ln 1 Col 25

125%



```
PS C:\Windows\System32\WindowsPowerShell\v1.0>
```





Get-SPOSite Summary



- ▶ Use the **–Detailed** parameter with **Get-SPOSite** if you need the following properties returned
 - LockIssue
 - WebsCount
 - CompatibilityLevel
 - AllowSelfServiceUpgrade
 - ResourceUsageCurrent
 - ResourceUsageAverage
 - StorageUsageCurrent

What Sites Do I Have?

- Inventory all Site Collections and output results to a CSV file using **Get-SPOSite**

SPO Site Inventory.ps1 X

```
1 #Retrieve basic information about each Site Collection (excluding My Sites)
2 Get-SPOSite -Detailed | Export-CSV -LiteralPath SiteInventory.csv -NoTypeInfo
```

LastContentModifiedDate	StorageUsageCurrent	WebsCount	Url	Owner	Template	Title	SharingCapability
03/06/2014 09:39	128	1	http://brendg-public.sharepoint.com/	admin@brendg.onmicrosoft.com	BLANKINTERNET#0	Contoso Bistro	ExternalUserAndGuestSharing
05/06/2014 10:03	71	2	https://brendg.sharepoint.com/	admin@brendg.onmicrosoft.com	STS#0	Contoso Team Site	ExternalUserAndGuestSharing
05/06/2014 10:03	25	1	https://brendg.sharepoint.com/search	admin@brendg.onmicrosoft.com	SRCHCEN#0		Disabled
05/06/2014 10:03	141	1	https://brendg.sharepoint.com/sites/BICenter	admin@brendg.onmicrosoft.com	BICENTERSITE#0	BI	Disabled
03/06/2014 09:37	1	1	https://brendg.sharepoint.com/sites/communities	admin@brendg.onmicrosoft.com	COMMUNITYPORTAL#0	Communities	Disabled
05/06/2014 10:03	1187	65	https://brendg.sharepoint.com/sites/contoso	admin@brendg.onmicrosoft.com	BLANKINTERNET#0	Home	Disabled
03/06/2014 10:19	2	1	https://brendg.sharepoint.com/sites/contosobeta	admin@brendg.onmicrosoft.com	STS#0	Contoso Beta	Disabled
03/06/2014 10:20	73	9	https://brendg.sharepoint.com/sites/contosopartners	admin@brendg.onmicrosoft.com	BLANKINTERNET#0	Contoso Partner Portal	Disabled
03/06/2014 13:38	2	1	https://brendg.sharepoint.com/sites/demo	admin@brendg.onmicrosoft.com	STS#0	Demo Site	Disabled
03/06/2014 09:39	5	8	https://brendg.sharepoint.com/sites/ediscovery	admin@brendg.onmicrosoft.com	EDISC#0	Legal Discovery Center	Disabled
04/06/2014 09:56	2	1	https://brendg.sharepoint.com/sites/Finance	admin@brendg.onmicrosoft.com	STS#0	Finance	ExternalUserAndGuestSharing
04/06/2014 09:56	2	1	https://brendg.sharepoint.com/sites/HR	admin@brendg.onmicrosoft.com	STS#0	HR	ExternalUserSharingOnly
03/06/2014 10:19	200	1	https://brendg.sharepoint.com/sites/KnowledgeCenter	admin@brendg.onmicrosoft.com	BLANKINTERNET#0	Knowledge Center	Disabled
04/06/2014 16:53	2	1	https://brendg.sharepoint.com/sites/marketing	robinc@brendg.onmicrosoft.com	STS#0	Marketing	Disabled
03/06/2014 10:19	95	11	https://brendg.sharepoint.com/sites/SMBverticals	admin@brendg.onmicrosoft.com	STS#0	MVP Demos	Disabled
03/06/2014 09:39	4	1	https://brendg-my.sharepoint.com/	admin@brendg.onmicrosoft.com	SPSMSITEHOST#0		Disabled

Who Has Access?

- List the SharePoint Groups that each user is a member of within a Site Collection using **Get-SPOUser**

```
SPO Site Users.ps1 X
1 $URL = "https://brendg.sharepoint.com/sites/HR"
2 Get-SPOUser -Limit All -Site $URL | Select DisplayName, LoginName, Groups
```

DisplayName	LoginName	Groups
Alex Darrow	alex.d@brendg.onmicrosoft.com	{HR Owners}
Brendan Griffin	live.com#brendankgriffin@hotmail.com	{HR Members}
Everyone	true	{}
Everyone except external users	spo-grid-all-users/391f79f7-db...	{}
Kari Furse	karif@brendg.onmicrosoft.com	{}
MOD Administrator	admin@brendg.onmicrosoft.com	{}
System Account	SHAREPOINT\system	{HR Owners}
YL0001_spocrawler_43_3936	yl0001_spocrawler_43_3936	{}

Site Level Administration

► **Set-SPOSite** can be used to:

- Change Site Owner

```
1 $URL = "https://brendg.sharepoint.com/sites/Marketing"  
2 Set-SPOSite -Identity $URL -Owner "RobinC@brendg.onmicrosoft.com"
```

- Configure the Storage and Resource Quota

```
1 Set-SPOSite -Identity $URL -StorageQuota 1000 -ResourceQuota 500
```

- Change the Title

```
1 Set-SPOSite $URL -Title "New Title"
```

Site Level Administration

► **Set-SPOSite** can also be used to:

- Lock a Site

```
1 Set-SPOSite -Identity $URL -LockState NoAccess
```

- Requests to Sites that are locked can be redirected to a specific URL

```
1 Write-Host (Get-SPOTenant).NoAccessRedirectURL -ForegroundColor Green  
https://brendg.sharepoint.com/SitePages/Site is Locked.aspx
```

- Redirect URL is configurable using
Set-SPOTenant -NoAccessRedirectURL

Tenant Level Settings

- **Get-SPOTenant** can be used to retrieve Tenant level settings

```
1 Get-SPOTenant
```

```
StorageQuota           : 30644
StorageQuotaAllocated  : 22650
ResourceQuota          : 5900
ResourceQuotaAllocated : 5400
CompatibilityRange     : 15,15
ExternalServicesEnabled : True
NoAccessRedirectUrl    : https://brendg.sharepoint.com/SitePages/Site is Locked.aspx
SharingCapability       : ExternalUserAndGuestSharing
DisplayStartASiteOption : True
StartASiteFormUrl      :
```

Configuring External Access

- ▶ Configure the External Sharing Capability for a specific Site Collection
- ▶ Use **Set-SPOSite** with the **SharingCapability** parameter, options:
 - *Disabled*
 - *ExternalUserAndGuestSharing*
 - *ExternalUserSharingOnly*

Configure-ExternalSharing.ps1* X

```
1 #No Sharing
2 Set-SPOSite -Identity https://brendg.sharepoint.com/sites/IT -SharingCapability Disabled
3 #External and Guest
4 Set-SPOSite -Identity https://brendg.sharepoint.com/sites/Finance -SharingCapability ExternalUserAndGuestSharing
5 #External Only
6 Set-SPOSite -Identity https://brendg.sharepoint.com/sites/HR -SharingCapability ExternalUserSharingOnly
```

Auditing External Access

- Who has access? Where do they have access? Who invited them?

```
ExternalSharingReport.ps1 X
1 #Report on External Sharing Configuration and Users
2 $Sites = Get-SPOSite | Where {$_.SharingCapability -ne "Disabled"}
3 Foreach ($Site in $Sites)
4 {
5     Write-Host "Checking" $Site.URL -ForegroundColor Green
6     Write-Host "-Sharing Capability:" $Site.SharingCapability -ForegroundColor Yellow
7     Get-SPOExternalUser -SiteUrl $Site.URL | Select DisplayName, Email, InvitedBy, WhenCreated | FT -AutoSize
8 }
```



```
Checking http://brendg-public.sharepoint.com/
-Sharing Capability: ExternalUserAndGuestSharing
Checking https://brendg.sharepoint.com/
-Sharing Capability: ExternalUserAndGuestSharing
Checking https://brendg.sharepoint.com/sites/Finance
-Sharing Capability: ExternalUserAndGuestSharing
```

DisplayName	Email	InvitedBy	WhenCreated
Lee Palmer	lee.palmer12@yahoo.co.uk	garthf@brendg.onmicrosoft.com	03/06/2014 13:53:48


```
Checking https://brendg.sharepoint.com/sites/HR
-Sharing Capability: ExternalUserSharingOnly
```

DisplayName	Email	InvitedBy	WhenCreated
Brendan Griffin	brendankgriffin@hotmail.com	alexdb@brendg.onmicrosoft.com	03/06/2014 13:51:48

How Does this Actually Work?

- ▶ **Get-SPOSite** “Under the Covers”
- ▶ Connect to the **client.svc** WCF service and sends a request as XML

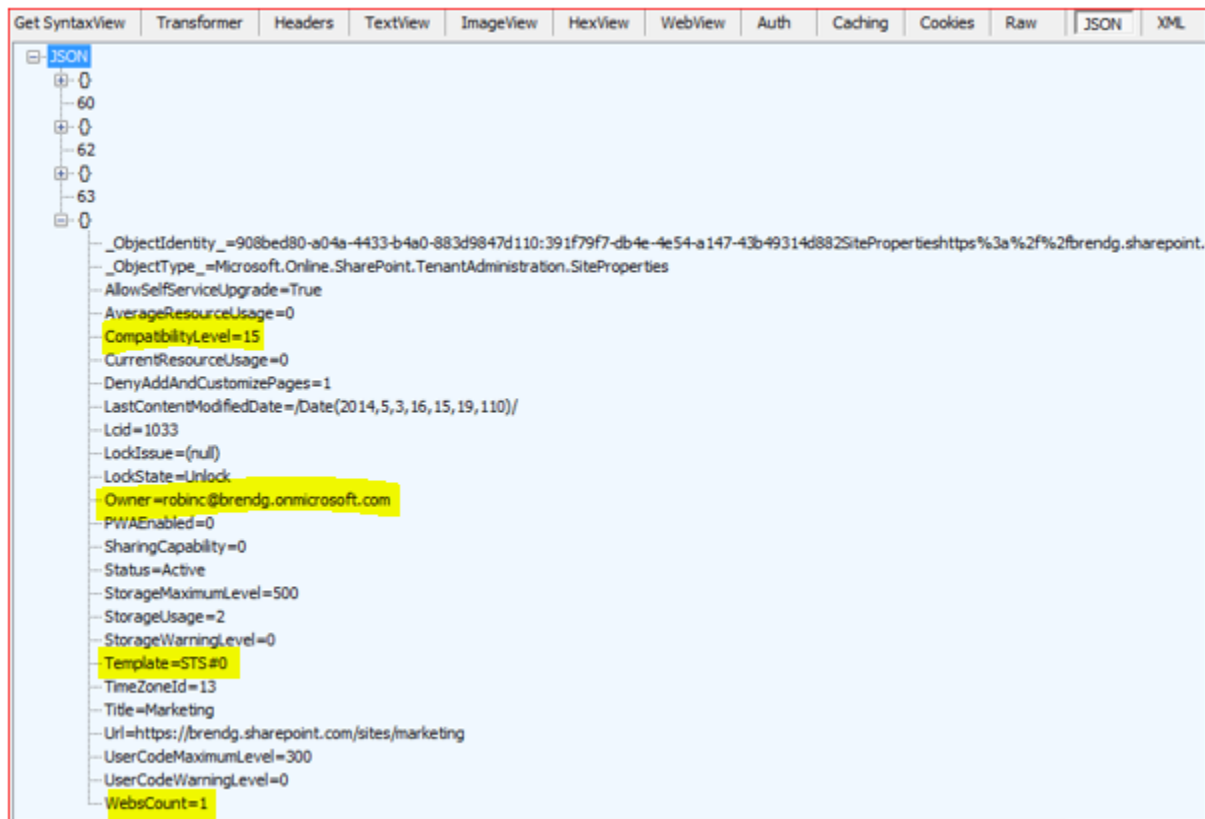
484 200 HTTPS brendg-admin.sharepoint.com /_vti_bin/client.svc/ProcessQuery 847 private application/json; charset=utf-8

The screenshot shows the XML request in the developer tools. The request is a WCF service call to `ProcessQuery` on the `client.svc` endpoint. The XML structure is as follows:

```
<Request [ AddExpandFieldTypeSuffix=true SchemaVersion=15.0.0.0 LibraryVersion=15.0.0.0 ApplicationName=.NET Library xmlns=http://schemas.microsoft.com...>  
  <Actions>  
    <ObjectPath Id="60" ObjectPathId="59" xmlns="http://schemas.microsoft.com/sharepoint/clientquery/2009" />  
    <ObjectPath Id="62" ObjectPathId="61" xmlns="http://schemas.microsoft.com/sharepoint/clientquery/2009" />  
    <Query [ Id=63 ObjectPathId=61 ]>  
      <Query [ SelectAllProperties=true ]>  
        <Properties xmlns="http://schemas.microsoft.com/sharepoint/clientquery/2009" />  
      </Query>  
    </Query>  
  </Actions>  
  <ObjectPaths>  
    <Constructor Id="59" TypeId="{268004ae-ef6b-4e9b-8425-127220d84719}" xmlns="http://schemas.microsoft.com/sharepoint/clientquery/2009" />  
    <Method [ Id=61 ParentId=59 Name=GetSitePropertiesByUrl ]>  
      <Parameters>  
        <Parameter [ Type=String ]>  
          https://brendg.sharepoint.com/sites/marketing  
        <Parameter [ Type=Boolean ]>  
          true  
      </Parameters>  
    </Method>  
  </ObjectPaths>  
</Request>
```


Under the Covers: Get-SPOSite

- Response is returned as **JSON**:



```
Get SyntaxView Transformer Headers TextView ImageView HexView WebView Auth Caching Cookies Raw JSON XML
[{"_ObjectIdentity_": "908bed80-a04a-4433-b4a0-883d9847d110:391f79f7-db4e-4e54-a147-43b49314d882", "SiteProperties": {"_ObjectType_": "Microsoft.Online.SharePoint.TenantAdministration.SiteProperties", "AllowSelfServiceUpgrade": true, "AverageResourceUsage": 0, "CompatibilityLevel": 15, "CurrentResourceUsage": 0, "DenyAddAndCustomizePages": 1, "LastContentModifiedDate": "Date(2014, 5, 3, 16, 15, 19, 110)", "Lcid": 1033, "LockIssue": null, "LockState": "Unlock", "Owner": "robinc@brendg.onmicrosoft.com", "PWAEEnabled": 0, "SharingCapability": 0, "Status": "Active", "StorageMaximumLevel": 500, "StorageUsage": 2, "StorageWarningLevel": 0, "Template": "STS#0", "TimeZoneId": 13, "Title": "Marketing", "Url": "https://brendg.sharepoint.com/sites/marketing", "UserCodeMaximumLevel": 300, "UserCodeWarningLevel": 0, "WebsCount": 1}}
```

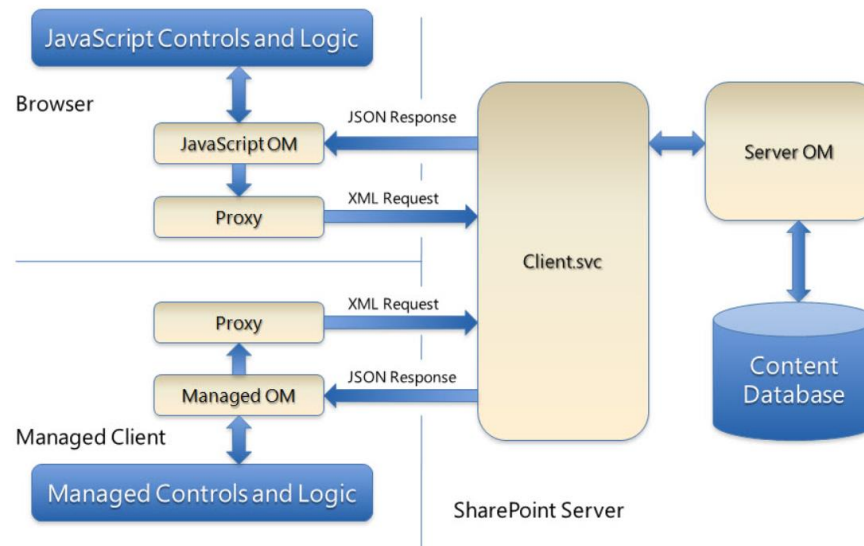
But, What about X?



- ▶ **SPOLE Management Shell** is great for basic Site Collection and User administration
- ▶ But...
 - I can do x On-Premises, it doesn't appear to be possible to do this using the SPOLE Management Shell?
 - I need to do something with Managed Metadata, User Profiles and Search, why isn't there a Cmdlet?

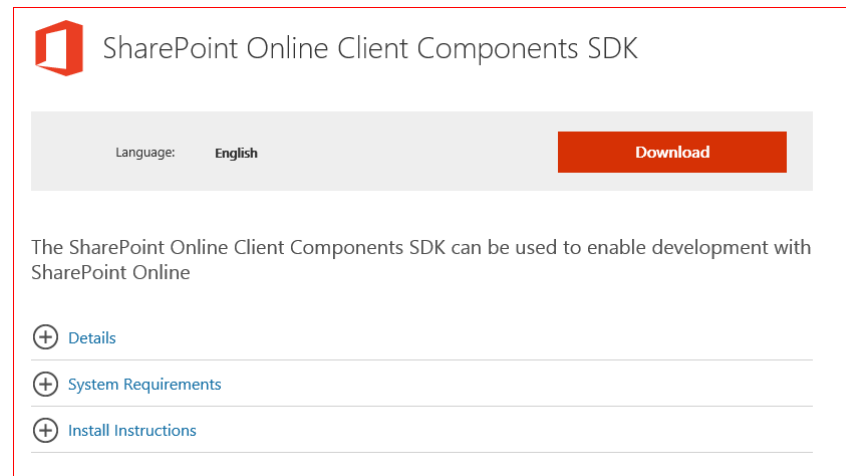
CSOM to the Rescue!

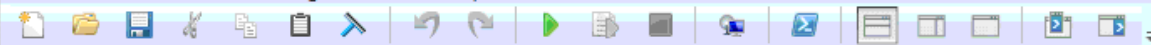
- ▶ The **C**lient **S**ide **O**bject **M**odel was introduced in SharePoint 2010 and is available in SharePoint Online
- ▶ Provides a subset of the Server Object Model
- ▶ Can be executed remotely using JavaScript or .NET



Getting Started with CSOM

- ▶ Download the **SharePoint Online Client Components SDK** [here](#)
- ▶ Includes the DLLs required for CSOM
- ▶ Fire up PowerShell and away we go...





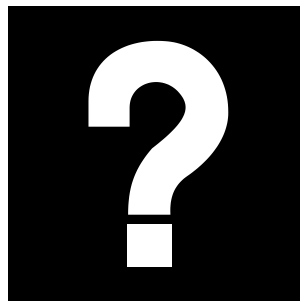
Getting Started - CSOM.ps1 X

```
1 #Specify User
2 $User = "admin@brendg.onmicrosoft.com"
3 #Specify Site to connect to
4 $SiteURL = "https://brendg.sharepoint.com/sites/Finance"
5
6 #Add references to SharePoint client assemblies and authenticate to Office 365 site
7 $16Hive = "C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\16"
8 Add-Type -Path "$16Hive\ISAPI\Microsoft.SharePoint.Client.dll"
9 Add-Type -Path "$16Hive\ISAPI\Microsoft.SharePoint.Client.Runtime.dll"
10
11 #Authenticate to Site
12 $Password = Read-Host -Prompt "Please enter your password" -AsSecureString
13 $Context = New-Object Microsoft.SharePoint.Client.ClientContext($SiteURL)
14 $Creds = New-Object Microsoft.SharePoint.Client.SharePointOnlineCredentials($User,$Password)
15 $Context.Credentials = $Creds
```

PS C:\>

CSOM – What Went Wrong?

- ▶ CSOM is different to the Server Side Object Model
- ▶ To minimize data transfer a slightly different approach is used
- ▶ Requests are bundled together and then sent to the server for execution
- ▶ Only when an object has been returned, can it be used



File Edit View Tools Debug Add-ons Help



CSOM in Action.ps1* X

```
1 #Attempt to retrieve the Site owner
2
3 #Wrong Way!
4 $Context.Site.Owner.Email
5
6 #Correct way
7 $Owner = $Context.Site.Owner
8 #Load the new variable into the context
9 $Context.Load($Owner)
10 #Send the request to the server for execution
11 $Context.ExecuteQuery()
```

PS C:\WINDOWS\system32>

Completed

Ln 1 Col 25

125%

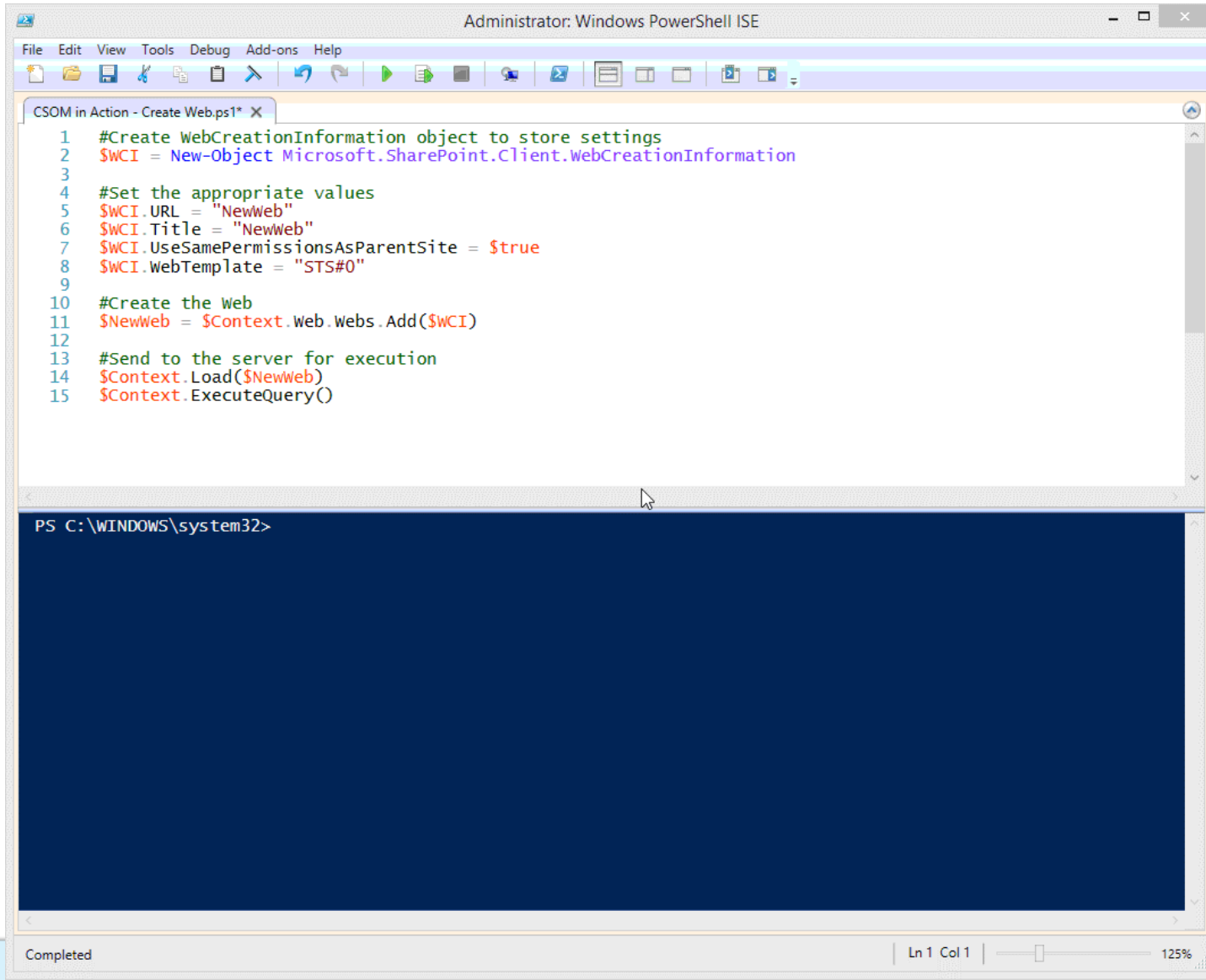
CSOM: 4 Steps to Success

1. Create Variable
2. Load into the Context
3. Execute the Query
4. Use the Variable

```
#Step 1 - Create Variable
$Lists = $Context.Site.RootWeb.Lists
#Step 2 - Load into the Context
$Context.Load($Lists)
#Step 3 - Send the request to the server for execution
$Context.ExecuteQuery()
#Step 4 - Use the Variable
$Lists | Select Title,ItemCount | FT -AutoSize
```

Title	ItemCount
-----	-----
appdata	0
Composed Looks	18
Content type publishing error log	0
Converted Forms	0
Documents	0
Form Templates	0
List Template Gallery	0
Master Page Gallery	209
MicroFeed	2
Project Policy Item List	0
Site Assets	1
Site Pages	2
Solution Gallery	0
Style Library	5
TaxonomyHiddenList	0
Theme Gallery	41
User Information List	17
Web Part Gallery	71
wfpub	0

Using CSOM to Create a Web

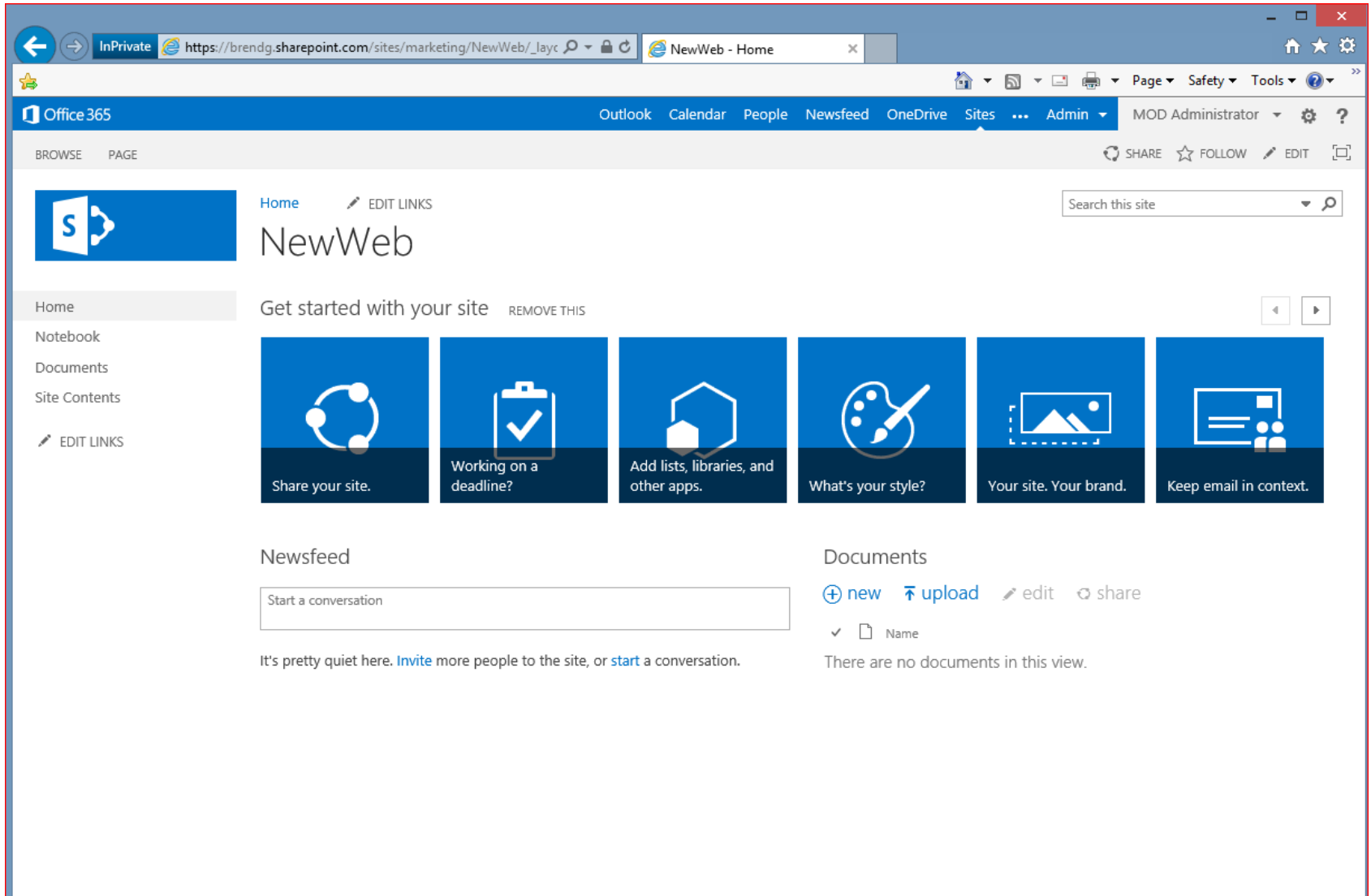


The screenshot shows the Windows PowerShell ISE interface. The title bar reads "Administrator: Windows PowerShell ISE". The menu bar includes File, Edit, View, Tools, Debug, Add-ons, and Help. The toolbar contains various icons for file operations, editing, and execution. The script editor displays a PowerShell script for creating a web using CSOM. The script is as follows:

```
1 #Create WebCreationInformation object to store settings
2 $WCI = New-Object Microsoft.SharePoint.Client.WebCreationInformation
3
4 #Set the appropriate values
5 $WCI.URL = "NewWeb"
6 $WCI.Title = "NewWeb"
7 $WCI.UseSamePermissionsAsParentSite = $true
8 $WCI.WebTemplate = "STS#0"
9
10 #Create the web
11 $NewWeb = $Context.Web.Webs.Add($WCI)
12
13 #Send to the server for execution
14 $Context.Load($NewWeb)
15 $Context.ExecuteQuery()
```

The console window at the bottom shows the prompt "PS C:\WINDOWS\system32>" on a dark blue background. The status bar at the bottom indicates "Completed", "Ln 1 Col 1", and "125%".

Using CSOM to Create a Web



What about Service Applications?

- ▶ CSOM can also be used with Managed Metadata, Search and User Profiles
- ▶ The relevant assemblies need to be referenced
 - Microsoft.SharePoint.Client.Taxonomy.dll (Managed Metadata)
 - Microsoft.SharePoint.Client.Search.dll (Search)
 - Microsoft.SharePoint.Client.UserProfiles.dll (User Profiles)

```
#Add references to SharePoint client assemblies
$16Hive = "C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\16"
Add-Type -Path "$16Hive\ISAPI\Microsoft.SharePoint.Client.dll"
Add-Type -Path "$16Hive\ISAPI\Microsoft.SharePoint.Client.Runtime.dll"
#Service Application specific assemblies
Add-Type -Path "$16Hive\ISAPI\Microsoft.SharePoint.Client.Taxonomy.dll"
Add-Type -Path "$16Hive\ISAPI\Microsoft.SharePoint.Client.Search.dll"
Add-Type -Path "$16Hive\ISAPI\Microsoft.SharePoint.Client.UserProfiles.dll"
```

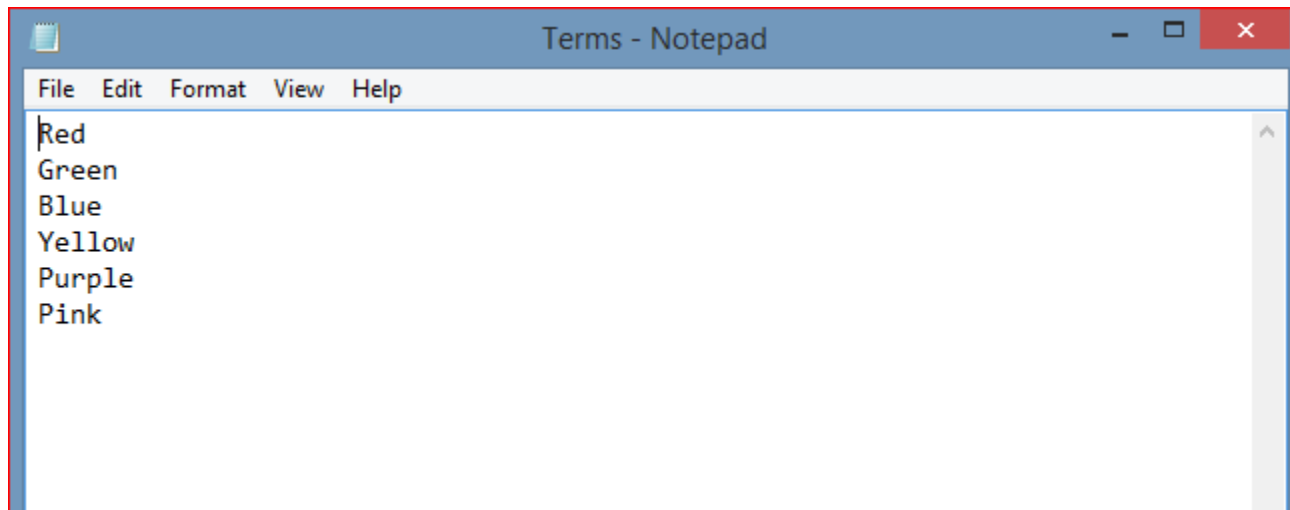
Exporting all User Profiles?

```
1 #Configure Site URL and User
2 $SiteURL = "https://brendg-my.sharepoint.com"
3 #Bind to Site Collection
4 $Context = New-Object Microsoft.SharePoint.Client.ClientContext($SiteURL)
5 $Context.Credentials = $Creds
6
7 #Identify users in the Site Collection
8 $Users = $Context.Web.SiteUsers
9 $Context.Load($Users)
10 $Context.ExecuteQuery()
11
12 #Create People Manager object to retrieve profile data
13 $Output = "D:\Output.csv"
14 $Headings = "Name","Email","OneDrive URL","Phone","Job Title","Department"
15 $Headings -join "," | Out-File -Encoding default -FilePath $Output
16
17 $PeopleManager = New-Object Microsoft.SharePoint.Client.UserProfiles.PeopleManager($Context)
18 Foreach ($User in $Users)
19 {
20     $UserProfile = $PeopleManager.GetPropertiesFor($User.LoginName)
21     $Context.Load($UserProfile)
22     $Context.ExecuteQuery()
23     If ($UserProfile.Email -ne $null)
24     {
25         $SUPP = $UserProfile.UserProfileProperties
26         $Properties = $UserProfile.DisplayName,$UserProfile.Email,$UserProfile.PersonalUrl, $SUPP.WorkPhone,$SUPP.'SPS-JobTitle',$SUPP.Department
27         $Properties -join "," | Out-File -Encoding default -Append -FilePath $Output
28     }
29 }
```

Exporting all User Profiles?

Name	Email	OneDrive URL	Phone	Job Title	Department
Alex Darrow	AlexD@brendg.onmicrosoft.com	https://brendg-my.sharepoint.com/personal/alex_d_brendg_onmicrosoft_com/	+1 858 555 0110	Marketing Assistant	Sales & Marketing
Allie Bellew	AllieB@brendg.onmicrosoft.com	https://brendg-my.sharepoint.com/personal/allieb_brendg_onmicrosoft_com/	+1 425 555 0105	Administrative Assistant	Finance
Anne Wallace	AnneW@brendg.onmicrosoft.com	https://brendg-my.sharepoint.com/personal/anne_w_brendg_onmicrosoft_com/	+1 502 555 0144	President	Executive Management
Aziz Hassouneh	AzizH@brendg.onmicrosoft.com	https://brendg-my.sharepoint.com/personal/azizh_brendg_onmicrosoft_com/	+20 255501070	Accountant II	Finance
Belinda Newman	BelindaN@brendg.onmicrosoft.com	https://brendg-my.sharepoint.com/personal/belinda_n_brendg_onmicrosoft_com/	+1 980 555 0101	Paralegal	Legal
Bonnie Kearney	BonnieK@brendg.onmicrosoft.com	https://brendg-my.sharepoint.com/personal/bonnie_k_brendg_onmicrosoft_com/	+1 858 555 0111	Sr. VP Sales & Marketing	Sales & Marketing
David Longmuir	DavidL@brendg.onmicrosoft.com	https://brendg-my.sharepoint.com/personal/david_l_brendg_onmicrosoft_com/	+1 262 555 0106	Corporate Security Officer	Operations
Denis Dehenne	DenisD@brendg.onmicrosoft.com	https://brendg-my.sharepoint.com/personal/denis_d_brendg_onmicrosoft_com/	+1 205 555 0106	Paralegal	Legal
Dorena Paschke	DorenaP@brendg.onmicrosoft.com	https://brendg-my.sharepoint.com/personal/dorena_p_brendg_onmicrosoft_com/	+1 502 555 0102	CVP Engineering	Engineering
Fabrice Canel	FabriceC@brendg.onmicrosoft.com	https://brendg-my.sharepoint.com/personal/fabrice_c_brendg_onmicrosoft_com/	+1 205 555 0103	Attorney	Legal
Garret Vargas	GarretV@brendg.onmicrosoft.com	https://brendg-my.sharepoint.com/personal/garret_v_brendg_onmicrosoft_com/	+1 206 555 0105	CVP Operations	Operations
Garth Fort	GarthF@brendg.onmicrosoft.com	https://brendg-my.sharepoint.com/personal/garth_f_brendg_onmicrosoft_com/	+1 918 555 0101	Web Marketing Manager	Sales & Marketing
Janet Schorr	JanetS@brendg.onmicrosoft.com	https://brendg-my.sharepoint.com/personal/janet_s_brendg_onmicrosoft_com/	+1 425 555 0109	Product Marketing Manager	Sales & Marketing
Julian Isla	JulianI@brendg.onmicrosoft.com	https://brendg-my.sharepoint.com/personal/julian_i_brendg_onmicrosoft_com/	+1 309 555 0101	Director	Legal
Junmin Hao	JunminH@brendg.onmicrosoft.com	https://brendg-my.sharepoint.com/personal/junmin_h_brendg_onmicrosoft_com/	+1 913 555 0101	CVP Research & Development	Research & Development
Kari Furse	KariF@brendg.onmicrosoft.com	https://brendg-my.sharepoint.com/personal/kari_f_brendg_onmicrosoft_com/	+81 345550115	Budget Analyst	Finance
Katie Jordan	KatieJ@brendg.onmicrosoft.com	https://brendg-my.sharepoint.com/personal/katie_j_brendg_onmicrosoft_com/	+1 412 555 0109	Auditor	Finance
MOD Administrator	admin@brendg.onmicrosoft.com	https://brendg-my.sharepoint.com/personal/admin_brendg_onmicrosoft_com/	8006427676		
Molly Dempsey	MollyD@brendg.onmicrosoft.com	https://brendg-my.sharepoint.com/personal/molly_d_brendg_onmicrosoft_com/	+1 918 555 0104	Product Manager	Sales & Marketing
Pavel Banský	PavelB@brendg.onmicrosoft.com	https://brendg-my.sharepoint.com/personal/pavel_b_brendg_onmicrosoft_com/	+1 732 555 0102	VP Sales	Sales & Marketing
Robin Counts	RobinC@brendg.onmicrosoft.com	https://brendg-my.sharepoint.com/personal/robin_c_brendg_onmicrosoft_com/	+1 954 555 0118	Marketing Assistant	Sales & Marketing
Rob Young	RobY@brendg.onmicrosoft.com	https://brendg-my.sharepoint.com/personal/rob_y_brendg_onmicrosoft_com/	+1 309 555 0104	CVP Legal	Legal
Sara Davis	SaraD@brendg.onmicrosoft.com	https://brendg-my.sharepoint.com/personal/sara_d_brendg_onmicrosoft_com/	+1 918 555 0107	Product Manager	Sales & Marketing
Tony Krijnen	TonyK@brendg.onmicrosoft.com	https://brendg-my.sharepoint.com/personal/tony_k_brendg_onmicrosoft_com/	+1 205 555 0108	CVP Finance	Finance
Zrinka Makovac	ZrinkaM@brendg.onmicrosoft.com	https://brendg-my.sharepoint.com/personal/zrinka_m_brendg_onmicrosoft_com/	+1 858 555 0109	VP Marketing	Sales & Marketing

Adding Terms to the Term Store?



Adding Terms to the Term Store?

```
1 $GroupName = "Group"
2 $TermSetName = "Term Set"
3 $Terms = Get-Content "D:\Terms.txt"
4
5 #Bind to MMS
6 $MMS = [Microsoft.SharePoint.Client.Taxonomy.TaxonomySession]::GetTaxonomySession($Context)
7 $Context.Load($MMS)
8 $Context.ExecuteQuery()
9
10 #Retrieve Term Stores
11 $TermStores = $MMS.TermStores
12 $Context.Load($TermStores)
13 $Context.ExecuteQuery()
14
15 #Bind to Term Store
16 $TermStore = $TermStores[0]
17 $Context.Load($TermStore)
18 $Context.ExecuteQuery()
19
20 #Bind to Group
21 $Group = $TermStore.Groups.GetByName($GroupName)
22 $Context.Load($Group)
23 $Context.ExecuteQuery()
24
25 #Bind to Term Set
26 $TermSet = $Group.TermSets.GetByName($TermSetName)
27 $Context.Load($TermSet)
28 $Context.ExecuteQuery()
29
30 Foreach ($Term in $Terms)
31 {
32     $TermAdd = $TermSet.CreateTerm($Term,1033,[System.Guid]::NewGuid().ToString())
33     $Context.Load($TermAdd)
34     $Context.ExecuteQuery()
35 }
```

Adding Terms to the Term Store?

SEARCH

TAXONOMY TERM STORE

English

Taxonomy_kajmtl61u6RkiKezqfOqhQ==

Contoso

Group

Term Set

Blue

Green

Pink

Purple

Red

Yellow

HR Group

Locations

People

Products

Search Dictionaries

System

GENERAL

Taxonomy_kajmtl61u6RkiKezqfOqhQ==

Available Service Applications

A site may consume multiple metadata applications. Select the one to see in the tree view.

Taxonomy_kajmtl61u6RkiKezqfOqhQ==

Sample Import

The SharePoint metadata manager can import a term set from a UTF-8 CSV format file. Use the sample file as a template for creating import files. Then import the file into the desired group to create a new term set.

[View a sample import file](#)

Term Store Administrators

You can enter user names, group names, or e-mail addresses. Separate them with semicolons. These users will be permitted to create new term set groups and assign users to the group manager role.

admin@brendg.onmicrosoft.com;

Default Language







Select the default language for all Metadata in the system. All terms must have a label defined in their default language.

English

SharePoint® Evolution Roadshow 2014

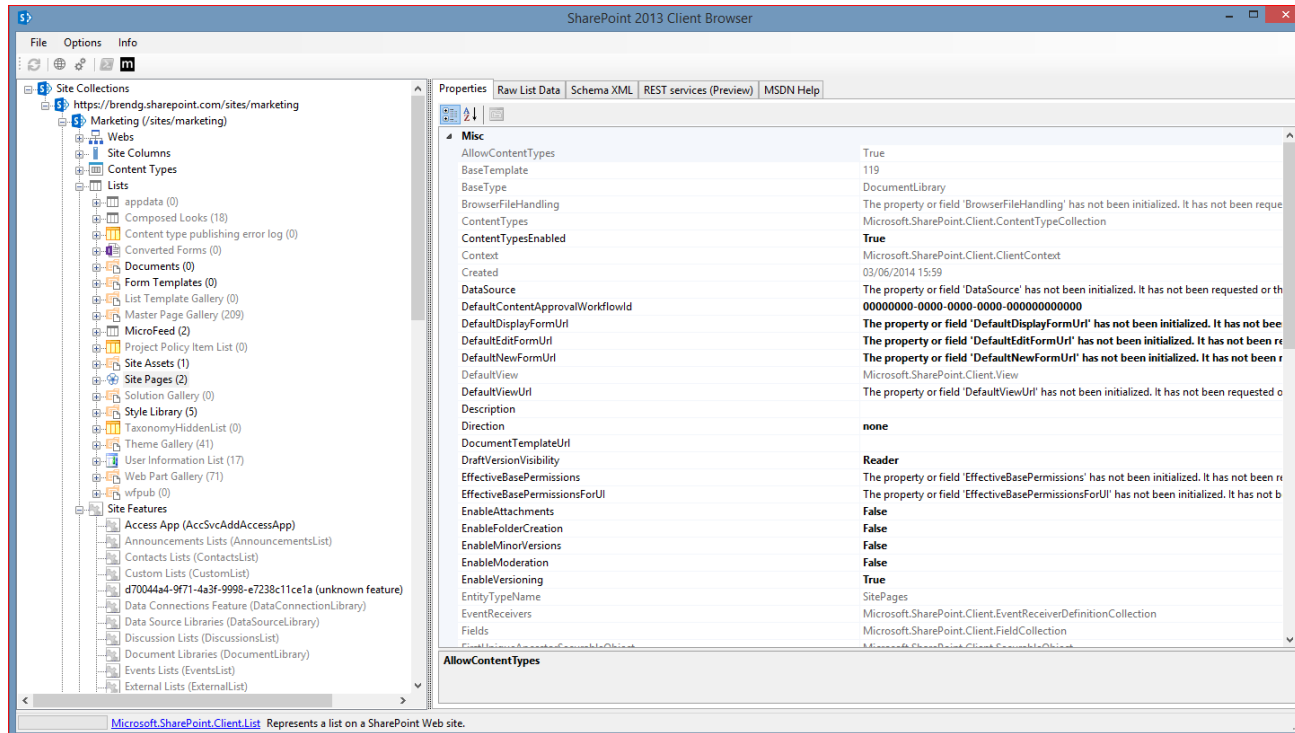
Next Steps...

- ▶ Need to do something else?
- ▶ Refer to the **Microsoft.SharePoint.Client** namespace [documentation](#) on MSDN
- ▶ The possibilities are endless (kind of!)

Microsoft.SharePoint.Client namespace		
SharePoint 2013 Other Versions 1 out of 4 rated this helpful - Rate this topic		
Provides a subset of types and members in the Microsoft.SharePoint namespace for working with a top-level site and its lists or child Web sites.		
◀ Classes		
Class	Description	
 AccessRequests	Operates on access requests and external user invitations.	
 AlternateUrl	Represents an incoming URL and the zone with which it is associated.	
 AlternateUrlPropertyNames	This class and its members are reserved for internal use and are not intended to be used in your code.	
 ApiBlockedException		
 ApiMetadata		
 App	Represents an app loaded on SharePoint and ready to be installed.	

Next Steps...

- SharePoint 2013 Client Browser is a great tool to determine the data that can be obtained using CSOM



Better Together

- ▶ CSOM and the SPOL Management Shell can be used together
- ▶ Why?
 - **Management Shell** is scoped at the **Tenant level**
 - Able to connect to a **Site Collection** but cannot go deeper into the hierarchy
 - **CSOM** is scoped at the Site Collection level
 - Able to work with **Webs, Lists, Items...**
 - Used together the possibilities are endless!

Better Together

► Example:

- Use the **SPOL Management Shell** to identify the URLs for each Site Collection within a Tenant
- Pass the URLs to **CSOM**
- **CSOM** can connect to the Site Collection
- Return useful information
 - Webs
 - Users
 - Lists
 - Features

Better Together

ListAllWebs.ps1* X

```
1  #Use the SPOL Management Shell to obtain a list of Site Collections
2  $SPOLSites = Get-SPOSite
3  Foreach ($Site in $SPOLSites)
4  {
5      #Use CSOM to connect to the Site Collection
6      $SiteURL = $Site.URL
7      $Context = New-Object Microsoft.SharePoint.Client.ClientContext($SiteURL)
8      $Creds = New-Object Microsoft.SharePoint.Client.SharePointOnlineCredentials($User,$Password)
9      $Context.Credentials = $Creds
10
11     #Create a variable listing each Web
12     $Webs = $Context.Web.Webs
13     $Context.Load($Webs)
14     $Context.ExecuteQuery()
15
16     #Output the URL of each Web
17     Write-Host $Context.URL -ForegroundColor Yellow
18     Foreach ($Web in $Webs)
19     {
20         Write-Host "-" $Web.URL -ForegroundColor Green
21     }
22 }
```

Better Together

```
https://brendg.sharepoint.com/sites/contosobeta
https://brendg.sharepoint.com/sites/contosopartners
- https://brendg.sharepoint.com/sites/contosopartners/Facilities
- https://brendg.sharepoint.com/sites/contosopartners/News
- https://brendg.sharepoint.com/sites/contosopartners/Resources
- https://brendg.sharepoint.com/sites/contosopartners/Search
https://brendg.sharepoint.com/sites/demo
https://brendg.sharepoint.com/sites/ediscovery
- https://brendg.sharepoint.com/sites/ediscovery/Case-CT77A11
- https://brendg.sharepoint.com/sites/ediscovery/Case-FD124R59
- https://brendg.sharepoint.com/sites/ediscovery/Case-FD951A88
- https://brendg.sharepoint.com/sites/ediscovery/Case-TM1A934
- https://brendg.sharepoint.com/sites/ediscovery/Case-TM3X122
- https://brendg.sharepoint.com/sites/ediscovery/Case-TM444
- https://brendg.sharepoint.com/sites/ediscovery/northwind
https://brendg.sharepoint.com/sites/Finance
https://brendg.sharepoint.com/sites/HR
https://brendg.sharepoint.com/sites/KnowledgeCenter
https://brendg.sharepoint.com/sites/marketing
- https://brendg.sharepoint.com/sites/marketing/Comms
- https://brendg.sharepoint.com/sites/marketing/CRM
- https://brendg.sharepoint.com/sites/marketing/NewWeb
- https://brendg.sharepoint.com/sites/marketing/Vendors
https://brendg.sharepoint.com/sites/SMBverticals
- https://brendg.sharepoint.com/sites/SMBverticals/construction
- https://brendg.sharepoint.com/sites/SMBverticals/consulting
- https://brendg.sharepoint.com/sites/SMBverticals/healthcare
- https://brendg.sharepoint.com/sites/SMBverticals/lawyer
- https://brendg.sharepoint.com/sites/SMBverticals/realestate
```

CSOM to the Rescue – My Sites

- ▶ In preparation for v15 upgrades, I needed a list of all My Sites running in v14 mode within a Tenant
- ▶ **Get-SPOSite** doesn't return My Sites
- ▶ With a little **CSOM** magic this is possible
- ▶ How?
 - Connect to the User Profile Service
 - Return the My Site URL for each user
 - Connect to each My Site
 - Retrieve the CompatibilityLevel property

CSOM to the Rescue – My Sites

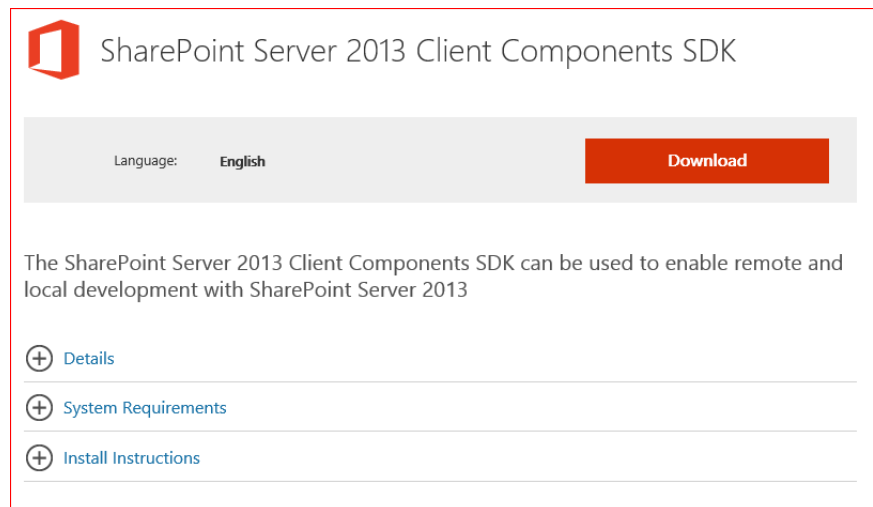
```
1 #Specify tenant admin user, URL, the URL of the MySite host and the location to output the results to
2 $User = "admin@brendg.onmicrosoft.com"
3 $TenantURL = "https://brendg-admin.sharepoint.com"
4 $MySiteURL = "https://brendg-my.sharepoint.com/"
5 $Logfile = "D:\SPOCompatibilityLog.csv"
6
7 #Create log file
8 "Site" + "," + "Compatibility " | Out-File -Encoding default $Logfile
9
10 #Authenticate to Office 365 site - required for CSOM
11 $Password = Read-Host -Prompt "Please enter your password" -AsSecureString
12 $Context = New-Object Microsoft.SharePoint.Client.ClientContext($MySiteURL)
13 $Creds = New-Object Microsoft.SharePoint.Client.SharePointOnlineCredentials($User,$Password)
14 $Context.Credentials = $Creds
15
16 #Identify users in the MySite host Site Collection
17 $Users = $Context.Web.SiteUsers
18 $Context.Load($Users)
19 $Context.ExecuteQuery()
20
21 #Create People Manager object to retrieve profile data
22 $PeopleManager = New-Object Microsoft.SharePoint.Client.UserProfiles.PeopleManager($Context)
23 #Loop through each user, retrieve their MySite URL and then check the compatibilitylevel property
24 Foreach ($User in $Users)
25 {
26     $UserProfile = $PeopleManager.GetPropertiesFor($User.LoginName)
27     $Context.Load($UserProfile)
28     $Context.ExecuteQuery()
29     If ($UserProfile.Email -ne $null -and $UserProfile.UserProfileProperties.PersonalSpace -ne $null)
30     {
31         $UserProfileMySite = $MySiteURL.TrimEnd("/") + $UserProfile.UserProfileProperties.PersonalSpace.TrimEnd("/")
32         Write-Host "Checking" $UserProfileMySite -ForegroundColor Green;
33         $Context2 = New-Object Microsoft.SharePoint.Client.ClientContext($UserProfileMySite)
34         $Context2.Credentials = $Creds
35         $Context2.ExecuteQuery()
36         $MySite = $Context2.Site
37         $Context2.Load($MySite)
38         $Context2.ExecuteQuery()
39         Write-Host "      CompatibilityLevel:"$MySite.CompatibilityLevel ""
40         $OutputString = $MySite.URL + "," + $MySite.CompatibilityLevel
41         Add-content $Logfile $OutputString
42     }
43 }
```


CSOM to the Rescue – My Sites

Site	Compatibility
https://brendg-my.sharepoint.com/personal/alex_d_brendg_onmicrosoft_com	14
https://brendg-my.sharepoint.com/personal/allieb_brendg_onmicrosoft_com	14
https://brendg-my.sharepoint.com/personal/annew_brendg_onmicrosoft_com	15
https://brendg-my.sharepoint.com/personal/azizh_brendg_onmicrosoft_com	15
https://brendg-my.sharepoint.com/personal/belindan_brendg_onmicrosoft_com	15
https://brendg-my.sharepoint.com/personal/bonniek_brendg_onmicrosoft_com	15
https://brendg-my.sharepoint.com/personal/davidl_brendg_onmicrosoft_com	14
https://brendg-my.sharepoint.com/personal/denisd_brendg_onmicrosoft_com	15
https://brendg-my.sharepoint.com/personal/dorenap_brendg_onmicrosoft_com	15
https://brendg-my.sharepoint.com/personal/fabricec_brendg_onmicrosoft_com	15
https://brendg-my.sharepoint.com/personal/garretv_brendg_onmicrosoft_com	15
https://brendg-my.sharepoint.com/personal/garthf_brendg_onmicrosoft_com	15
https://brendg-my.sharepoint.com/personal/janets_brendg_onmicrosoft_com	14
https://brendg-my.sharepoint.com/personal/juliani_brendg_onmicrosoft_com	15
https://brendg-my.sharepoint.com/personal/junminh_brendg_onmicrosoft_com	15
https://brendg-my.sharepoint.com/personal/karif_brendg_onmicrosoft_com	15
https://brendg-my.sharepoint.com/personal/katiej_brendg_onmicrosoft_com	14
https://brendg-my.sharepoint.com/personal/admin_brendg_onmicrosoft_com	15
https://brendg-my.sharepoint.com/personal/mollyd_brendg_onmicrosoft_com	15
https://brendg-my.sharepoint.com/personal/pavelb_brendg_onmicrosoft_com	14
https://brendg-my.sharepoint.com/personal/robinc_brendg_onmicrosoft_com	15
https://brendg-my.sharepoint.com/personal/roby_brendg_onmicrosoft_com	15
https://brendg-my.sharepoint.com/personal/sarad_brendg_onmicrosoft_com	15
https://brendg-my.sharepoint.com/personal/tonyk_brendg_onmicrosoft_com	15
https://brendg-my.sharepoint.com/personal/zrinkam_brendg_onmicrosoft_com	14

CSOM On-Premises?

- ▶ CSOM can also be used On-Premises
- ▶ Download the [SharePoint Server 2013 Client Components SDK](#)
- ▶ Scripts could be run against SharePoint Online and On-Premises*



Summary

- ▶ PowerShell can be used to administrate SharePoint Online
- ▶ The Management Shell provides basic capabilities
- ▶ CSOM can be used to supplement

Thank you for attending!