

HEADLINE: A SIMPLE WAY TO INTERACT WITH SPO SITE COLLECTIONS AND SUBSITES

What is the number of site collections I have? What is the number of subsites in a site collection or site? When was the last time users made changes to this site?

Questions like these are raised quite often by our customers.

A simple answer

We created a solution based on a SharePoint Online list and a PowerShell Script. The Idea is to create SharePoint Online list and populate or synch the information of a Site collection / sites using PowerShell Script.

Here we go....

- 1 First step is to create SharePoint Online list. We created SharePoint Online list "Sites List" as an example. Columns of SharePoint Online List "Sites List" is as under:

Columns		
A column stores information about each item in the list. The following columns are currently available in this list:		
Column (click to edit)	Type	Required
Title	Single line of text	✓
LastItemModifiedDate	Date and Time	
LastUserModifiedDate	Date and Time	
LastContentModifiedDate	Date and Time	
NoOfSubsites	Single line of text	
SiteOwner	Single line of text	
SiteUsagelnMB	Single line of text	
SiteCollection	Yes/No	
URL	Hyperlink or Picture	
Modified	Date and Time	
Created	Date and Time	
Created By	Person or Group	
Modified By	Person or Group	

- 2 Next step is to create the PowerShell script. As we mentioned on some of the other blogs in our series, we need to make sure we check a few key things:

- A make sure that Microsoft.SharePoint.Client.dll and Microsoft.SharePoint.Client.Runtime.dll at following location

```
Add-Type -Path "C:\Program Files\Common Files\microsoft shared\Web Server  
Extensions\15\ISAPI\Microsoft.SharePoint.Client.dll" -ErrorAction Stop  
Add-Type -Path "C:\Program Files\Common Files\microsoft shared\Web Server  
Extensions\15\ISAPI\Microsoft.SharePoint.Client.Runtime.dll" -ErrorAction Stop
```

- B Next to specify values of following parameters

```
$userName = "XXXXXX"  
$password = ConvertTo-SecureString "XXXXXX" -AsPlainText -Force  
$SPOneAdminURL = "XXXXXXXX"  
$Global:webinfo = @()
```

- 3 The following Function return `$context` of SharePoint Online admin site and site collections

```
function getSiteCollectionContext($siteUrl, $siteCollectionAdmin, $securePassword)
{
    $context = New-Object Microsoft.SharePoint.Client.ClientContext($siteUrl)
    $credentials = New-Object
Microsoft.SharePoint.Client.SharePointOnlineCredentials($siteCollectionAdmin,
$securePassword)
    $context.Credentials = $credentials
    return $context
}
```

- 4 The following Function gets all site collections, update `$Global:webinfo` and call recursive function to get desired data of all websites below each site collection

```
function GetSPOSites{
    Begin
    {
        $credentials = New-Object
Microsoft.SharePoint.Client.SharePointOnlineCredentials($username, $password)
    }

    Process
    {
        Connect-SPOService -Url $SPOnlineAdminURL -Credential $credentials
        $sites = Get-SPOSite -Limit All -Detailed
        foreach ($site in $sites)
        {
            # If you want to remove site collection like search
            if(
                $site.Url.Equals("XXXXXX") -or
                $site.Url.Equals("XXXXXX")
            )
            {
                continue;
            }

            try
            {
                $context = getSiteCollectionContext $site.Url $username $password
                $web = $context.Web
                $context.Load($web)
                $context.ExecuteQuery()
                $context.Load($web.Webs)
                $context.ExecuteQuery()

                $Properties = @{
                    Title = $site.Title
                    URL = $site.Url
                    SiteSizeLimitInMB = $site.StorageQuota
                    SiteUsageInMB = $site.StorageUsageCurrent
                    NoOfSubsites = $web.Webs.Count
                    SiteOwner = $site.Owner
                    LastContentModifiedDate = $site.LastContentModifiedDate
                    LastUserModifiedDate = $web.LastItemUserModifiedDate
                    SiteCollection = 'Yes'
                    LastItemModifiedDate = $web.LastItemModifiedDate
                }

                $Global:webinfo += New-Object psobject -Property $properties

                if ($web.Webs.Count -gt 0)
                {
                    getAllWebsitesRecursively $context $web
                }
            }
            catch
            {
                Write-Host $_ -ForegroundColor Red
            }
        }
    }
}
```

- 5 We have created a recursive function to go through each site collection / web, find all websites and add desired data to `$Global:webinfo`

```
function getAllWebSitesRecursively($context, $web){
    foreach($webSite in $web.webs)
    {
        $context.Load($webSite.webs);
        $context.ExecuteQuery();

        $Properties = @{
            Title = $webSite.Title
            URL = $webSite.Url
            NoOfSubsites = $webSite.webs.Count
            LastUserModifiedDate = $webSite.LastItemUserModifiedDate
            LastItemModifiedDate = $webSite.LastItemModifiedDate
            SiteCollection = 'No'
            SiteOwner = $webSite.Author.Title
        }

        $Global:webinfo += New-Object psobject -Property $properties

        if ($webSite.webs.Count -gt 0)
        {
            getAllWebSitesRecursively $context $webSite
        }
    }
}
```

- 6 If we want to add / update to SharePoint Online list, we can use the following script:

```
function SPOSiteListSynch
{
    [CmdletBinding()]
    param(
        [Parameter(Mandatory=$true,
            ValueFromPipeline=$true,
            ValueFromPipelineByPropertyName=$true)]
        [String]
        $siteURL,
        [Parameter(Mandatory=$true,
            ValueFromPipeline=$true,
            ValueFromPipelineByPropertyName=$true)]
        [String]
        $ListName
    )

    Begin
    {
        $dtTable = New-Object System.Data.DataTable
        $col1 = New-Object System.Data.DataColumn Title, ([string])
        $col2 = New-Object System.Data.DataColumn ItemID, ([string])
        $col3 = New-Object System.Data.DataColumn URL, ([string])

        $dtTable.Columns.Add($col1);
        $dtTable.Columns.Add($col2);
        $dtTable.Columns.Add($col3);
    }

    Process
    {
        GetSPOsites

        $context = getSiteCollectionContext $siteURL $userName $password

        $list = $Context.Web.Lists.GetByTitle($ListName)
        $Context.Load($list)
        $Context.ExecuteQuery()

        $qry = New-Object Microsoft.SharePoint.Client.CamlQuery
        $qry.ViewXml = "<View><Query><Where></Where><OrderBy></OrderBy></Query>" +
            "<ViewFields>" +
            "<FieldRef Name='Title' /> " +
            "<FieldRef Name='URL' /> " +
            "</ViewFields>" +
            "</View>"
    }
}
```

```

$items = $list.GetItems($qry)
$Context.Load($items)
$Context.ExecuteQuery()

foreach($item in $items)
{
    $row = $dtTable.NewRow()
    $row.Title = $item["Title"]
    $row.URL = $item["URL"].Url
    $row.ItemID = $item["ID"]
    $dtTable.Rows.Add($row)
}

$TZ = [System.TimeZoneInfo]::FindSystemTimeZoneById("AUS Eastern Standard Time")

try
{
    foreach($property in $Global:webinfo)
    {
        $dtrow = $dtTable.Select("URL='" + $property.URL + "'")

        if($dtrow.Count -eq 1)
        {
            $spoListItem = $list.GetItemById($dtrow.ItemID)
            $spoListItem["Title"]=$property.Title
            $spoListItem["LastItemModifiedDate"] =
[System.TimeZoneInfo]::ConvertTimeFromUtc($property.LastItemModifiedDate, $TZ)
            $spoListItem["LastUserModifiedDate"] =
[System.TimeZoneInfo]::ConvertTimeFromUtc($property.LastUserModifiedDate, $TZ)
            if($property.LastContentModifiedDate){$spoListItem["LastContentModifiedDate"] =
[System.TimeZoneInfo]::ConvertTimeFromUtc($property.LastContentModifiedDate, $TZ)}
            $spoListItem["NoOfSubsites"] = $property.NoOfSubsites
            $spoListItem["SiteOwner"] = $property.SiteOwner
            $spoListItem["SiteUsageInMB"] = $property.SiteUsageInMB
            $spoListItem["SiteCollection"] = if($property.SiteCollection -eq
"Yes"){true}else{$false}
            $spoListItem["URL"]= $property.URL + ", " + $property.Title
            $spoListItem.Update()
            $Context.ExecuteQuery()
        }
        else
        {
            $spoListItemCreationInformation = New-Object
Microsoft.SharePoint.Client.ListItemCreationInformation
            $spoListItem=$list.AddItem($spoListItemCreationInformation)
            $spoListItem["Title"]=$property.Title
            $spoListItem["LastItemModifiedDate"] =
[System.TimeZoneInfo]::ConvertTimeFromUtc($property.LastItemModifiedDate, $TZ)
            $spoListItem["LastUserModifiedDate"] =
[System.TimeZoneInfo]::ConvertTimeFromUtc($property.LastUserModifiedDate, $TZ)
            if($property.LastContentModifiedDate){$spoListItem["LastContentModifiedDate"] =
[System.TimeZoneInfo]::ConvertTimeFromUtc($property.LastContentModifiedDate, $TZ)}
            $spoListItem["NoOfSubsites"] = $property.NoOfSubsites
            $spoListItem["SiteOwner"] = $property.SiteOwner
            $spoListItem["SiteUsageInMB"] = $property.SiteUsageInMB
            $spoListItem["SiteCollection"] = if($property.SiteCollection -eq
"Yes"){true}else{$false}
            $spoListItem["URL"]= $property.URL + ", " + $property.Title
            $spoListItem.Update()
            $Context.ExecuteQuery()
        }
    }
}
catch [Exception]
{
    Write-Host $_.Exception
}
}

SPOSiteListSynch -siteURL XXXXX -ListName 'Sites List'

```

....and here's the full scripts...

```
Add-Type -Path "C:\Program Files\Common Files\Microsoft Shared\Web Server
Extensions\15\ISAPI\Microsoft.SharePoint.Client.dll" -ErrorAction Stop
Add-Type -Path "C:\Program Files\Common Files\Microsoft Shared\Web Server
Extensions\15\ISAPI\Microsoft.SharePoint.Client.Runtime.dll" -ErrorAction Stop

$username = "XXXXXX"
$password = ConvertTo-SecureString "XXXXXX" -AsPlainText -Force
$SPOnlineAdminURL = "XXXXXX"
$Global:webinfo = @()

function GetSiteCollectionContext($siteUrl, $siteCollectionAdmin, $securePassword)
{
    $context = New-Object Microsoft.SharePoint.Client.ClientContext($siteUrl)
    $credentials = New-Object
Microsoft.SharePoint.Client.SharePointOnlineCredentials($siteCollectionAdmin, $securePassword)
    $context.Credentials = $credentials
    return $context
}

function GetAllWebsitesRecursively($context, $web){
    foreach($website in $web.Webs)
    {
        $context.Load($website.Webs);
        $context.ExecuteQuery();

        $properties = @{
            Title = $website.Title
            URL = $website.Url
            NoOfSubsites = $website.Webs.Count
            LastUserModifiedDate = $website.LastItemUserModifiedDate
            LastItemModifiedDate = $website.LastItemModifiedDate
            SiteCollection = 'No'
            SiteOwner = $website.Author.Title
        }

        $Global:webinfo += New-Object psobject -Property $properties

        if ($website.Webs.Count -gt 0)
        {
            GetAllWebsitesRecursively $context $website
        }
    }
}

function GetSPOsites{
    Begin
    {
        $credentials = New-Object
Microsoft.SharePoint.Client.SharePointOnlineCredentials($username, $password)
    }

    Process
    {
        Connect-SPOService -Url $SPOnlineAdminURL -Credential $credentials
        $sites = Get-SPOSite -Limit All -Detailed
        foreach ($site in $sites)
        {
            # If you want to remove site collection like search
            if(
                $site.Url.Equals("XXXXXX") -or
                $site.Url.Equals("XXXXXX")
            )
            {
                continue;
            }

            try
            {
                $context = GetSiteCollectionContext $site.Url $username $password
                $web = $context.Web
                $context.Load($web)
                $context.ExecuteQuery()
            }
        }
    }
}
```

```

        $context.Load($web.webs)
        $context.ExecuteQuery()

$Properties = @{
    Title = $site.Title
    URL = $site.Url
    SiteSizeLimitInMB = $site.StorageQuota
    SiteUsageInMB = $site.StorageUsageCurrent
    NoOfSubsites = $web.Webs.Count
    SiteOwner = $site.Owner
    LastContentModifiedDate = $site.LastContentModifiedDate
    LastUserModifiedDate = $web.LastItemUserModifiedDate
    SiteCollection = 'Yes'
    LastItemModifiedDate = $web.LastItemModifiedDate
}

$Global:webinfo += New-Object psobject -Property $properties

    if ($web.webs.Count -gt 0)
    {
        getAllWebsitesRecursively $context $web
    }
}
catch
{
    Write-Host $_ -ForegroundColor Red
}
}
}

function SPOSiteListSynch
{
    [CmdletBinding()]
    param(
        [Parameter(Mandatory=$true,
            ValueFromPipeline=$true,
            ValueFromPipelineByPropertyName=$true)]
        [String]
        $siteURL,
        [Parameter(Mandatory=$true,
            ValueFromPipeline=$true,
            ValueFromPipelineByPropertyName=$true)]
        [String]
        $ListName
    )

    Begin
    {
        $dtTable = New-Object System.Data.DataTable
        $col1 = New-Object System.Data.DataColumn Title, ([string])
        $col2 = New-Object System.Data.DataColumn ItemID, ([string])
        $col3 = New-Object System.Data.DataColumn URL, ([string])

        $dtTable.Columns.Add($col1);
        $dtTable.Columns.Add($col2);
        $dtTable.Columns.Add($col3);
    }

    Process
    {
        GetSPOsites

        $context = getSiteCollectionContext $siteURL $userName $password

        $list = $context.web.Lists.GetByTitle($ListName)
        $context.Load($list)
        $context.ExecuteQuery()

        $qry = New-Object Microsoft.SharePoint.Client.CamlQuery
        $qry.ViewXml = "<View><Query><Where></Where><OrderBy></OrderBy></Query>" +
            "<ViewFields>" +
            "<FieldRef Name='Title' /> " +
            "<FieldRef Name='URL' /> " +
            "</ViewFields>" +
            "</View>"
    }
}

```

```

$items = $list.GetItems($qry)
$Context.Load($items)
$Context.ExecuteQuery()

foreach($item in $items)
{
    $row = $dtTable.NewRow()
    $row.Title = $item["Title"]
    $row.URL = $item["URL"].Url
    $row.ItemID = $item["ID"]
    $dtTable.Rows.Add($row)
}

$TZ = [System.TimeZoneInfo]::FindSystemTimeZoneById("AUS Eastern Standard Time")

try
{
    foreach($property in $Global:webinfo)
    {
        $dtrow = $dtTable.Select("URL='" + $property.URL + "'")

        if($dtrow.Count -eq 1)
        {
            $spoListItem = $list.GetItemById($dtrow.ItemID)
            $spoListItem["Title"]=$property.Title
            $spoListItem["LastItemModifiedDate"] =
[System.TimeZoneInfo]::ConvertTimeFromUtc($property.LastItemModifiedDate, $TZ)
            $spoListItem["LastUserModifiedDate"] =
[System.TimeZoneInfo]::ConvertTimeFromUtc($property.LastUserModifiedDate, $TZ)
            if($property.LastContentModifiedDate){$spoListItem["LastContentModifiedDate"]
= [System.TimeZoneInfo]::ConvertTimeFromUtc($property.LastContentModifiedDate, $TZ)}
            $spoListItem["NoOfSubsites"] = $property.NoOfSubsites
            $spoListItem["SiteOwner"] = $property.SiteOwner
            $spoListItem["SiteUsageInMB"] = $property.SiteUsageInMB
            $spoListItem["SiteCollection"] = if($property.SiteCollection -eq
"Yes"){true}else{false}
            $spoListItem["URL"]= $property.URL + ", " + $property.Title
            $spoListItem.Update()
            $Context.ExecuteQuery()
        }
        else
        {
            $spoListItemCreationInformation = New-Object
Microsoft.SharePoint.Client.ListItemCreationInformation
            $spoListItem=$list.AddItem($spoListItemCreationInformation)
            $spoListItem["Title"]=$property.Title
            $spoListItem["LastItemModifiedDate"] =
[System.TimeZoneInfo]::ConvertTimeFromUtc($property.LastItemModifiedDate, $TZ)
            $spoListItem["LastUserModifiedDate"] =
[System.TimeZoneInfo]::ConvertTimeFromUtc($property.LastUserModifiedDate, $TZ)
            if($property.LastContentModifiedDate){$spoListItem["LastContentModifiedDate"]
= [System.TimeZoneInfo]::ConvertTimeFromUtc($property.LastContentModifiedDate, $TZ)}
            $spoListItem["NoOfSubsites"] = $property.NoOfSubsites
            $spoListItem["SiteOwner"] = $property.SiteOwner
            $spoListItem["SiteUsageInMB"] = $property.SiteUsageInMB
            $spoListItem["SiteCollection"] = if($property.SiteCollection -eq
"Yes"){true}else{false}
            $spoListItem["URL"]= $property.URL + ", " + $property.Title
            $spoListItem.Update()
            $Context.ExecuteQuery()
        }
    }
}
catch [Exception]
{
    Write-Host $_.Exception
}
}

SPOSiteListSynch -siteURL XXXXX -ListName 'Sites List'

```