HEADLINE: EXPORTING LARGE SHAREPOINT LISTS - IT'S EASIER THAN YOU THINK

Exporting SharePoint list to Excel or CVS is one of the requirements we get from our customers time and time again.

Typically, we address this by opening Excel and importing data from SharePoint Online. We will go through details later but first; let's see why this approach has one HUGE problem and by HUGE we mean **HUGE**.

The problem

Let's have a look at the following scenario...

We have used the "Excel Import" method and only started to get data from the SharePoint Online list and after about 105 minutes:



7815 rows loaded out of 32932. Do the math and you will have an idea how much time it will take. <u>MS</u> <u>documentations</u> states that we can have 30 million items in a list, so in our scenario with 32932 rows we are working with only 0.1 percent of the limit.

The solution

To export a list with large volume of data, we return to the trusted PowerShell and CSOM.

Don't get us wrong, when list data is not too big, we prefer importing data to Excel but when data volume is very large, we will be using PowerShell and CSOM.

So let's see the first method ... Excel Import

Let's start with importing SharePoint Online list data to Excel. Start Excel and go to select "Data" tab. Click on "Get Data" -> "From Online Services" -> "From SharePoint Online List".



Pop up will appear with the heading "SharePoint lists". It's a little misleading so don't get fooled by the heading and enter URL of your SharePoint Online *site*, not the URL of SharePoint Online *list*.

SharePoint lists	
URL ()	_
http: Enter the root URL for the SharePoint site not including subfolders.	
	OK Cancel

The following Pop up will confirm that you are on the right track

	×
Connecting	
Please wait while we establish a connection to	
	Cancel

Once connected you will see all your available lists





Commercial in Confidence



Select desired list and data will appear in preview window



	×	
Processing Querie	ès	
Determining automatic tr	ansformations	
Waiting for		
-		
	Skip	

Eventually, you will be able to see the data from the SharePoint Online list in Excel. Don't be alarmed by all the data that appeared in Excel. Go to "Query" tab under "Query Tools" and click on "Edit".



"Query Editor" window will appear. Scroll horizontally and find column "FieldValuesAsText".

Close & Load •	Refresh Preview • Manage	es d Editor Choose Rei Columns - Colu	move µmns • Rows • Rows • Rows •	↓ ↓ Split Column ▼ By ↓ 2 Replace	Record V Mer rst Row as Headers V App ce Values	ge Queries end Queries Manage D Parameters	ata source settings
Close	Query	Manage Colu	imns Reduce Rows Sc	ort Transform	n C	ombine Parameters Da	ata Sources New Q
> [mentFiles	ContentType	GetDlpPolicyTip	FieldValuesAsHtml	FieldValuesAsText	FieldValuesForEdit	File fir
	1	Record	Record	Record	Record	Record	Record
s	2	Record	Record	Record	Record	Record	Record
lerie	3	Record	Record	Record	Record	Record	Record
ð	4	Record	Record	Record	Record	Record	Record
	5	Record	Record	Record	Record	Record	Record
	6	Record	Record	Record	Record	Record	Record
	7	Record	Record	Record	Record	Record	Record
	8	Record	Record	Record	Record	Record	Record
	9	Record	Record	Record	Record	Record	Record
	10	Record	Record	Record	Record	Record	Record
	11	Record	Record	Record	Record	Record	Record
	12	Record	Record	Record	Record	Record	Record
	13	Record	Record	Record	Record	Record	Record
	14	Record	Record	Record	Record	Record	Record
	15	Record	Record	Record	Record	Record	Record
	16	Record	Record	Record	Record	Record	Record
	17	Record	Record	Record	Record	Record	Record
	18	Record	Record	Record	Record	Record	Record
	19	Record	Record	Record	Record	Record	Record
	20	Record	Record	Record	Record	Record	Record
	21	Record	Record	Record	Record	Record	Record
	22	Record	Record	Record	Record	Record	Record
	23 🔨						>
63 COLUI	MNS, 999+ ROWS						



Make sure "FieldValuesAsText" is selected and click on "Remove Columns" -> "Remove Other Columns". This will remove all columns from "Query Editor".

Choo Colum Mai	Dese Remove Remove Reverse Remove Rows * Row	x 2↓ nove ws → fit fit	Data Type: Record Use First Row as Head Use First Row as Head By Transform	ders • 😨 Append Queries •	Manage Data source settings Parameters Data Sources
	Remove Other Colu	mns			
41th		untentType ♦	GetDlpPolicyTip	FieldValuesAsHtml	FieldValuesAsText
	Table Remove al	ll columns from this table,	Record	Record	Record F
	Table except the	currently selected ones.	Record	Record	Record F
	Table	Record	Record	Record	Record F
	Table	Record	Record	Record	Record F
	Table	Record	Record	Record	Record F
	Table	Record	Record	Record	Record F
	Table	Record	Record	Record	Record F
	Table	Record	Record	Record	Record F
	Table	Record	Record	Record	Record F
	Table	Record	Record	Record	Record F
	Table	Record	Record	Record	Record F
	Table	Record	Record	Record	Record F
	Table	Record	Record	Record	Record F
	Table	Record	Record	Record	Record F
	Table	Record	Record	Record	Record F
	Table	Record	Record	Record	Record F
	Table	Record	Record	Record	Record
	Table	Record	Record	Record	Record F
	Table	Record	Record	Record	Record F
	Table	Record	Record	Record	Record F
	Table	Record	Record	Record	Record F
	Table	Record	Record	Record	Record F
					>

	FieldValuesAsText	
1	Record	₹↓
2	Record	
3	Record	✓ (Select All Columns)
4	Record	Content lypeld
5	Record	✓ OData_ModerationComments
6	Record	
7	Record	Modified_x0020_By
8	Record	IV Created_XUU2U_By
9	Record	Image: Market Marke
10	Record	HIML_X0020_File_X0020_Type
11	Record	OData_SourceUn
12	Record	
13	Record	
14	Record	✓ xd ProalD
15	Record	✓ xd Signature
16	Record	✓ OData ShortcutUrl
17	Record	✓ OData_ShortcutSiteId
18	Record	✓ OData_ShortcutWebId
19	Record	OData ChartautUniquald
20	Record	✓ Use original column name as prefix
21	Record	A List mucho incomplete
22	Record	List may be incomplete. Load more
23	Record	OK Cancel

Once all columns are removed, click on the right corner on "FieldValuesAsText" and the following will appear:

Select the columns that you need. Excel will take time to load and you will have all columns from the SharePoint List along with the relevant data. Rename the columns as needed and click "Save & Load". This will take time as Excel will update itself based on the changes made and we have imported data from SharePoint Online list to Excel.

TIP: One little tip about the "Query Settings" window. This view displays all the changes performed in sequential order so can undo any change at any point!





This approach works fine when we are working with small data size but takes a lot time with each step added in Query Settings. One way to try and speed this up is to not add any steps in Query Settings and try to with work with the data populated in Excel from the SharePoint Online list in first place. Again, this works OK for smaller data volumes.

The Other method ... PowerShell

Other way that works much better for larger data volumes is to write a PowerShell script and export data from SharePoint Online list to CSV or Excel. We will be using CSV format for simplicity.

1 First step is to make sure that Microsoft.SharePoint.Client.dll and Microsoft.SharePoint.Client.Runtime.dll are at the 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
```

2 Now, let's record the values of the following parameters to get the context of SharePoint Online site:

```
$username = "XXXXXX"
$userPassword = "XXXXXXXX"
$siteURL = "XXXXXXX"
$listtitle = "XXXXXX"
```

3 We will be using the following CAML query which will have OOTB columns. We can add as many columns as desired

```
$qCommand = @"
<View Scope="RecursiveAll">
        <Query>
            <OrderBy Override='True'><FieldRef Name='Modified' /></OrderBy>
        </Query>
        <ViewFields>
            <FieldRef Name='Modified' /><FieldRef Name='Created' />
            </viewFields>
            <Fields="TRUE">5000</RowLimit>
        </viewFields>
            </viewF
```

4 Accessing SharePoint Online site and list in script below

\$secpasswd = ConvertTo-SecureString \$userPassword -AsPlainText -Force \$context = New-Object Microsoft.SharePoint.Client.ClientContext(\$siteURL) \$context.Credentials = New-Object Microsoft.SharePoint.Client.SharePointOnlineCredentials(\$UserName, \$secpasswd) \$list = \$context web Lists CotPyTitle(\$listtitle)

\$list = \$context.Web.Lists.GetByTitle(\$listtitle)
\$context.Load(\$list)
\$context.ExecuteQuery()

- 5 A couple of important parts of the script:
 - A First is **\$position** variable which will store the current position
 - B and **\$itemsinfo** is array of PSObject

\$position = \$null
\$itemsinfo = @()



6 We will be using Do Until loop and will be terminating when \$position variable is null and adding data in \$itemsinfo

```
D0
{
    $camlQuery = New-Object Microsoft.SharePoint.Client.CamlQuery
    $camlQuery_ListItemCollectionPosition = $position
    $currentCollection = $list GetItems($qCommand)
$context.Load($currentCollection)
    $context ExecuteQuery()
    $position = $currentCollection.ListItemCollectionPosition
    foreach($listitem in $currentCollection)
    {
         try
         $fieldvalue = @{
             Created =
[System.TimeZoneInfo]::ConvertTimeFromUtc($listitem["Created"], $TZ)
             Modified
[System.TimeZoneInfo]::ConvertTimeFromUtc($listitem["Modified"], $TZ)
         }catch
             Write-Host $_
         }
         $itemsinfo += New-Object psobject -Property $fieldvalue
    }
Until($position -eq $null)
```

7 Last step is to create CSV file

\$itemsinfo | Select-Object Created, Modified | export-csv "C:\DataFromSharePointOnlineList.csv" -NoTypeInformation



...and the full script...

```
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
$strCurrentTimeZone = (Get-WmiObject win32_timezone).StandardName
$TZ = [System.TimeZoneInfo]::FindSystemTimeZoneById($strCurrentTimeZone)
$username = "XXXXXX"
$userPassword = "XXXXXXX"
$siteURL = "XXXXXXX"
$listtitle = "XXXXXX"
$qCommand = @"
<View Scope="RecursiveAll">
     <Query>
         <OrderBy Override='True'><FieldRef Name='Modified' /></OrderBy>
     </Query>
     <ViewFields>
          <FieldRef Name='Modified' /><FieldRef Name='Created' />
     </ViewFields>
     <RowLimit Paged="TRUE">5000</RowLimit>
</View>
$secpasswd = ConvertTo-SecureString $userPassword -AsPlainText -Force
$context = New-Object Microsoft.SharePoint.Client.ClientContext($siteURL)
$context.Credentials = New-Object Microsoft.SharePoint.Client.SharePointOnlineCre-
dentials($UserName, $secpasswd)
$list = $context.Web.Lists.GetByTitle($listtitle)
$context.Load($list)
$context.ExecuteQuery()
$position = $null
$itemsinfo = @()
DO
{
     $cam]Query = New-Object Microsoft.SharePoint.Client.Cam]Query
     $camlQuery.ListItemCollectionPosition = $position
     $currentCollection = $list.GetItems($qCommand)
     $context.Load($currentCollection)
     $context.ExecuteQuery()
     $position = $currentCollection_ListItemCollectionPosition
     foreach($listitem in $currentCollection)
          try
          $fieldvalue = @{
               Created = [System.TimeZoneInfo]::ConvertTimeFromUtc($listitem["Cre-
ated"],
          $TZ)
               Modified = [System.TimeZoneInfo]::ConvertTimeFromUtc($listitem["Modi-
fied"], $TZ)
               }
          }catch
               Write-Host $_
          }
          $itemsinfo += New-Object psobject -Property $fieldvalue
     }
3
Until($position -eq $null)
$itemsinfo | Select-Object Created, Modified | export-csv "C:\DataFromSharePoin-
tOnlineList.csv" -NoTypeInformation
```

One last workaround if want to create Excel, following code will generate Excel from CSV



```
$excel = New-Object -ComObject excel.application
$excel.visible=$false
$reportOut = $excel.Workbooks.Add()
$wb = $excel.WorkBooks.Open("C:\DataFromSharePointOnlineList.csv")
$wb.Worksheets.Item(1).Name = "DataFromSharePointOnlineList"
$wb.Worksheets.Copy($reportOut.WorkSheets.Item(1))
$wb.Close(0)
$reportOut.worksheets.item("Sheet1").Delete()
$strdate = get-date
$filename = "C:\DataFromSharePointOnlineList.xlsx"
$reportOut.SaveAs($filename,[Microsoft.Office.Interop.Excel.XlFileFormat]::xlOpenXM
LWorkbook)
$reportOut.Close(0)
$excel.Quit()
```