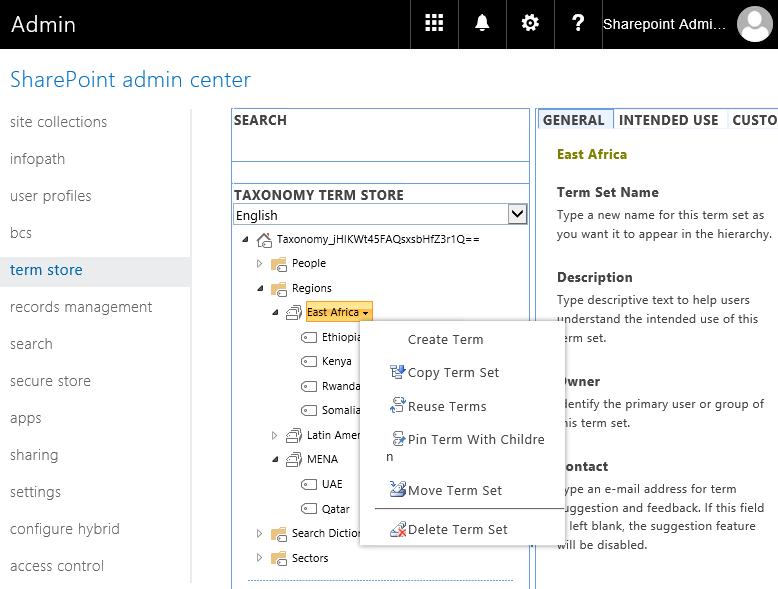
**SharePoint Online: Export Term Set to CSV using PowerShell**

[December 13, 2016](http://www.sharepointdiary.com/2016/12/sharepoint-online-powershell-to-export-term-set-to-csv.html)  [Client Side Object Model (CSOM)](http://www.sharepointdiary.com/search/label/Client%20Side%20Object%20Model%20%28CSOM%29?&max-results=10), [CSV](http://www.sharepointdiary.com/search/label/CSV?&max-results=10), [Managed Metadata](http://www.sharepointdiary.com/search/label/Managed%20Metadata?&max-results=10), [PowerShell](http://www.sharepointdiary.com/search/label/PowerShell?&max-results=10), [SharePoint Online](http://www.sharepointdiary.com/search/label/SharePoint%20Online?&max-results=10)

**Requirement:**Had to extract all terms in a term set to CSV format in SharePoint Online.  
  
Unfortunately, there is no way to export sharepoint online export term set from UI. But PowerShell can help!

**[](https://4.bp.blogspot.com/-iiEk8oiY340/WjAjuQg7LCI/AAAAAAAANho/-RF6AOGpOAAE3pCZs3eu6bN1RtAPDhPtgCLcBGAs/s1600/sharepoint%2Bonline%2Bpowershell%2Bexport%2Bterm%2Bset.png)**

**SharePoint Online PowerShell to Export Term Set**  
PowerShell to export all terms from a given term set in Term store of SharePoint Online.

#Read more: <http://www.sharepointdiary.com/2016/12/sharepoint-online-powershell-to-export-term-set-to-csv.html#ixzz5hUSGI4Lk>

#Load SharePoint CSOM Assemblies

Add-Type -Path "C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\16\ISAPI\Microsoft.SharePoint.Client.dll"

Add-Type -Path "C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\16\ISAPI\Microsoft.SharePoint.Client.Runtime.dll"

Add-Type -Path "C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\16\ISAPI\Microsoft.SharePoint.Client.Taxonomy.dll"

#Variables for Processing

$AdminURL = "https://crescent-admin.sharepoint.com/"

$TermGroupName = "Regions"

$TermSetName = "East Africa"

$CSVFile="C:\Temp\TermSetData.csv"

Try {

    #Get Credentials to connect

    $Cred = Get-Credential

    $Credentials = New-Object Microsoft.SharePoint.Client.SharePointOnlineCredentials($Cred.Username, $Cred.Password)

    #Setup the context

    $Ctx = New-Object Microsoft.SharePoint.Client.ClientContext($AdminURL)

    $Ctx.Credentials = $Credentials

    #Get the term store

    $TaxonomySession=[Microsoft.SharePoint.Client.Taxonomy.TaxonomySession]::GetTaxonomySession($Ctx)

    $TermStore =$TaxonomySession.GetDefaultSiteCollectionTermStore()

    $Ctx.Load($TaxonomySession)

    $Ctx.Load($TermStore)

    $Ctx.ExecuteQuery()

    #Get the Term Group

    $TermGroup=$TermStore.Groups.GetByName($TermGroupName)

    #Get the term set

    $TermSet = $TermGroup.TermSets.GetByName($TermSetName)

    #Get all tersm from the term set

    $Terms = $TermSet.Terms

    $Ctx.Load($Terms)

    $Ctx.ExecuteQuery()

    Write-Output $TermsetName > $CSVFile

    #Export Terms to CSV

    Foreach($Term in $Terms)

    {

        Write-Output $Term.Name >> $CSVFile

    }

    Write-host "Term Set Data Exported Successfully!" -ForegroundColor Green

}

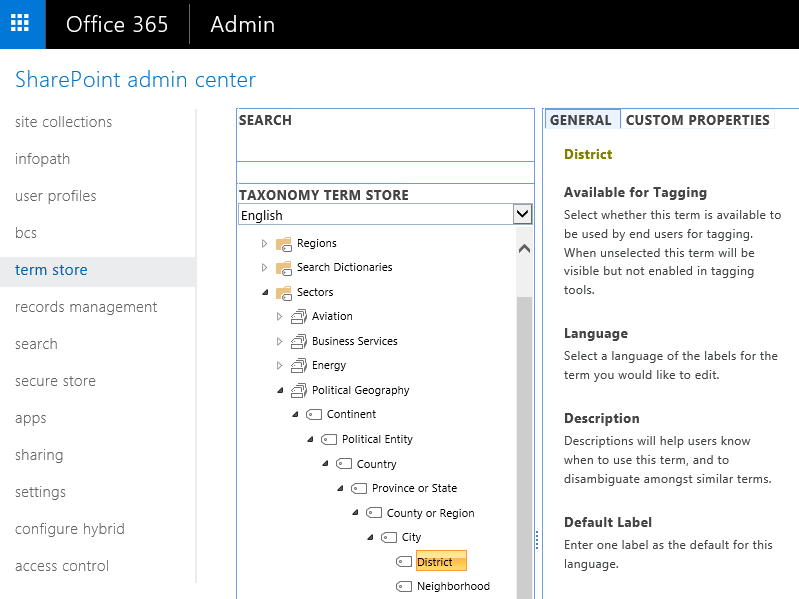
Catch {

    write-host -f Red "Error Exporting Term Set!" $\_.Exception.Message

}

#Read more: <http://www.sharepointdiary.com/2016/12/sharepoint-online-powershell-to-export-term-set-to-csv.html#ixzz5hUSBPpxy>

This script extract each term in the given term set into a CSV file. But wait! each term can go up to 7 levels deep! While the above script exports terms at one-single level, Here is my another PowerShell script to export terms and child terms of all levels of a given term set.

[](https://2.bp.blogspot.com/-WDWmBK19ZQU/WjAl3dbgWGI/AAAAAAAANhs/HFLS0a8XL2Ue-_hKDkmJqRsFT1BMtjyOwCLcBGAs/s1600/SharePoint%2BOnline%2BExport%2BTerm%2BSet.png)

**Export Term Set in SharePoint Online using PowerShell:**  
  
#Read more: <http://www.sharepointdiary.com/2016/12/sharepoint-online-powershell-to-export-term-set-to-csv.html#ixzz5hUSMwz5o>

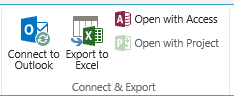
|  |
| --- |
| #Load SharePoint CSOM Assemblies  Add-Type -Path "C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\16\ISAPI\Microsoft.SharePoint.Client.dll"  Add-Type -Path "C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\16\ISAPI\Microsoft.SharePoint.Client.Runtime.dll"  Add-Type -Path "C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\16\ISAPI\Microsoft.SharePoint.Client.Taxonomy.dll"    #Variables for Processing  $AdminURL = "https://crescent-admin.sharepoint.com/"  $TermGroupName = "Sectors"  $TermSetName = "Political Geography"  $CSVFile="C:\Temp\TermSet.csv"    #Custom Function get child terms of a given term  Function Get-Terms([Microsoft.SharePoint.Client.Taxonomy.Term] $Term,[String]$ParentTerm,[int] $Level)  {    $ChildTerms = $Term.Terms    $Ctx.Load($ChildTerms)    $Ctx.ExecuteQuery()    if($ParentTerm)    {      $ParentTerm = $ParentTerm + "," + $Term.Name    }    else    {      $ParentTerm = $Term.Name    }      Foreach ($SubTerm in $ChildTerms)    {       $Level = $Level + 1       #Terms may have upto 7 levels       $NumofCommas =  7 - $Level       $commas =""         #Append Commas       For ($j=0; $j -lt $NumofCommas; $j++)       {          $Commas = $Commas + ","       }        #Append the Output to CSV File      "," + "," + "," + $Term.IsAvailableForTagging + ",""$($Term.Description)""," + $ParentTerm + "," + $SubTerm.Name + $Commas >> $CSVFile        #Call the function recursively      Get-Terms -Term $SubTerm -ParentTerm $ParentTerm -Level $Level    }  }  Try {      #Get Credentials to connect      $Cred = Get-Credential      $Credentials = New-Object Microsoft.SharePoint.Client.SharePointOnlineCredentials($Cred.Username, $Cred.Password)        #Setup the context      $Ctx = New-Object Microsoft.SharePoint.Client.ClientContext($AdminURL)      $Ctx.Credentials = $Credentials        #Get the term store      $TaxonomySession=[Microsoft.SharePoint.Client.Taxonomy.TaxonomySession]::GetTaxonomySession($Ctx)      $TermStore =$TaxonomySession.GetDefaultSiteCollectionTermStore()      $Ctx.Load($TaxonomySession)      $Ctx.Load($TermStore)      $Ctx.ExecuteQuery()        #Write Termset CSV Header (As in the standard format)      "Term Set Name,Term Set Description,LCID,Available for Tagging,Term Description,Level 1 Term,Level 2 Term,Level 3 Term,Level 4 Term,Level 5 Term,Level 6 Term,Level 7 Term" > $CSVFile        #Get the Term Group      $TermGroup=$TermStore.Groups.GetByName($TermGroupName)        #Get the term set      $TermSet = $TermGroup.TermSets.GetByName($TermSetName)      $Ctx.Load($Termset)      $Ctx.ExecuteQuery()        #Get all tersm from the term set      $Terms = $TermSet.Terms      $Ctx.Load($Terms)      $Ctx.ExecuteQuery()        #Write 2nd line as Termset properties(As per standard format)      $TermSet.Name + ",""$($TermSet.Description)""," + $TermStore.DefaultLanguage + "," + $TermSet.IsAvailableForTagging + ",""$($Terms[0].Description)""," + $Terms[0].Name + "," + "," + "," + "," + "," + "," >> $CSVFile        #Process each Term in the termset      Foreach($Term in $Terms)      {          write-host $Term.Name          Get-Terms $Term -Level 1 -ParentTerm ""      }      Write-host "Term Set Data Exported Successfully!" -ForegroundColor Green  }  Catch {      write-host -f Red "Error Exporting Term Set!" $\_.Exception.Message  } |

#Read more: <http://www.sharepointdiary.com/2016/12/sharepoint-online-powershell-to-export-term-set-to-csv.html#ixzz5hUSSdEiK>

# Missing Ribbon in SharePoint Online or SharePoint On-Premises List

BY [ZUBAIR ALEXANDER](https://www.zubairalexander.com/blog/author/sp_admin/) · FEBRUARY 19, 2017

You may run into a situation where you don’t see the ribbon on your SharePoint Online or SharePoint on-premises list. The ribbon offers easy access to common tasks in a SharePoint list. It also offers helpful tools, such as Connect to Outlook, Export to Excel, and Open with Access.

[](https://www.zubairalexander.com/blog/wp-content/uploads/2017/02/SP_ribbon_03.png)

Thanks to Danish Islam for the solutions posted on [TechNet wiki](https://social.technet.microsoft.com/wiki/contents/articles/31204.ribbon-missing-on-sharepoint-list-or-library-page.aspx). This article expands on the solutions in the TechNet article.

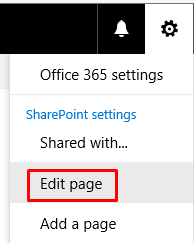
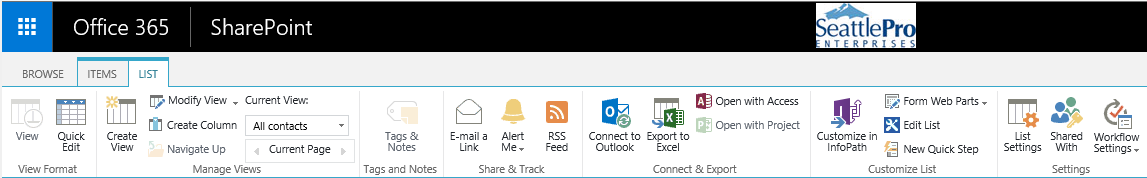
#### **Scenario #1**

One of the most common reasons for a missing ribbon that I have experienced is the addition of **Content Editor Web Part (CEWP)**. This is because SharePoint no longer looks at the page as a List View page, it reclassifies it as an Application page and therefore the ribbon disappears. Here are the details.

#### **Solution #1**

In this article, I will use images from SharePoint Online as an example. SharePoint Online is part of Microsoft Office 365. The solution also applies to SharePoint on-premises, such as SharePoint 2013 and SharePoint 2016.

Because the CEWP often causes the ribbon to disappear, you need to delete it to see if that’s the root of the problem.

1. Edit the list page that contains the CEWP.  
   [](https://www.zubairalexander.com/blog/wp-content/uploads/2017/02/SP_ribbon_02.png)
2. Export the CEWP, in case you need it at a later time or if you are going to use it in another list or library.
3. Delete the CEWP because that’s often the culprit, and then save the page.
4. Your ribbon should now be visible.  
   [](https://www.zubairalexander.com/blog/wp-content/uploads/2017/02/SP_ribbon_01.png)

Keep in mind that some of the the options, tabs, or sections on the ribbon are only available when an item is selected. For example, the Edit Document or Download a Copy options will be grayed out until an item in the list is selected. Some options are only available when you select a single item, while others work when you select multiple list items. Also, often the user is clicking in the section where the CEWP is located. If the user clicks on the page closer to the area where the list items are listed, the ribbon will be displayed. Keep this in mind even after you have removed the CEWP. You may not see the ribbon when you go to the list, but clicking on the page near the rows (not necessarily selecting a row) will display the ribbon. Yes, I know…..it’s weird.

#### **Scenario #2**

On some earlier versions of SharePoint, after migration to a later version you may notice that the ribbon has disappeared.

**Solution #2**

Try this JavaScript provided by [Danish Islam](https://social.technet.microsoft.com/wiki/contents/articles/31204.ribbon-missing-on-sharepoint-list-or-library-page.aspx) on TechNet wiki. You can add the script on any CEWP on the page.

*<scripttype="text/javascript"src="https://code.jquery.com/jquery-1.11.3.min.js"></script>*

*<scripttype="text/javascript">*

*$(document).ready(function(){*

*var elem = document.getElementById("MSOZoneCell\_WebPartWPQ2");*

*if(elem != null) {*

*var dummyevent = new Array();*

*dummyevent["target"] = elem;*

*dummyevent["srcElement"] = elem;*

*WpClick(dummyevent);*

*\_ribbonStartInit("Ribbon.Browse", true)*

*}*

*});*

*</script>*

#### **Problem #3**

You are not using CEWP and you haven’t migrated from an earlier version of SharePoint.

#### **Solution #3**

In SharePoint Online, try switching to the Classic mode. This can bring back the ribbon.

#### **Browser Warning!**

For best results, use 32-bit Internet Explorer. Unfortunately, 32-bit Internet Explorer is the ONLY Web browser supported by Microsoft to work properly with all versions of SharePoint. Google Chrome, Mozilla Firefox, Apple Safari and Microsoft Edge browsers are not fully supported because they don’t support ActiveX controls. All 64-bit versions of Internet Explorer are also not fully supported. For more information check out my article [Web Browser Support for SharePoint Server 2016](https://www.zubairalexander.com/blog/web-browser-support-for-sharepoint-server-2016/).