|  |  |
| --- | --- |
| **TABLE OF CONTENTS** | **PAGE NUMBER** |
| * PowerShell Script to fetch all lookup   Column from SharePoint Web Application | 2 |
| * PowerShell Script to list out all Check-Out   File from SharePoint Web Application | 7 |

**PowerShell Script to Copy PowerShell Script to fetch all lookup Column from SharePoint Web Application**

This script will pull all lookup column with (URL, List name, Field Name) from SharePoint web Application, this script is very useful while migration planning

**Estimated Time Saved: 30 Hrs/month** manual work (This may vary based on the action frequency & files count/size)

**Implementation done: August 2020**

**SCRIPT:**

[System.Reflection.Assembly]::LoadWithPartialName("Microsoft.SharePoint")

try{

#Create a variable based on the current date and time

$StartTime = (Get-Date -UFormat "%Y-%m-%d\_%I-%M-%S %p").tostring()

$0 = $MyInvocation.MyCommand.Definition

$dp0 = [System.IO.Path]::GetDirectoryName($0)

$output = "C:\lookup.csv"

$logFile="C:\lookup.txt"

Write "Script started running at " $StartTime >> $logFile

##Creating and Returning a DataTable##

function createDT()

{

###Creating a new DataTable###

$tempTable = New-Object System.Data.DataTable

##Creating Columns for DataTable##

$col1 = New-Object System.Data.DataColumn("URL")

$col2 = New-Object System.Data.DataColumn("List Name")

$col3 = New-Object System.Data.DataColumn("Field Name")

###Adding Columns for DataTable###

$tempTable.columns.Add($col1)

$tempTable.columns.Add($col2)

$tempTable.columns.Add($col3)

return ,$tempTable

}

[System.Data.DataTable]$dTable = createDT

#Initialize Workflow Count variable

$TTNExcludeLists = "Solution Gallery",

"Master Page Gallery"

$farm = [Microsoft.SharePoint.Administration.SPFarm]::Local

$websvcs = $farm.Services | where -FilterScript {$\_.GetType() -eq [Microsoft.SharePoint.Administration.SPWebService]}

foreach ($websvc in $websvcs)

{

try

{

foreach ($webApplication in $websvc.WebApplications)

{

try

{

Write "`r`n Inside the loop for web application" $webApplication.Url >> $logFile

foreach($site in $webApplication.Sites)

{

try

{

Write "`r`n Inside the loop for site" $site.Url >> $logFile

foreach($web in $site.AllWebs)

{

Write "`r`n Inside the loop for web" $web.Url >> $logFile

foreach($list in $web.Lists)

{

Write "`r`n Inside the loop for the list" $list.Title >> $logFile

if(-Not ($TTNExcludeLists -Contains $list.Title))

{

foreach($fld in $list.Fields)

{

if($fld.TypeDisplayName -eq "Lookup" -and $fld.Hidden -eq $false -and $fld.FromBaseType -eq $false)

{

Write-Host "web URL: " $web.Url "ListName: " $list.Title "FeildName: " $fld.Title

$row = $dTable.NewRow()

$row["URL"] = $web.Url

$row["List Name"] = $list.Title

$row["Field Name"] = $fld.Title

$dTable.rows.Add($row)

}

}

}

}

$web.Dispose()

}

}

catch [Exception]{

Write $\_.Exception|format-list -force >>$logFile

Write-Host -f red $\_.Exception|format-list -force

}

finally{

if($web){

$web.Dispose()

}

}

$site.Dispose()

}

}

catch [Exception]{

Write $\_.Exception|format-list -force >>$logFile

Write-Host -f red $\_.Exception|format-list -force

}

finally{

if($site){

$site.Dispose();

}

}

}

}

catch [Exception]{

Write $\_.Exception|format-list -force >>$logFile

Write-Host -f red $\_.Exception|format-list -force

}

}

if($dTable -ne $null)

{

$dTable | Export-CSV -path $output -notype

#Write-Host "Done" -ForegroundColor Green

}

else

{

Write-Host "Could not write anything to table" -ForegroundColor red

}

}

catch [Exception]{

Write $\_.Exception|format-list -force >>$logFile

Write-Host -f red $\_.Exception|format-list -force

}

$EndTime = (Get-Date -UFormat "%Y-%m-%d\_%I-%M-%S %p").tostring()

Write "Script stopped at" $EndTime >> $logFile

**PowerShell Script to list out all Check-Out File from SharePoint Web Application**

This script will list out all Checkout file with (URL, Check-out By, File size) from SharePoint Site Collection, this script is very useful while migration planning

**Estimated Time Saved: 40 Hrs/month** manual work (This may vary based on the action frequency & files count/size)

**Implementation done: September 2020**

**SCRIPT:**

#check to see if the PowerShell Snapin is added

if((Get-PSSnapin | Where {$\_.Name -eq "Microsoft.SharePoint.PowerShell"}) -eq $null) {

Add-PSSnapin Microsoft.SharePoint.PowerShell;

}

#enter your site URL

$spWeb = Get-SPWeb "<Web Application URL>"

function GetCheckedItems($spWeb)

{

Write-Host "Scanning Site: $($spWeb.Url)"

foreach ($list in ($spWeb.Lists | ? {$\_ -is [Microsoft.SharePoint.SPDocumentLibrary]})) {

Write-Host "Scanning List: $($list.RootFolder.ServerRelativeUrl)"

foreach ($item in $list.CheckedOutFiles) {

if (!$item.Url.EndsWith(".aspx")) { continue }

$writeTable = @{

"URL"=$spWeb.Site.MakeFullUrl("$($spWeb.ServerRelativeUrl.TrimEnd('/'))/$($item.Url)");

"Checked Out By"=$item.CheckedOutBy;

"Author"=$item.File.CheckedOutByUser.Name;

"Checked Out Since"=$item.CheckedOutDate.ToString();

"File Size (KB)"=$item.File.Length/1000;

}

New-Object PSObject -Property $writeTable

}

foreach ($item in $list.Items) {

if ($item.File.CheckOutStatus -ne "None") {

if (($list.CheckedOutFiles | where {$\_.ListItemId -eq $item.ID}) -ne $null) { continue }

$writeTable = @{

"URL"=$spWeb.Site.MakeFullUrl("$($spWeb.ServerRelativeUrl.TrimEnd('/'))/$($item.Url)");

"Checked Out By"=$item.File.CheckedOutByUser.LoginName;

"Author"=$item.File.CheckedOutByUser.Name;

"Checked Out Since"=$item.File.CheckedOutDate.ToString();

"File Size (KB)"=$item.File.Length/1000;

}

New-Object PSObject -Property $writeTable

}

}

}

foreach($subWeb in $spWeb.Webs)

{

GetCheckedItems($subWeb)

}

$spWeb.Dispose()

}

GetCheckedItems($spWeb) | Out-GridView

# alternative output file

GetCheckedItems($spWeb) | Export-Csv C:\CheckedOutItemsflightops.csv