# Title: Remove-DuplicateAuthorizedTypeEntryFromWebConfig.ps1  
# Author: [joeric@microsoft.com](mailto:joeric@microsoft.com)  
# License: This PowerShell script is provided as-is with no warranty expressed or implied.  Use at your own risk.  
# Usage: Run as-is to detect the problem in each web application without making any changes  
# Example: ./Remove-DuplicateAuthorizedTypeEntryFromWebConfig.ps1  
# Change $detectOnly to $false to have it do the cleanup.    
# Keep in mind that each applicaiton pool will recycle due to the web.config change.

[bool]$detectOnly = $true

if($null -eq (Get-PSSnapin Microsoft.SharePoint.PowerShell -ErrorAction SilentlyContinue) -and $null -eq (Get-Command Get-SPFarm -ErrorAction SilentlyContinue))  
{  
    Add-PSSnapin Microsoft.SharePoint.PowerShell -ErrorAction SilentlyContinue | Out-Null  
}

function Remove-DuplicateAuthorizedTypeEntryFromWebConfig  
{  
    [CmdletBinding()]  
    param (  
        [Parameter(Mandatory=$false,ValueFromPipeline=$true)]  
        [Microsoft.SharePoint.Administration.SPWebApplication]  
        $WebApplication,

        [Parameter(Mandatory=$false,ValueFromPipeline=$true)]  
        [bool]  
        $detectOnly = $true  
    )  
    $zones = [enum]::GetValues([Microsoft.SharePoint.Administration.SpUrlZone])  
    foreach($zone in $zones)  
    {  
        $null = $iisSettings  
        try  
        {  
            $iisSettings = $WebApplication.IisSettings[$zone]  
        }  
        catch [System.Management.Automation.RuntimeException]  
        {  
            continue  
        }

        if($null -ne $iisSettings)  
        {  
            try  
            {  
                $webConfigPath = $iisSettings.Path.ToString() + "\web.config"  
                if(!(Test-Path -Path $webConfigPath))  
                {  
                    Write-Warning "no web.config found at $($webConfigPath)"  
                    continue  
                }

                Write-Host "Processing web.config for $($zone) zone of web application `"$($WebApplication.DisplayName)`""

                #backup existing web.config  
                if(!$detectOnly)  
                {  
                    $bakPath = ($iisSettings.Path.ToString() + "\" + [datetime]::Now.ToString("yyyy\_MM\_dd\_hh\_mm\_ss\_") + "web.config.bak")  
                    Copy-Item -Path $webConfigPath -Destination $bakPath

                    if(!(Test-Path -Path $bakPath))  
                    {  
                        Write-Warning "unable to verify backup file at $($bakPath), skipping additional processing"  
                        continue  
                    }  
                }

                [xml]$x = Get-Content $webConfigPath  
                $authorizedTypes = $x.configuration.'System.Workflow.ComponentModel.WorkflowCompiler'.authorizedTypes.targetFx.ChildNodes

                $simplifiedAuthorizedTypes = @()  
                $c = 0  
                [bool]$foundDuplicates = $false

                foreach($authorizedType in $authorizedTypes)  
                {  
                    $existingAuthorizedType = $null  
                    $existingAuthorizedType = $simplifiedAuthorizedTypes | ?{$\_.Assembly -eq $authorizedType.Assembly -and $\_.NameSpace -eq $authorizedType.Namespace -and $\_.TypeName -eq $authorizedType.TypeName -and $\_.Authorized -eq $authorizedType.Authorized}  
                    if($null -eq $existingAuthorizedType)  
                    {  
                        $simplifiedAuthorizedTypes += $authorizedType  
                    }  
                    else  
                    {  
                        $c++  
                        $foundDuplicates = $true  
                    }  
                }

                if($foundDuplicates)  
                {  
                    Write-Host "Found $($c) duplicates in $($webConfigPath)"

                    $simplifiedXml = $null  
                    foreach($simplifiedType in $simplifiedAuthorizedTypes)  
                    {  
                        $simplifiedXml += ("`n        <authorizedType Assembly=`"{0}`" Namespace=`"{1}`" TypeName=`"{2}`" Authorized=`"{3}`" />" -f $simplifiedType.Assembly, $simplifiedType.Namespace, $simplifiedType.TypeName, $simplifiedType.Authorized)  
                    }  
                    $originalLength = $x.InnerXml.Length  
                    $x.configuration.'System.Workflow.ComponentModel.WorkflowCompiler'.authorizedTypes.targetFx.InnerXml = $simplifiedXml  
                    $updatedLength = $x.InnerXml.Length  
                    $difference = $originalLength - $updatedLength

                    if(!$detectOnly)  
                    {  
                        Write-Host "Committing changes to $($webConfigPath)"  
                        $x.Save($webConfigPath)  
                        Write-Host "after removal of duplicates web.config size is $($updatedLength), reduced by $($difference) bytes."  
                    }  
                    else  
                    {  
                        Write-Host "DETECT ONLY: after removal of duplicates web.config size would be $($updatedLength), reduced by $($difference) bytes."  
                    }  
                }  
                else  
                {  
                    Write-Host "No duplicate authrozied type entries found in $($webConfigPath)"  
                }  
            }  
            catch  
            {  
                Write-Warning "Error encountered in $($zone) zone of web application `"$($WebApplication.DisplayName)`""  
                continue  
            }

        }  
        else  
        {  
            continue  
        }

    }  
}

$server = [Microsoft.SharePoint.Administration.SPServer]::Local  
if($server.Role -in [Microsoft.SharePoint.Administration.SPServerRole]"WebFrontEnd", "SingleServer", "SingleServerFarm", "WebFrontEndWithDistributedCache" `  
    -or $null -ne ([Microsoft.SharePoint.Administration.SPWebService]::ContentService.Instances | ?{$\_.Server -eq $server -and $\_.Status -eq "Online"}))  
{  
    $webApps = Get-SPWebApplication  
    foreach($webApp in $webApps)  
    {  
        Remove-DuplicateAuthorizedTypeEntryFromWebConfig -WebApplication $webApp -detectOnly $detectOnly  
    }  
}

if($null -ne ([Microsoft.SharePoint.Administration.SPWebService]::AdministrationService.Instances | ?{$\_.Server -eq $server -and $\_.Status -eq "Online"}))  
{  
    Remove-DuplicateAuthorizedTypeEntryFromWebConfig -WebApplication ([Microsoft.SharePoint.Administration.SPAdministrationWebApplication]::Local) -detectOnly $detectOnly  
}[Top of the Document](#_top)