

```

## Remove_PortGroup_at_Cluster_Level.ps1
## build 1.06
## Eric Krejci
## ekrejci.wordpress.com
## Twitter - @ekrejci
## on error stop the script
$erroractionpreference = "Stop"
$Username = "DOMAIN\USERNAME"
$Password = read-host "Enter Password" -assecurestring

### declare your vcenter

$VCenterServer = "vcenter.fqdn"

### declare the name of the DataCenter

$TargetDataCenter = "DataCenterName"

### general variables

$IsVMconnectedtoPortgroup = $false

$Isoldportpresent = $false

### Functions Start

function call-listbox {
    Param( $DataCenter ## This is the DataCenter
    )

[void]
[System.Reflection.Assembly]::LoadWithPartialName("System.Windows.Forms")
[void] [System.Reflection.Assembly]::LoadWithPartialName("System.Drawing")

$objForm = New-Object System.Windows.Forms.Form
$objForm.Text = "Select a Cluster"
$objForm.Size = New-Object System.Drawing.Size(300,200)
$objForm.StartPosition = "CenterScreen"

$objForm.KeyPreview = $True
$objForm.Add_KeyDown({if ($_.KeyCode -eq "Enter")
    {$x=$objListBox.SelectedItem;$objForm.Close()}})
$objForm.Add_KeyDown({if ($_.KeyCode -eq "Escape")
    {$objForm.Close()}})

$OKButton = New-Object System.Windows.Forms.Button
$OKButton.Location = New-Object System.Drawing.Size(75,120)
$OKButton.Size = New-Object System.Drawing.Size(75,23)
$OKButton.Text = "OK"
$OKButton.Add_Click({$x=$objListBox.SelectedItem;$objForm.Close()})
$objForm.Controls.Add($OKButton)

$CancelButton = New-Object System.Windows.Forms.Button
$CancelButton.Location = New-Object System.Drawing.Size(150,120)
$CancelButton.Size = New-Object System.Drawing.Size(75,23)
$CancelButton.Text = "Cancel"
$CancelButton.Add_Click({$objForm.Close()})
$objForm.Controls.Add($CancelButton)

$objLabel = New-Object System.Windows.Forms.Label

```

```

$objLabel.Location = New-Object System.Drawing.Size(10,20)
$objLabel.Size = New-Object System.Drawing.Size(280,20)
$objLabel.Text = "Please select a Cluster:"
$objForm.Controls.Add($objLabel)

$objListBox = New-Object System.Windows.Forms.ListBox
$objListBox.Location = New-Object System.Drawing.Size(10,40)
$objListBox.Size = New-Object System.Drawing.Size(260,20)
$objListBox.Height = 80
$objListBox.Sorted = $True

$clusters = Get-Cluster -Location (Get-Datacenter -name $DataCenter)

foreach ($cluster in $clusters) {

[void] $objListBox.Items.Add($cluster)
}

$objForm.Controls.Add($objListBox)

$objForm.Topmost = $True

$objForm.Add_Shown({$objForm.Activate()})
[void] $objForm.ShowDialog()

$X
}

### Functions End

### connect to the vCenter
Connect-VIServer -Server $vCenterserver -user $Username -Password
([Runtime.InteropServices.Marshal]::PtrToStringAuto([Runtime.InteropServices.Marshal]::SecureStringToBSTR($password)))

### call the list of the clusters within the specified DataCenter

$Cluster = call-listbox -DataCenter $TargetDataCenter

## testing if $Cluster not $null

if ($Cluster -eq $null ) {

Write-Host "cannot be empty"
exit
}

### manually enter the portgroup name you want to remove

$OldPortGroupName = Read-Host "Enter PortGroup name"

## testing if $OldPortGroupName not $null

if ($OldPortGroupName -eq $null ) {

Write-Host "cannot be empty"
exit
}

```

```

### get the list of the VMs member of the cluster you selected before
$clusterVMs = Get-VM -Location $Cluster

### check for each VM if one nic is connected to the portgroup
if ($clusterVMs -ne $null) {
    foreach ($clusterVM in $clusterVMs) {
        $clusterVMNics = Get-NetworkAdapter -VM $clusterVM
        foreach ($clusterVMNic in $clusterVMNics) {
            if ($clusterVMNic.NetworkName -like $OldPortGroupName) {
                Write-Host "the VM ""$clusterVM"" as its adapter
                ""$clusterVMNic.Name"" connected to the portgroup that you want to
                remove"

                ### if on VM is connected to this port group, the variable
                "$isVMconnectedtoPortgroup" is updated to $true

                $isVMconnectedtoPortgroup = $true
            }
        }
    }
}

### check if the global variable "$isVMconnectedtoPortgroup" is set to
$true

if ($isVMconnectedtoPortgroup -eq $true) {
    Write-Host "Please unconnect the VMs listed above from the portgroup you
    want to remove.
    This will lead to errors during the removal process.

    Once the VMs will be unconnected from the portgroup retry!"

    Disconnect-VIServer -Server $vCenterserver -Confirm:$false

    exit
}

### list vswitch for each esx under the cluster to see if the portgroup is
present

$esxs = Get-VMHost -Location (get-cluster -name $Cluster)

if ($esxs -ne $null) {

```

```

foreach ($esx in $esxs) {

### get every vSwitch of the current ESX
$VMSwitchs = get-VirtualSwitch -VMHost $esx

    foreach ($VMSwitch in $VMSwitchs) {

### get every PortGroup of the current vSwitch
$LocalPortGroups = Get-VirtualPortGroup -VirtualSwitch $VMSwitch

        foreach ($LocalPortGroup in $LocalPortGroups) {

            ### for each portgroup check if it has the same name than
the one you want to remove

                if ($LocalPortGroup -like $OldPortGroupName) {

                    Write-Host "the portgroup ""$OldPortGroupName"" is present
in the $esx "

                    $isoldportpresent = $true

                    ### delete the portGroup

                    Write-Host "removing ""$OldPortGroupName"" to $esx"

                    $removePortGroup=Remove-VirtualPortGroup -VirtualPortGroup
$LocalPortGroup -Confirm:$false

                }

            }

        }

    } else {

Write-Host "No ESX present in the cluster ""$Cluster""!"

    }

### check if any portgroup has been deleted

if ($isoldportpresent -ne $true) {

Write-Host "the portgroup ""$OldPortGroupName"" is not present in the
cluster "

    }

"Finish"

Disconnect-VIServer -Server $vCenterserver -Confirm:$false

```