Function Get-DiskSpaceReport

{

param(

[String]$ComputerList,[int]$warning,[int]$critical,[String]$To,[String]$From,[string]$SMTPMail

)

$script:list = $ComputerList

$freeSpaceFileName = "C:\FreeSpace.htm"

if ($Warning -eq "$NULL")

{

$Warning=25

}

if ($critical -eq "$NULL")

{

$critical=15

}

$critical = $critical

$warning = $warning

New-Item -ItemType file $freeSpaceFileName -Force

# Getting the freespace info using WMI

#Get-WmiObject win32\_logicaldisk | Where-Object {$\_.drivetype -eq 3 -OR $\_.drivetype -eq 2 } | format-table DeviceID, VolumeName,status,Size,FreeSpace | Out-File FreeSpace.txt

# Function to write the HTML Header to the file

Function writeHtmlHeader

{

param($fileName)

$date = ( get-date ).ToString('yyyy/MM/dd')

Add-Content $fileName "<html>"

Add-Content $fileName "<head>"

Add-Content $fileName "<meta http-equiv='Content-Type' content='text/html; charset=iso-8859-1'>"

Add-Content $fileName '<title>DiskSpace Report</title>'

add-content $fileName '<STYLE TYPE="text/css">'

add-content $fileName "<!--"

add-content $fileName "td {"

add-content $fileName "font-family: Tahoma;"

add-content $fileName "font-size: 11px;"

add-content $fileName "border-top: 1px solid #999999;"

add-content $fileName "border-right: 1px solid #999999;"

add-content $fileName "border-bottom: 1px solid #999999;"

add-content $fileName "border-left: 1px solid #999999;"

add-content $fileName "padding-top: 0px;"

add-content $fileName "padding-right: 0px;"

add-content $fileName "padding-bottom: 0px;"

add-content $fileName "padding-left: 0px;"

add-content $fileName "}"

add-content $fileName "body {"

add-content $fileName "margin-left: 5px;"

add-content $fileName "margin-top: 5px;"

add-content $fileName "margin-right: 0px;"

add-content $fileName "margin-bottom: 10px;"

add-content $fileName ""

add-content $fileName "table {"

add-content $fileName "border: thin solid #000000;"

add-content $fileName "}"

add-content $fileName "-->"

add-content $fileName "</style>"

Add-Content $fileName "</head>"

Add-Content $fileName "<body>"

add-content $fileName "<table width='100%'>"

add-content $fileName "<tr bgcolor='#CCCCCC'>"

add-content $fileName "<td colspan='7' height='25' align='center'>"

add-content $fileName "<font face='tahoma' color='#003399' size='4'><strong>DiskSpace Report - $date</strong></font>"

add-content $fileName "</td>"

add-content $fileName "</tr>"

add-content $fileName "</table>"

}

# Function to write the HTML Header to the file

Function writeTableHeader

{

param($fileName)

Add-Content $fileName "<tr bgcolor=#CCCCCC>"

Add-Content $fileName "<td width='10%' align='center'>Drive</td>"

Add-Content $fileName "<td width='50%' align='center'>Drive Label</td>"

Add-Content $fileName "<td width='10%' align='center'>Total Capacity(GB)</td>"

Add-Content $fileName "<td width='10%' align='center'>Used Capacity(GB)</td>"

Add-Content $fileName "<td width='10%' align='center'>Free Space(GB)</td>"

Add-Content $fileName "<td width='10%' align='center'>Freespace %</td>"

Add-Content $fileName "</tr>"

}

Function writeHtmlFooter

{

param($fileName)

Add-Content $fileName "</body>"

Add-Content $fileName "</html>"

}

Function writeDiskInfo

{

param($fileName,$devId,$volName,$frSpace,$totSpace)

$totSpace=[math]::Round(($totSpace/1073741824),2)

$frSpace=[Math]::Round(($frSpace/1073741824),2)

$usedSpace = $totSpace - $frspace

$usedSpace=[Math]::Round($usedSpace,2)

$freePercent = ($frspace/$totSpace)\*100

$freePercent = [Math]::Round($freePercent,0)

if ($freePercent -gt $warning)

{

Add-Content $fileName "<tr>"

Add-Content $fileName "<td>$devid</td>"

Add-Content $fileName "<td>$volName</td>"

Add-Content $fileName "<td>$totSpace</td>"

Add-Content $fileName "<td>$usedSpace</td>"

Add-Content $fileName "<td>$frSpace</td>"

Add-Content $fileName "<td>$freePercent</td>"

Add-Content $fileName "</tr>"

}

elseif ($freePercent -le $critical)

{

Add-Content $fileName "<tr>"

Add-Content $fileName "<td>$devid</td>"

Add-Content $fileName "<td>$volName</td>"

Add-Content $fileName "<td>$totSpace</td>"

Add-Content $fileName "<td>$usedSpace</td>"

Add-Content $fileName "<td>$frSpace</td>"

Add-Content $fileName "<td bgcolor='#FF0000' align=center>$freePercent</td>"

#<td bgcolor='#FF0000' align=center>

Add-Content $fileName "</tr>"

}

else

{

Add-Content $fileName "<tr>"

Add-Content $fileName "<td>$devid</td>"

Add-Content $fileName "<td>$volName</td>"

Add-Content $fileName "<td>$totSpace</td>"

Add-Content $fileName "<td>$usedSpace</td>"

Add-Content $fileName "<td>$frSpace</td>"

Add-Content $fileName "<td bgcolor='#FBB917' align=center>$freePercent</td>"

# #FBB917

Add-Content $fileName "</tr>"

}

}

writeHtmlHeader $freeSpaceFileName

foreach ($server in Get-Content $script:list)

{

if(Test-Connection -ComputerName $server -Count 1 -ea 0) {

Add-Content $freeSpaceFileName "<table width='100%'><tbody>"

Add-Content $freeSpaceFileName "<tr bgcolor='#CCCCCC'>"

Add-Content $freeSpaceFileName "<td width='100%' align='center' colSpan=6><font face='tahoma' color='#003399' size='2'><strong> $server </strong></font></td>"

Add-Content $freeSpaceFileName "</tr>"

writeTableHeader $freeSpaceFileName

$dp = Get-WmiObject win32\_logicaldisk -ComputerName $server | Where-Object {$\_.drivetype -eq 3 }

foreach ($item in $dp)

{

Write-Host $item.DeviceID $item.VolumeName $item.FreeSpace $item.Size

writeDiskInfo $freeSpaceFileName $item.DeviceID $item.VolumeName $item.FreeSpace $item.Size

}

}

Add-Content $freeSpaceFileName "</table>"

}

writeHtmlFooter $freeSpaceFileName

Function sendEmail

{

param($from,$to,$subject,$smtphost,$htmlFileName)

[string]$receipients="$to"

$body = Get-Content $htmlFileName

$body = New-Object System.Net.Mail.MailMessage $from, $receipients, $subject, $body

$body.isBodyhtml = $true

$smtpServer = $MailServer

$smtp = new-object Net.Mail.SmtpClient($smtphost)

$validfrom= Validate-IsEmail $from

if($validfrom -eq $TRUE)

{

$validTo= Validate-IsEmail $to

if($validTo -eq $TRUE)

{

$smtp.Send($body)

write-output "Email Sent!!"

}

}

else

{

write-output "Invalid entries, Try again!!"

}

}

# Email our report out

function Validate-IsEmail ([string]$Email)

{

return $Email -match "^(?("")("".+?""@)|(([0-9a-zA-Z]((\.(?!\.))|[-!#\$%&'\\*\+/=\?\^`\{\}\|~\w])\*)(?<=[0-9a-zA-Z])@))(?(\[)(\[(\d{1,3}\.){3}\d{1,3}\])|(([0-9a-zA-Z][-\w]\*[0-9a-zA-Z]\.)+[a-zA-Z]{2,6}))$"

}

$date = ( get-date ).ToString('yyyy/MM/dd')

sendEmail -from $From -to $to -subject "Disk Space Report - $Date" -smtphost $SMTPMail -htmlfilename $freeSpaceFileName

}