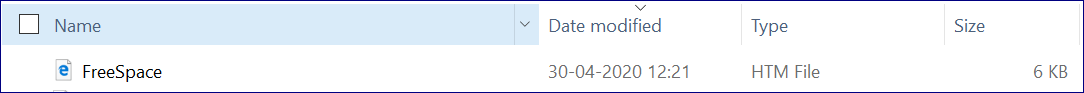
**PowerShell – Monitor Disk Space of a Group of SQL Servers – HTML Formatted Email Output - TESTED**

<https://sqlpowershell.blog/2013/10/01/powershell-script-to-monitor-disk-space-of-a-group-of-servers-html-formatted-email-output/>

**Please test below PowerShell Code in your Dev Environment first using below given steps.**

**NOTE: I’ve TESTED THIS PowerShell Code in my Local Lab VM Machine for few Test Servers and it worked successfully.**

**It created a HTML file named “FreeSpace” on my C:\ drive and even received a HTML mail to my mail id.**



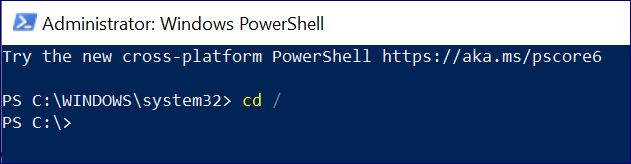
**Step-1: Create a notepad in C:\ drive as “Computer” and enter the list of Server Names (with FQDN) as follows.**

Server1.domain.com

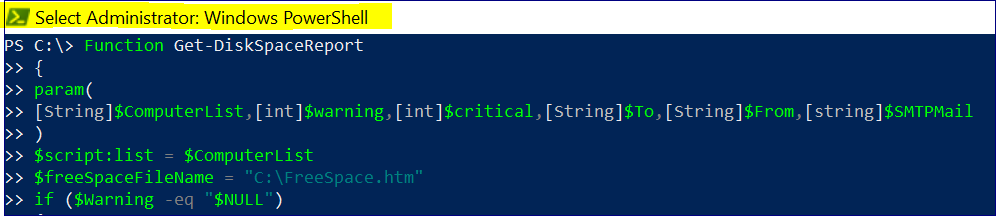
Server2.domain.com

Server3.domain.com

**Step-2: Open PowerShell window as “Run as Administrator” and TYPE cd /. This will open a new line as “PS C:\>”**



**Step-3: Now here run\execute the function Get-DiskSpaceReport  as shown in below screenshot.**



**Step-4: Post this execution completion, Run below script.**

PS:\>Get-DiskSpaceReport -ComputerList C:\computer.txt -warning 15 -critical 10 -To **praveensqldba12@gmail.com** -From **praveensqldba12@gmail.com** -SMTPMail **smtp.domain.com**

**Note: Change the Warning & Critical Values as per your requirement.**

This post explains how to monitor DiskSpace of a group of listed servers in a text file.

The function Get-DiskSpaceReport comprises of various cmdLets and function to monitor Disk Drives.

* Get-Win32LogicalDisks
* HTML Ouptut
* Email Address validation

You can customize it as per your requirement.The Function **Get-DiskSpaceReport** has six input parameters:-

1. ComputerList – List of Servers – Path of a input file where servers are listed
2. Warning – Warning Threshold – Default = 25%
3. Critical – Critical Threshold – Default =15%
4. SMTPMail – SMTP mail address
5. FromID – Valid Email ID
6. ToID – Valid Email ID

**Example 1:- Execute with default threshold values. By default, the threshold are set to 25(Warning) and 15(Critical)**

PS:\>Get-DiskSpaceReport -ComputerList c:\computer.txt  -To pjayaram@Appvion.com -From pjayaram@appvion.com -SMTPMail qqma01.ppp.com

[[Image](https://sqlpowershell.files.wordpress.com/2013/10/disk2.jpg)](https://sqlpowershell.files.wordpress.com/2013/10/disk2.jpg)

**Example 2:- Customize the default threshold values.**

PS:\>Get-DiskSpaceReport -ComputerList c:\computer.txt -warning 15 -critical 10 -To pjayaram@Appvion.com -From pjayaram@appvion.com -SMTPMail qqma01.ppp.com

[Image](https://sqlpowershell.files.wordpress.com/2013/10/disk3.jpg)

Below attached 2 scripts and Code copied below are all same.

**First create a function Get-DiskSpaceReport using the below code and do a function call. which is shown above**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

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}))$”  
}

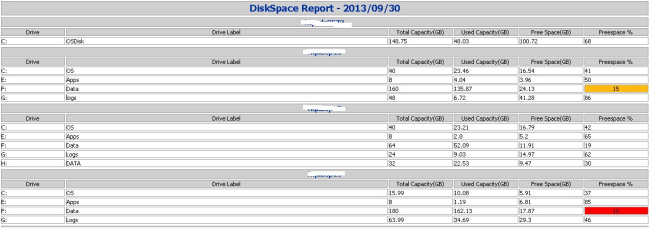
$date = ( get-date ).ToString(‘yyyy/MM/dd’)

sendEmail -from $From -to $to -subject “Disk Space Report – $Date” -smtphost $SMTPMail -htmlfilename $freeSpaceFileName

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Output :-**

[](https://sqlpowershell.files.wordpress.com/2013/10/disk1.jpg)

<https://sqlpowershell.blog/2013/10/01/powershell-script-to-monitor-disk-space-of-a-group-of-servers-html-formatted-email-output/>