**PowerShell – Monitoring Multiple SQL Services on Multiple Servers.**

<https://sqlpowershell.blog/2013/12/04/powershell-monitoring-multiple-service-on-a-group-of-servers/>

* The Author got a request to monitor multiple services and send an email to intended recipients.
* This post explains how to monitor multiple services on a group of servers.
* The function Get-ServiceStatusReport comprises of various cmdLets and function to monitor services on a list of servers.
* Get-Service
* HTML Output
* Email Address validation

The Function Get-ServiceStatusReport contains five parameters

1. ComputerList – List of Servers
2. ServiceName – Name of Services separated by comma
3. SMTPMail – SMTP mail address
4. FromID – Valid Email ID
5. ToID – Valid Email ID

**Function call –**

Get-ServiceStatusReport -ComputerList C:\server.txt -includeService  "MySQL","MpsSvc",

"W32Time" -To pjayaram@app.com -From pjayaram@ app.com -SMTPMail  app01. app.com

(or)

Get-ServiceStatusReport -ComputerList C:\server.txt -includeService  MySQL,MpsSvc,

W32Time -To pjayaram@app.com -From pjayaram@ app.com -SMTPMail  app01. app.com

**Open below attached script and save it as a powershell script.**

**Note: Below attached script and script kept in CODE Section are one and the same.**



Code:

Function Get-ServiceStatusReport

{

param(

[String]$ComputerList,[String[]]$includeService,[String]$To,[String]$From,[string]$SMTPMail

)

$script:list = $ComputerList

$ServiceFileName= "c:\ServiceFileName.htm"

New-Item -ItemType file $ServiceFilename -Force

# 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>Service Status 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='4' height='25' align='center'>"

add-content $fileName  "<font face='tahoma' color='#003399' size='4'><strong>Service Stauts 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'>ServerName</td>"

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

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

Add-Content $fileName "</tr>"

}

Function writeHtmlFooter

{

param($fileName)

Add-Content $fileName "</body>"

Add-Content $fileName "</html>"

}

Function writeDiskInfo

{

param($filename,$Servername,$name,$Status)

if( $status -eq "Stopped")

{

 Add-Content $fileName "<tr>"

 Add-Content $fileName "<td bgcolor='#FF0000' align=left ><b>$servername</td>"

 Add-Content $fileName "<td bgcolor='#FF0000' align=left ><b>$name</td>"

 Add-Content $fileName "<td bgcolor='#FF0000' align=left ><b>$Status</td>"

 Add-Content $fileName "</tr>"

}

else

{

Add-Content $fileName "<tr>"

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

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

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

Add-Content $fileName "</tr>"

}

}

writeHtmlHeader $ServiceFileName

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

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

 Add-Content $ServiceFileName "<td width='100%' align='center' colSpan=3><font face='tahoma' color='#003399' size='2'><strong> Service Details</strong></font></td>"

 Add-Content $ServiceFileName "</tr>"

 writeTableHeader $ServiceFileName

#Change value of the following parameter as needed

$InlcudeArray=@()

#List of programs to exclude

#$InlcudeArray = $inlcudeService

Foreach($ServerName in (Get-Content $script:list))

{

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

$service = Get-Service -ComputerName $servername

if ($Service -ne $NULL)

{

foreach ($item in $service)

 {

 #$item.DisplayName

 Foreach($include in $includeService)

     {

 write-host $inlcude

 if(($item.serviceName).Contains($include) -eq $TRUE)

    {

    Write-Host  $item.MachineName $item.name $item.Status

    writeDiskInfo $ServiceFileName $item.MachineName $item.name $item.Status

    }

}

    }

 }

}

}

Add-Content $ServiceFileName "</table>"

writeHtmlFooter $ServiceFileName

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}))$"

}

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!!"

}

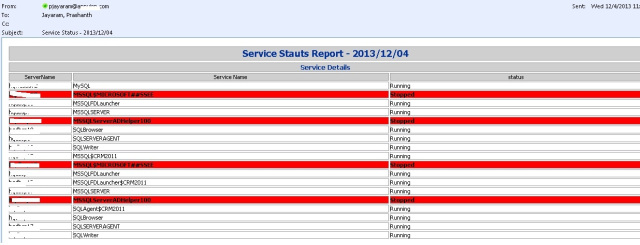
}

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

sendEmail -from $From -to $to -subject "Service Status - $Date" -smtphost $SMTPMail -htmlfilename $ServiceFilename

}

**Sample Output: -**

[](https://sqlpowershell.files.wordpress.com/2013/12/multipleservicesreport.jpg)