**PowerShell-Monitor SQL Services and Notify Stopped Services**

<https://gallery.technet.microsoft.com/scriptcenter/PowerShell-Monitor-Notify-a5fe1538>

This post explains how to monitor a multiple services on a list of servers and send alert when the given service is in stopped state. This script will allow you to read a list of servers from the input file and establish connection to all remote servers to pull service status. Notify users Only when the listed services are not running and send an alert to all intended recipients

The function Get- ServiceSQLAlert does the following action

1. Connect to remote server and loops through each server
2. Accepts more than one services
3. Concatenate the output into HTML tags
4. Sends notification only when there is any given services are stopped

 The Function Get- ServiceSQLAlert 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

**Sample Execution from PowerShell Window:**

**ps**:\>Get-ServiceSQLAlert -ComputerList C:\server.txt -includeService  SQLSERVERAGENT,MSSQLSERVER

-To pj@vion.com -**From** pj@vion.com -SMTPMail ap.com

**PowerShell Code: Attached Script & below copied code are one and the same.**



**Function** Get-ServiceSQLAlert

{

**param**(

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

)

$script:list = $ComputerList

#Make sure to check write acess on c:\ drive. if not, change the path

$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 Alert - $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")

{

 increment $global:a

**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>"

}

}

$global:a=0

**function** increment {

  $global:a++

}

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))

{

$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** 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)

$smtp.Send($body)

}

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

**if** ($global:a -ge 1)

{

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

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

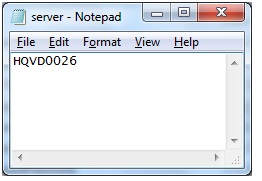
}

}

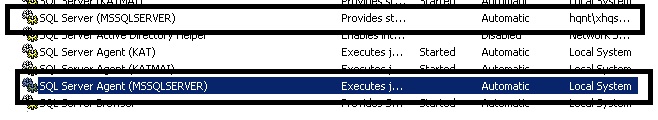
**Demonstration**:

Add the Servers in the Server.txt

 -ComputerList Parameter - C:\Server.txt consists of one server – HQVD0026.



**Stop SQL Server**



IncludeService  Parameter– In this case searching for SQLServer and Agent Services hence  the servicenames are given as its input **SQLSERVERAGENT,MSSQLSERVER**

-To Parameter – Valid email Id or Id’s

-From Parameter – Valid email id

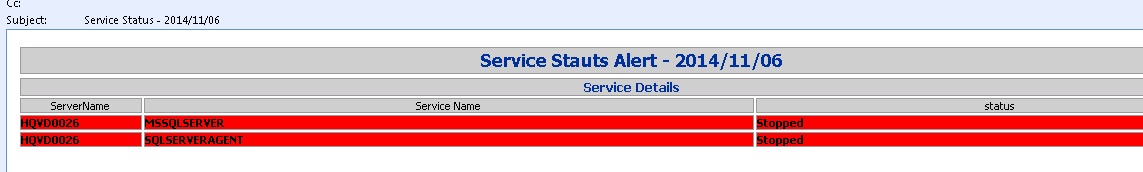
-SMTPMail  - Valid SMTP Mail Server name

**PowerShell**

**ps**:\>Get-ServiceSQLAlert -ComputerList C:\server.txt -includeService  SQLSERVERAGENT,MSSQLSERVER

-To pj@vion.com -**From** pj@vion.com -SMTPMail ap.com

**Output:**



Note: I’ve not tested above PowerShell Script execution in my Local Lab Machine. I apologize for this.