|  |
| --- |
| Opensky_white_tag |

# Technical Implement Document for SQL Server Health Check Script with Powershell

National Transport Authority

SQL Server Health Check Script Automation

**Prepared by** Ajatha Indira T

[aindira@openskydata.com](mailto:aindira@openskydata.com)



# Description

This document helps us to perform SQL Server Instance health check simultaneously on more than one server using Power shell, output is provided in HTML file/s and also emailed to concerned teams. For each SQL Server Instance Healthcheck script will generate two reports.

**Report 1**: This report will include all the parameters(**Section 1 to Section14)** mentioned below are sent to **Development** **Team**, This Report will have more detailed information regarding resource intensive queries and long running queries

**Report 2**: This report will include basic parameters(**Section 1 to Section7)** mentioned below are sent to **HelpDesk** **Team**, This Report will have database state is not online or not and other couple of check points before the business hours starts

# Code / Report Description

Health Check Report has various sections as listed below:

**Section 0**: This section defines our HTML look for the report.

**Section 1**: This section extracts the SQL Server properties like SQL Server Name, SQL Server Version, Edition, Machine Name and Node on which SQL Server is running and put in HTML fragment.

**Section 2**: This section extracts SQL Agent Service Information like Running or Not Running and puts in HTML fragment.

**Section 3**: This section extracts all database status, if any database state is not online, it will display here otherwise it will say all DBs are online and put the comments in HTML fragment.

**Section 4**: This section extracts any SQL job which has failed in the last 24 hours and put in HTML fragment.

**Section 5**: This section extracts all databases which has not been backed up (either differential or full) in the last 24 hours and puts the comments in HTML fragment.

**Section 6**: This section extracts all errors encountered in SQL Server error log in the last 24 hours and puts the comments in HTML fragment.

**Section 7**: This section extracts CPU information on the server, defines how many logical / physical CPUs we have and the hyper thread ratio we have on the server with how much memory is allocated to the Server and puts all the information in HTML fragment.

**Section 8**: This section extracts memory allocation on the server. It has three sections and puts that in HTML fragments:

* + Memory given and currently used by SQL Server Instance
  + Top 10 Memory Consuming SQL Objects in that particular SQL Instance
  + SQL Instance is having Memory Issues or not

**Section 9**: This section extracts Top 10 Long Running Queries in SQL Server and puts that in HTML fragments.

**Section 10**: This section extracts Top 10 CPU Consuming Query in SQL Server and put that in HTML fragments.

**Section 11**: This section extracts Top 10 IO Consuming Query in SQL Server and put that in HTML fragments.

**Section 12**: This section extracts CPU Pressure that exists on the server because of SQL Server or not and puts the comment in HTML fragment.

**Section 13**: This section extracts wait type events and their accumulative consumption in percentage on the SQL server and puts that data in HTML fragments.

**Section 14**: Accumulates all fragments, then converts the code in HTML and generates an HTML report which will be saved in location c:\dba\_temp and emailed to your email id (please edit your email-id, SMTP server in section 13).

# Prerequisites:

DB Mail needs to be configured and

Get SMTP Server Name and email-id and edit in the code in line 697

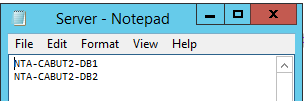


# Steps to Implement

**Steps 1:** Copy the **HealthCheckMonitoringPowershellScript.zip** file to **C:\** drive of one of the Servers and **unzip** the file

**Step 2:** **Open** the **servers.txt** filefrom **HealthCheckMonitoringPowershellScript** and add servers for reporting and alerting in separate rows

Example**:**



**Step3:** **Edit** the powershell script **AllPrametersHealthcheckReport\_Developers.ps1** in the **HealthCheckMonitoringPowershellScript** and configure the required values for **sender, recipients email and smtp server** and save the file

**Edit line number 697:**

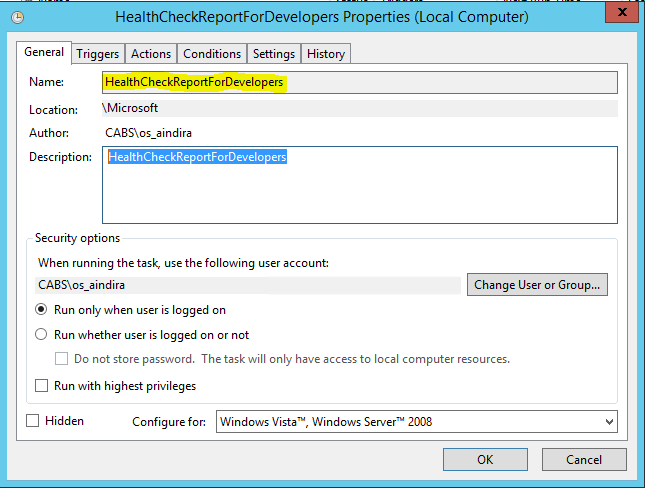
Send-MailMessage -to "RecipientEmailid1@domain.com "," RecipientEmailid2 @domain.com" -from "no\_replyuat2@nationaltransport.ie" -SmtpServer "NTA-CABUT2-APP2" -subject $subject -credential $anonCredentials -BodyAsHtml "$body" -Attachments $attachment #$OutputFile\_new

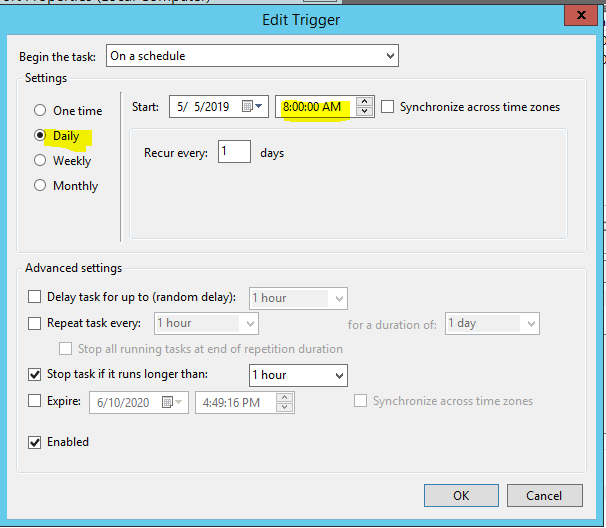
**Step 4:** **Edit** the powershell script **DailyCheckReport\_Helpdesk.ps1** in the **HealthCheckMonitoringPowershellScript** and configure the required values for **sender, recipients email and smtp server** and save the file

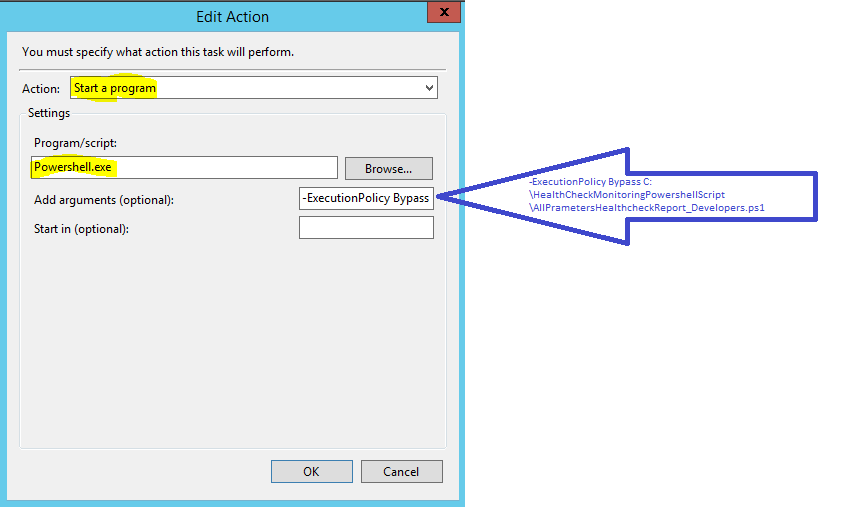
**Edit line number 697:**

Send-MailMessage -to " RecipientEmailid1@domain.com "," RecipientEmailid2 @domain.com” -from "no\_replyuat2@nationaltransport.ie" -SmtpServer "NTA-CABUT2-APP2" -subject $subject -credential $anonCredentials -BodyAsHtml "$body" -Attachments $attachment #$OutputFile\_new

**Step 5:** Create task **HealthCheckReportForDevelopers** under **TaskScheduler** as given below.

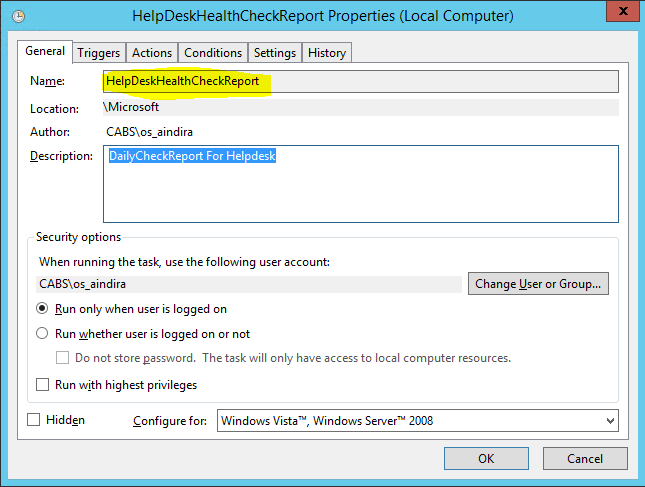


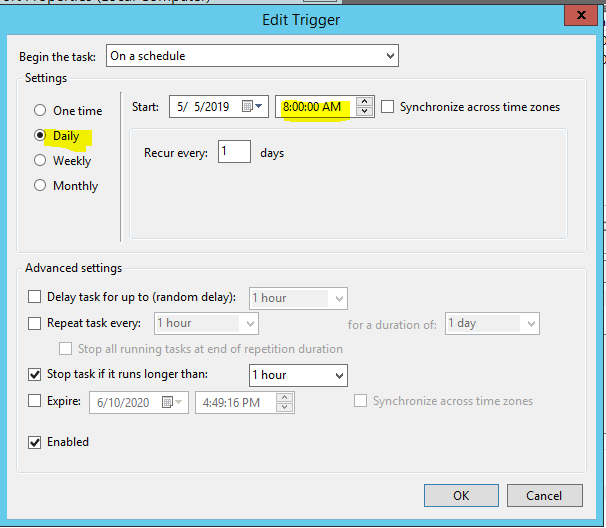


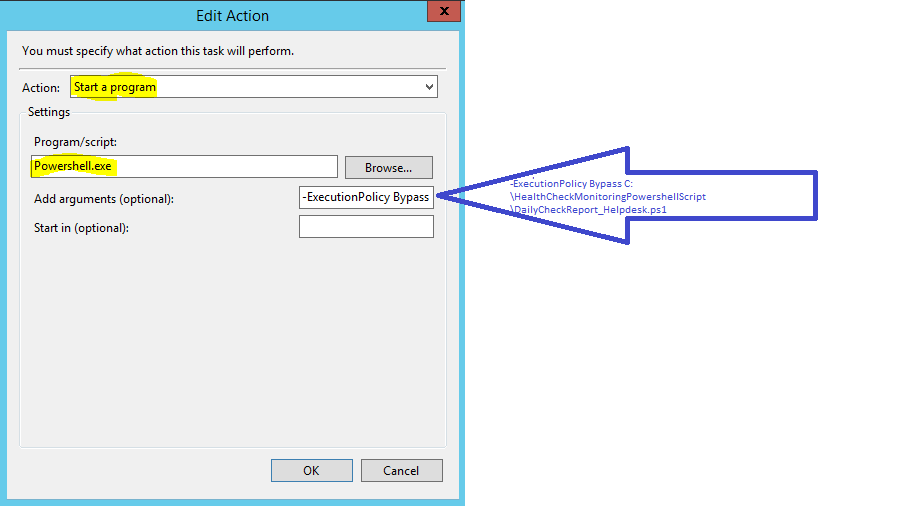


Next select the default setting and click ok

**Step 6:** Create task **DailyCheckReport For Helpdesk** under **TaskScheduler** as given below.







Next select the default setting and click ok

**Steps for adding New SQL Server instance for HealthCheck Report:**

Browse the **server.txt** file inside the folder **HealthCheckMonitoringPowershellScript** and append the sql server instance name.

Thank you!