

Find Database Type - Port - Remote Address.

With the below script `Check-DatabaseStatus.ps1` you can executed and find what **Database Type** the Server is using also the **Port** where is listening and the **Remote Address** that is configure to listen.

? One-Line Download & Execute :

```
Set-ExecutionPolicy -ExecutionPolicy Bypass -Scope Process -Force; Invoke-Expression (Invoke-WebRequest -Uri "https://docs.greenhome.stream/attachments/56" -UseBasicP) | Out-Null
```

? Script Features

- Clears the screen.
- Searches for **MSSQL** and **MySQL** services.
- Checks default listening ports: **1433 (MSSQL)** and **3306 (MySQL)**.
- Displays results in **color-coded output**.
- Shows a **Windows message box** with the final result.

? PowerShell Script

Save this script as `Check-DatabaseStatus.ps1` and run it with PowerShell:

```
# Step 1: Clear screen
Clear-Host

# Step 2: Setup variables
$foundDatabase = $null
$listenPort = $null
$remoteAddr = $null

# Function to write colored output
function Write-Color {
    param (
        [string]$Message,
        [ConsoleColor]$Color = "White"
    )
    Write-Host $Message -ForegroundColor $Color
```

```
}

# Step 3: Search for database type
Write-Color "Searching for Database Type..." -Color DarkYellow

# Check for MSSQL
$mssqlService = Get-Service | Where-Object { $_.Name -like "*sql*" -and $_.Status -eq
"Running" }

# Check for MySQL
$mysqlService = Get-Service | Where-Object { $_.Name -like "*mysql*" -and $_.Status -eq
"Running" }

if ($mssqlService) {
    $foundDatabase = "MSSQL"
    Write-Color "Database Found Type : MSSQL" -Color Green
} elseif ($mysqlService) {
    $foundDatabase = "MySQL"
    Write-Color "Database Found Type : MySQL" -Color Green
} else {
    Write-Color "Database NOT Found." -Color Red
}

# Leave a space line
Write-Host ""

# Step 4: Search for port
Write-Color "Searching for Port where database are listening by default..." -Color DarkYellow

if ($foundDatabase -eq "MSSQL") {
    $connection = Get-NetTCPConnection -State Listen -LocalPort 1433 -ErrorAction
SilentlyContinue
} elseif ($foundDatabase -eq "MySQL") {
    $connection = Get-NetTCPConnection -State Listen -LocalPort 3306 -ErrorAction
SilentlyContinue
} else {
    $connection = $null
}

if ($connection) {
```

```
$listenPort = $connection.LocalPort
$remoteAddr = $connection.LocalAddress

Write-Color "Found Database listening at port : $listenPort" -Color Green
Write-Color "For Remote Address : $remoteAddr" -Color Green
} else {
    Write-Color "NOT Found Database listening port." -Color Red
}

# Step 5: Show Windows popup message
Add-Type -AssemblyName PresentationFramework

if ($foundDatabase) {
    $popupMsg = "Found Database: $foundDatabase`nListen on Port: $listenPort`nRemote Address:
$remoteAddr"
    [System.Windows.MessageBox]::Show($popupMsg.Trim(), "Database Check Result", "OK",
"Information")
} else {
    [System.Windows.MessageBox]::Show("NO Database Found.!", "Database Check Result", "OK",
"Warning")
}
```

Revision #5

Created 2025-09-21 22:23:10 EEST by Green

Updated 2025-09-21 23:26:22 EEST by Green