How to Fix Microsoft Common Language Runtime Native Compiler High CPU Usage?

If you're experiencing high CPU usage by the Microsoft Common Language Runtime Native Compiler (mscorsvw.exe), it's likely because the .NET Framework is optimizing assemblies on your system. While this process typically runs in the background and doesn't consume too many resources, there are instances where it can cause high CPU usage or prolonged activity. Here's how to address the issue:

1. **Wait for the Process to Complete**:
    - In most cases, the mscorsvw.exe process will complete its optimization tasks and stop using high CPU resources on its own. It's a background process that runs when .NET assemblies need to be optimized. Depending on your system and the number of assemblies, this may take some time.

2. **Manually Trigger Optimization**:
    - You can manually trigger the optimization process by running the following command in an elevated (Run as Administrator) Command Prompt:

      ```
      ngen.exe executeQueuedItems
      ```
    - This command will force .NET to optimize queued assemblies, which can help reduce CPU usage. After running this command, monitor your CPU usage to see if it improves.

3. **Update .NET Framework**:
    - Ensure that your .NET Framework is up-to-date. Sometimes, updating to the latest version can resolve performance issues.

4. **Check for Malware**:
    - High CPU usage can be caused by malware or viruses. Run a full system scan using a reputable antivirus program to make sure your system is clean.

5. **Check for Software Conflicts**:
    - Certain software applications or third-party libraries may conflict with the .NET optimization process. Try to identify if any specific application or recent software installations coincide with the high CPU usage. Temporarily uninstalling or disabling such software may help.

6. **Optimize .NET Assemblies Manually**:
    - If you have a specific application that is causing high CPU usage due to .NET optimization, you can try optimizing the assemblies for that application manually using the `ngen.exe` tool. Here's an example command:
      ```
      ngen.exe install "C:\Path\To\Your\Application.exe"
      ```
    - Replace `"C:\Path\To\Your\Application.exe"` with the actual path to your application's executable file.

7. **Adjust .NET Optimization Priority**:
    - You can adjust the priority of the .NET optimization process to make it consume fewer CPU resources. Run the following command with administrator privileges:
      ```
      ngen.exe queue [install|uninstall] [assemblyName]
      ```
    - Replace `[assemblyName]` with the name of the assembly you want to optimize. Use `install` to increase priority and `uninstall` to decrease priority.

8. **Consider .NET Core**:
    - If the high CPU usage issue persists and you can use .NET Core for your applications, you might consider migrating to .NET Core or .NET 5+ as it has a more efficient compilation system.

However considering you have already purchased Restoro and would like us to assist you in doing that utilizing a remote session, please contact our support and they will be happy to help you.

- Email: support@restoro.com

Revision #1

Created 8 months ago by Ghicel Santos

Updated 8 months ago by Ghicel Santos