

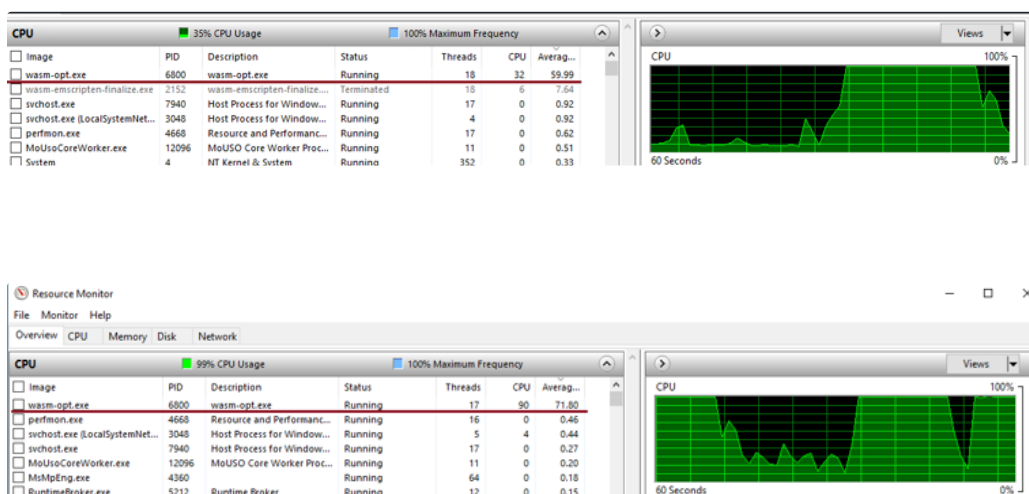
## Reduce CPU Usage On CI Pipeline

### Explanation of Update:

#### Summary:

- This update follows the recent merge of (Jira ticket at the bottom of this page) and aims to reduce VM CPU usage. By setting the WebAssembly optimization level to 0 in the CI pipeline **(not in the CD pipeline)**, the WebGL build test stage's CPU usage, which previously reached 100% for 2-3 minutes or more, will be significantly reduced. This change will allow the VM to handle more pipeline operations efficiently.

#### Current Situation:



- The above screenshots are the VM's Resource Monitor during the WebGL Build Stage in the pipeline, showing CPU usage. It was observed that a single executable file, wasm-opt.exe, was consuming 50-75% of CPU usage. Depending on the project size, in the case of the Paramedic-Ambulance project, the CPU usage reached 100% due to wasm-opt.exe, lasting for at least 2 minutes or longer. This situation could severely impact the VM's capability and performance if multiple pipelines accidentally run wasm-opt.exe simultaneously. This could affect the processes of other pipelines or even cause the VM server to crash.

#### Direction/Solution:

- To reduce or eliminate the CPU usage of `wasm-opt.exe`, I investigated what `WASM` means in Unity using the [Unity - Manual: Web native plug-ins for Emscripten](#). I found that `WASM` is related to cross-compiling WebAssembly using the Emscripten compiler toolchain.

Several files with `wasm` in their name are executed during the WebGL build process, with `wasm-opt.exe` specifically involved in optimizing code. According to the [Optimizing Code — Emscripten 4.0.23-git \(dev\) documentation](#), it is possible to set the optimization level to 0. After confirming that this level does not affect the WebGL build, I adjusted the optimization level to 0 for the CI pipeline, as the main concern here is whether the WebGL build succeeds or fails.

### Result:

- When running the CI pipeline, I observed significantly reduced total CPU usage, with `wasm-opt.exe` not being executed in the CI pipeline but only in the CD pipeline. Only `IL2CPP.exe` caused a brief spike of 70-90% CPU usage during the entire pipeline operation for about 1-2 seconds. For the rest of the process, the CPU usage remained around 10-20%.
- `IL2CPP.exe` converts the .NET game code in C# scripts to Wasm.

### Expectation / Next Steps:

- Since the CI pipeline runs more frequently than the CD pipeline, reducing the CPU usage in the CI pipeline is expected to have a significant positive impact on the VM's capability and performance. With the improved CPU usage, it will be possible to implement multi-branch pipelines or configure the pipeline in various styles according to the project's and team's development style.

### Jira Ticket

The Issue related to same as [Implement one single project files for all branch](#)