

Using Native Tools on Machine

The replacement for native tools bootstrapping is having all tools needed come pre-installed on the build machines. This document describes how to onboard to this new process.

This document only applies to Windows machines as Linux machines already come with native tools installed via package manager.

Steps

1. Before calling `InitializeNativeTools`, `init-tools-native.ps1`, or `eng/common's build.ps1` in PowerShell, set `$env:NativeToolsOnMachine = $True`. Alternatively, simply set it as an environment variable when calling a template such as `jobs.yml` in YAML.
2. Modify your `global.json's` `native-tools` section to change the version of your tools to one of the following values:
 - `latest` (e.g. `"cmake": "latest"`) – Grabs the latest version of the tool on the machine; this should be what you use in the majority of cases
 - Major version (e.g. `"cmake": "3"`) – Grabs the latest minor & patch version of a particular major version on the machine; this is useful for artifacts you want to pin to a specific major version
 - Minor version (e.g. `"python": "3.10"`) – Grabs the latest patch version of a particular minor version on the machine; this is useful for artifacts you want to pin to a specific minor version (such as Python)
3. Adjust any usage of the artifacts on the machines in your scripts. The artifacts you specify in your `global.json` are promoted to the path, so in general simply calling `cmake` will work. However, if you need the specific locations of tools, `InitializeNativeTools` will return those to you as a dictionary, e.g.:

```
pwsh $nativeToolsLocs = InitializeNativeTools $cmakeLoc = $nativeToolsLocs["cmake"]
```
4. Switch the image you're using from a `build.*` image to the equivalent `windows.*` image (the full list of which can be found on helix.dot.net). For example, if you were previously using `build.windows.amd64.vs2022`, switch to `windows.amd64.vs2022`.

Once you've executed these steps, you'll be using the native tools installed on the machines.

Documentation Updates

After switching, the local dev experience will change as devs will be expected to install dependencies ahead of time. Consider creating a document similar to this [dotnet/runtime requirements doc](#).

Was this helpful?  