Maintaining the build files

This document explains how to maintain the build files in the codebase.

Overview

On how to build the Node.js core, see Building Node.js.

There are three main build files that may be directly run when building Node.js:

- `configure`: A Python script that detects system capabilities and runs GYP. It generates `config.gypi` which includes parameters used by GYP to create platform-dependent build files. Its output is usually in one of these formats: Makefile, MSbuild, ninja, or XCode project files (the main Makefile mentioned below is maintained separately by humans). For a detailed guide on this script, see configure.
- `vcbuild.bat`: A Windows Batch Script that locates build tools, provides a subset of the targets available in the Makefile, and a few targets of its own. For a detailed guide on this script, see vcbuild.bat.
- `Makefile`: A Makefile that can be run with GNU Make. It provides a set of targets that build and test the Node.js binary, produce releases and documentation, and interact with the CI to run benchmarks or tests. For a detailed guide on this file, see Makefile.

On Windows `vcbuild.bat` runs configure before building the Node.js binary, on other systems `configure` must be run manually before running `make` on the `Makefile`.

vcbuild.bat

To see the help text, run `.\vcbuild help`. Update this file when you need to update the build and testing process on Windows.

configure

The `configure` script recognizes many CLI flags for special build formulas. Many are not represented by `vcbuild` shortcuts, and need to be passed either by:

- Calling `python configure --XXX --YYY=PPPP` directly, followed by `vcbuild noprojgen`
- Setting `set config_flags=--XXX --YYY=PPPP` before calling `vcbuild`

To see the help text, run `python configure --help`. Update this file when you need to update the configuration process.

Makefile

To see the help text, run `make help`. This file is not generated, it is maintained by humans. This is not usually run on Windows, where vcbuild.bat is used

instead.

Options

- `-j <n>`: number of threads used to build the binary. On the non-CI targets, the parallel tests will take up all the available cores, regardless of this option.