How to set up Eclipse for V8 development

Note: As of 2020 or so, Eclipse's C++ indexer tends to get confused and crash for V8. At the cost of retraining your muscle memory for shortcuts, a good alternative is Visual Studio Code + clangd. Consider this document here deprecated in favor of that.

(Last update: 2019-10-15, using Eclipse "2019-09")

If you want to restart from scratch: Delete ~/.eclipse, your eclipse workspace, and from your V8 checkout .project, .cproject, .settings/, and .pydevproject (if you've used PyDev before).

1. Download "Eclipse IDE for C/C++ Developers" from http://www.eclipse.org/downloads/eclipse-packages/
2. Extract it. Suggestion: extract to ~/eclipse
3. Edit the end of ~/eclipse/eclipse.ini to read:
   ```
   -XX:MaxPermSize=2048m
   -Xms1G
   -Xmx16G
   ```
4. Open eclipse, create a new workspace. Suggestion: ~/v8/eclipse-workspace
5. Help > Install New Software, Work with: Photon, install: JavaScript development tools
6. Help > Eclipse Marketplace...,
   - Search for "CppStyle" and install it
   - Optional: search for "PyDev" and install it, if you want to use it for Python scripts in <v8>/tools/
   - Note: in Eclipse Photon, the Marketplace only opens if "Enable theming" is activated in Window > Preferences > General > Appearance
7. Adjust Window > Preferences:
   - General > Startup and Shutdown: check "Refresh workspace on startup"
   - General > Workspace: uncheck "Build automatically"
   - General > Editors > Text Editors:
     - check "Insert spaces for tabs"
     - check "Show print margin"
     - [optional: save screen space] uncheck "Show line numbers"
   - General > Editors > Text Editors > Spelling: uncheck "Enable spell checking"
   - General > Appearance > Colors and Fonts > Basic > Text font:
     - choose what you like; with "DejaVu Sans Mono" in size 9 you can fit 4x 80 columns side-by-side on a 30" monitor (or 3x 80 on 24")
   - C/C++ > Build > Settings:
     - On "Discovery" tab, select "CDT GCC Built-in Compiler Settings"
     - Add -std=c++11 to the command line
   - C/C++ > Code Style > Formatter:
     - change "Code Formatter" from "[builtin]" to "CppStyle (clang-format)"

- C/C++ > CppStyle:
  - Clang-format path: <v8>/buildtools/linux64/clang-format
- C/C++ > Editor:
  - "When formatting code with an empty selection:" -> "Format the statement on the current line" (with Ctrl+Shift+F)
- C/C++ > Editor > Scalability:
  - uncheck "Alert me when...",
  - limit: 20K lines,
  - uncheck all "Scalability mode settings"
- [optional: save screen space] C/C++ > Editor > Folding:
  - uncheck "Enable folding"
- C/C++ > Indexer > Cache limits:
  - relative 30%, absolute 6400 MB
- JavaScript > Code Style > Formatter:
  - create a "New…" profile, change indentation:
    - Tab policy: Spaces only
    - Indentation size: 2
  - change more formatting settings if you wish (or format manually while writing code)

8. File > New > Makefile Project with Existing Code, select code directory,
   - Toolchain: Linux GCC
9. Define some resource filters to make indexing faster. Use the “Project Explorer” view, start with the top-level project entry to define the first level of filters (right-click -> “Properties”, “Resource”, “Resource Filters” -> “Include only”, “Folders”, “Project Relative Path matches:”), then recursively right-click the respective subdirectories and choose “Properties” to add lower-level filters (“Include only”, “Name matches”) until you get the following tree of included directories:
   - include
   - out
     - x64.release
       - gen
   - src
   - samples
   - test
   - testing
   - third_party
     - googletest
       - src
         a. googlemock
         b. googletest
     - wasm-api
10. Adjust Project > Properties:

- C/C++ General > Paths and Symbols > Symbols: Select language "GNU C++", add symbols (set each value to 1):
  - DEBUG
  - ENABLE_DEBUGGER_SUPPORT
  - ENABLE_DISASSEMBLER
  - ENABLE_EXTRA_CHECKS
  - ENABLE_GDB_JIT_INTERFACE
  - ENABLE_LOGGING_AND_PROFILING
  - ENABLE_MINOR_MC
  - ENABLE_SLOW_DCHECKS
  - OBJECT_PRINT
  - V8_DOUBLE_FIELDS_UNBOXING
  - V8_EMBEDDED_BUILTINS
  - V8_HAS_CXX11_FINAL
  - V8_HOST_ARCH_64_BIT
  - V8_INTL_SUPPORT
  - V8_TARGET_ARCH_ARM
  - V8_TARGET_ARCH_ARM64
  - V8_TARGET_ARCH_IA32
  - V8_TARGET_ARCH_MIPS
  - V8_TARGET_ARCH_MIPS64
  - V8_TARGET_ARCH_PPC
  - V8_TARGET_ARCH_S390
  - V8_TARGET_ARCH_X64
  - V8_TRACE_MAPS
  - V8_USE_ADDRESS_SANITIZER
  - VERIFY_HEAP
- C/C++ General > Preprocessor Include
  - Select “Providers” tab
  - Select “CDT GCC Built-in Compiler Settings [Shared]”
  - Select “Use global provider shared between projects” checkbox