

Debugging in VSCode

This guide goes over how to set up VSCode debugging for both your own Electron project as well as the native Electron codebase.

Debugging your Electron app

Main process

1. Open an Electron project in VSCode.

```
$ git clone git@github.com:electron/electron-quick-start.git
$ code electron-quick-start
```

2. Add a file `.vscode/launch.json` with the following configuration:

```
{
  "version": "0.2.0",
  "configurations": [
    {
      "name": "Debug Main Process",
      "type": "node",
      "request": "launch",
      "cwd": "${workspaceFolder}",
      "runtimeExecutable": "${workspaceFolder}/node_modules/.bin/electron",
      "windows": {
        "runtimeExecutable": "${workspaceFolder}/node_modules/.bin/electron.cmd"
      },
      "args": ["."],
      "outputCapture": "std"
    }
  ]
}
```

3. Debugging Set some breakpoints in `main.js`, and start debugging in the Debug View. You should be able to hit the breakpoints.

Here is a pre-configured project that you can download and directly debug in VS-Code: <https://github.com/octref/vscode-electron-debug/tree/master/electron-quick-start>

Debugging the Electron codebase

If you want to build Electron from source and modify the native Electron codebase, this section will help you in testing your modifications.

For those unsure where to acquire this code or how to build it, Electron's Build Tools automates and explains most of this process. If you wish to manually set

up the environment, you can instead use these build instructions.

Windows (C++)

1. Open an Electron project in VSCode.

```
$ git clone git@github.com:electron/electron-quick-start.git
$ code electron-quick-start
```

2. Add a file `.vscode/launch.json` with the following configuration:

```
{
  "version": "0.2.0",
  "configurations": [
    {
      "name": "(Windows) Launch",
      "type": "cppvsdbg",
      "request": "launch",
      "program": "${workspaceFolder}\\out\\your-executable-location\\electron.exe",
      "args": ["your-electron-project-path"],
      "stopAtEntry": false,
      "cwd": "${workspaceFolder}",
      "environment": [
        {"name": "ELECTRON_ENABLE_LOGGING", "value": "true"},
        {"name": "ELECTRON_ENABLE_STACK_DUMPING", "value": "true"},
        {"name": "ELECTRON_RUN_AS_NODE", "value": ""},
      ],
      "externalConsole": false,
      "sourceFileMap": {
        "o:\\": "${workspaceFolder}",
      },
    },
  ],
}
```

Configuration Notes

- `cppvsdbg` requires the built-in C/C++ extension be enabled.
- `${workspaceFolder}` is the full path to Chromium's `src` directory.
- `your-executable-location` will be one of the following depending on a few items:
 - **Testing:** If you are using the default settings of Electron's Build-Tools or the default instructions when building from source.
 - **Release:** If you built a Release build rather than a Testing build.
 - **your-directory-name:** If you modified this during your build process from the default, this will be whatever you specified.
- The `args` array string `"your-electron-project-path"` should be the absolute path to either the directory or `main.js` file of the Electron project

you are using for testing. In this example, it should be your path to `electron-quick-start`.

3. Debugging Set some breakpoints in the .cc files of your choosing in the native Electron C++ code, and start debugging in the Debug View.