

# WebAssembly local build

## Installing WebAssembly (WASM) /Emscripten

### 1. qt maintenance tool

1. -> install component
2. qt->qt x.x.x -> webassembly
3. enable multi and single threaded and install

### 2. emscripten compiler

follow install instructions from [https://emscripten.org/docs/getting\\_started/downloads.html](https://emscripten.org/docs/getting_started/downloads.html)

```
git clone https://github.com/emscripten-core/emsdk.git
cd emsdk
git pull
# note version is dependent on qt version -> look for the correct
version/check .github/workflows/deploy.yml in renderer repo
./emsdk install 3.1.56
./emsdk activate 3.1.56
source ./emsdk_env.sh
```

### 3. enable WebAssembly build in Qt Creator

1. go to Projects (tab on left side)
2. there should be a WebAssembly "Build and Run" option that is grayed out  
-> click on it to activate (only the Release build should work)
3. Build the release version
4. after build is complete: the application output should show you the link/port of the web address where you can see it and open the website automatically

## Shader HotReloading

1. In the build settings (Project tab on the left side), set the ALP\_ENABLE\_SHADER\_NETWORK\_HOTRELOAD option to ON
2. go into the gl\_engine/shaders directory
3. create a server.py file with the following content  
<https://gist.github.com/razor-x/9542707>

```

import os
import sys
import http.server
import socketserver

PORT = 5500

class HTTPRequestHandler(http.server.SimpleHTTPRequestHandler):
    def end_headers(self):
        self.send_header('Access-Control-Allow-Origin', '*')
        http.server.SimpleHTTPRequestHandler.end_headers(self)

def server(port):
    httpd = socketserver.TCPServer(('', port), HTTPRequestHandler)
    return httpd

if __name__ == "__main__":
    port = PORT
    httpd = server(port)
    try:
        print("\nserving from build/ at localhost:" + str(port))
        httpd.serve_forever()
    except KeyboardInterrupt:
        print("\n...shutting down http server")
        httpd.shutdown()
        sys.exit()

```

4. Start the server in the terminal with the following command

```
python3 server.py
```

5. after this you can start the WebAssembly build and change the code in the shaders as usual.

The hot reload works as usual by pressing F6 in the application web browser tab. All shader errors should be visible in the "browser developer tools" -> console tab.