Building femm-qt on macOS

D. Meeker

May 19, 2026

1 Overview

femm-qt is a cross-platform port of FEMM (Finite Element Method Magnetics) using Qt 6 for the GUI and CMake/Ninja for the build system. This document describes how to build it from source on macOS (Intel or Apple Silicon).

The build produces six executables:

- `femm_gui` — the main GUI application
- `femm_headless` — headless IPC server (no GUI)
- `fkn` — magnetics solver
- `belasolv` — electrostatics solver
- `hsolv` — heat flow solver
- `csolv` — current flow solver

2 Prerequisites

2.1 Xcode Command Line Tools

Install the Apple compiler toolchain (Clang, make, git, etc.) from Terminal:

```
xcode-select --install
```

Follow the prompts to complete the installation. This provides `clang++`, `git`, and standard system headers.

2.2 Homebrew

If not already installed, install the Homebrew package manager:

```
/bin/bash -c "$(curl -fsSL \
    https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

Follow the post-install instructions to add Homebrew to your `PATH`.

2.3 CMake and Ninja

```
brew install cmake ninja
```

2.4 Qt 6

Install Qt 6 via Homebrew:

```
brew install qt@6
```

Homebrew installs Qt to a prefix such as `/opt/homebrew/opt/qt@6` (Apple Silicon) or `/usr/local/opt/qt@6` (Intel). Note this path; it is needed for the CMake configure step.

To find your Qt prefix:

```
brew --prefix qt@6
```

2.5 OpenMP (Optional)

macOS's system Clang does not ship with OpenMP support. Without OpenMP, the solvers build and run correctly but are single-threaded. To enable multithreaded solving:

```
brew install libomp
```

CMake will automatically detect Homebrew's `libomp` and configure the necessary flags. If it is not found, CMake prints a notice and proceeds without it.

3 Getting the Source

If you have a source zip, unzip it:

```
unzip femm-qt-src.zip
cd femm-qt
```

Or, if the source is hosted on GitHub, clone the repository:

```
git clone <repository-url> femm-qt
cd femm-qt
```

4 Building

From the `femm-qt` directory, run the build script:

```
./macos_build.sh
```

This checks that the build tools are present, locates Qt 6 via Homebrew, configures the project with CMake, and builds it with Ninja. All executables are placed in `build/bin/`.

To wipe the build directory and rebuild from scratch, pass `clean`:

```
./macos_build.sh clean
```

If the build fails because CMake cannot find Qt 6, see *Manual configuration* under Troubleshooting.

5 Running

Launch the GUI:

```
./bin/femm_gui
```

Or the headless server:

```
./bin/femm_headless
```

5.1 Setting FEMM_DIR

The language bindings (OctaveFEMM, MathFEMM, pyFEMM, JuliaFEMM) locate the FEMM executables via the `FEMM_DIR` environment variable. Set it to the `bin` directory:

```
export FEMM_DIR=/path/to/femm-qt/build/bin
```

Add this to your `~/.zshrc` (or `~/.bash_profile`) to make it persistent.

6 Troubleshooting

6.1 Manual configuration

The build script just wraps the standard CMake and Ninja steps. To run them by hand:

```
mkdir build && cd build
cmake .. -G Ninja \
    -DCMAKE_BUILD_TYPE=Release \
    -DCMAKE_PREFIX_PATH="$(brew --prefix qt@6)"
ninja
```

6.2 Qt not found

If CMake reports that it cannot find Qt6, verify the prefix path:

```
brew --prefix qt@6
```

Pass the result to `-DCMAKE_PREFIX_PATH`.

6.3 OpenMP not found

If CMake prints "OpenMP not found," the solvers will still build and run but will be single-threaded. To enable OpenMP, install `libomp` via Homebrew (`brew install libomp`) and re-run the CMake configure step.

6.4 Permission denied on first run

macOS Gatekeeper may block unsigned executables. If you see a "cannot be opened because the developer cannot be verified" dialog, go to System Settings → Privacy & Security and click "Allow Anyway," or run:

```
xattr -dr com.apple.quarantine bin/femm_gui
```