

# Using the NFE electrical toolkit with Claude

Claude, our AI assistant, can read an architect's floor plan, check our electrical drawings and lay out devices through the **NFE electrical toolkit**, an open-source program NFE wrote that runs on your own computer. Claude names the rooms, device types and rules; the toolkit works out every position and writes only to a separate overlay file on NFE layers. It never writes to the architect's drawing. This page covers installing the toolkit, connecting it to Claude, the propose-review-approve workflow and plotting sheet PDFs. For what it achieved on LSHA, see the [phase 1 pilot](#).

## What Claude can do with it

- **Read a floor plan.** Open a DWG or DXF, detect its real units, pick out the plotted copy of the architect's plan, and index every electrical symbol.
- **Find the rooms**, including mirrored apartments, and show you a room map.
- **Check our drawings** (read-only): title blocks and drawing numbers, legends against the symbols on each sheet, and device counts per room.
- **Lay out devices** by rule in one room, apply a whole rule set, copy a reviewed apartment onto its mirror image, or draw a schematic circuit run.
- **Compare** a layout with the engineer's existing one, and show you review images.
- **Plot sheets to PDF** at true size. DXF + PDF is NFE's standard deliverable.

## One-time setup

You need three things:

1. **Claude Code or the Claude desktop app** on your own computer, signed in with your NFE Team seat. If you don't have a seat yet, ask Aaron. The toolkit has been tested on macOS; Windows is untested so far.
2. **Python 3.11 or newer, and git.**
3. **LibreDWG**, the open-source DWG converter. On a Mac with Homebrew, run `brew install libredwg`. On Linux, install your distribution's `libredwg` package. On Windows, LibreDWG publishes its own builds; if they are not on your PATH, point the toolkit at them with the `NFE_DWG2DXF`, `NFE_DXF2DWG` and `NFE_DWGREAD` environment variables.

## Install

In Terminal, run these one at a time:

```
cd ~
git clone https://github.com/Nearly-Free-Energy/nfe-electrical-toolkit.git
cd nfe-electrical-toolkit
python3 -m venv .venv
.venv/bin/python -I -m pip install -e ".[test]"
.venv/bin/python -I -m pytest -q
```

The last command runs the tests. 12 should pass; the 4 that need the LSHA F6 drawing are skipped unless you have it. Always run the toolkit's Python with `-I` (isolated mode), so that nothing in a drawing folder can be imported by accident.

Your drawings and outputs go in a work folder outside the repository: `~/nfe-electrical-work` in the examples below. Keeping it outside means client drawings can never be committed by mistake.

## Connect Claude Code

Run this once in Terminal. `--scope user` makes the tools available in every project:

```
claude mcp add nfe-electrical --scope user -- ~/nfe-electrical-toolkit/.venv/bin/python -I -m
nfe_electrical.mcp_server --workdir ~/nfe-electrical-work
```

Then start a new Claude Code session, because MCP servers only connect when a session starts.

`claude mcp list` should show `nfe-electrical` as connected.

## Connect Claude Desktop

Add the server to Claude Desktop's config file, `claude_desktop_config.json`:

- macOS: `~/Library/Application Support/Claude/claude_desktop_config.json`
- Windows: `%APPDATA%\Claude\claude_desktop_config.json`

Use full paths, replacing `YOUR-NAME` with your user name:

```
{
  "mcpServers": {
    "nfe-electrical": {
      "command": "/Users/YOUR-NAME/nfe-electrical-toolkit/.venv/bin/python",
      "args": ["-I", "-m", "nfe_electrical.mcp_server", "--workdir", "/Users/YOUR-NAME/nfe-
electrical-work"]
    }
  }
}
```

```
}  
}
```

If the file already has an `mcpServers` section, add the `nfe-electrical` entry inside it instead of starting a second one. Quit Claude Desktop completely and reopen it. On Windows, the command is `...\nfe-electrical-toolkit\.venv\Scripts\python.exe` (untested).

## Check it works

Ask Claude: "Using the NFE electrical toolkit, list the device types you can place." It should answer from the toolkit's `list_device_types` tool.

# Working on a floor: propose, review, approve

Claude doesn't write anything until you approve it. Every tool that adds to the overlay does a **dry run** unless told otherwise. A typical session:

1. **Open the floor.** "Open ~/Drawings/LSHA/F6.dwg as F6." The toolkit copies the drawing into `~/nfe-electrical-work/input/F6/`, makes the copy read-only and converts it. Your original is never opened for writing. Check the units and the plotted base copy it reports. If the job isn't under BS 7671, say so: "...as F6, under IEC 60364."
2. **Find the rooms.** "Find the rooms and show me the room map." Check that every room you need was found and named. Labels the toolkit couldn't bound are listed.
3. **Run the checks.** "Run the checks and list the errors." Title blocks, legends and device counts. The checks only read.
4. **Ask for a proposal (dry run).** "Propose bedside sockets for both bedrooms with pair:bed\_head", or "Apply the rules in ~/nfe-electrical-toolkit/rules/bs7671/apartment\_type\_a\_small\_power.yaml to unit W, as a dry run." Claude reports what would go where, room by room, with warnings such as "only 1 of 2 positions fit the rule". Nothing is written.
5. **See it drawn.** A dry run gives a list, not a picture. To see the proposal, ask Claude to write it to a trial overlay (for example `F6_trial`) and show you the render: proposed devices in magenta over the grey plan, the engineer's existing symbols in blue. Your real overlay is untouched. A trial overlay is just a file in `~/nfe-electrical-work/F6/overlays/`; delete it when you no longer need it.
6. **Approve, and only then write.** "Approved. Write it to F6\_small\_power." Claude repeats the call with `dry_run=false`. To take something out, ask Claude to remove those items. Removal happens straight away, but only ever from the overlay.
7. **Compare and validate.** "Compare unit W with the engineer's sockets" pairs each new socket with the nearest existing one. "Validate the F6\_small\_power overlay" re-reads it strictly before you hand it over.

## 8. Plot the sheets (next section).

Rules for single placements include `pair:bed_head` (two either side of the bed head), `wall:media_wall:3` (three on the media wall), `perimeter:2`, `beside_door`, `centre` and `corners`. Rule sets live in the repository under `rules/bs7671/`, and the reasoning behind them is in the [Small-power rules playbook, BS 7671 \(draft\)](#). In mirrored apartments, design and review one unit, then ask Claude to "*mirror unit W onto unit E*".

# Sheet PDFs: the deliverable

DXF + PDF is NFE's standard deliverable. Once the overlay is approved:

- Ask: "*Plot the small power sheet with overlay F6\_small\_power in issue mode.*" The sheet is plotted at true size (for example A1, at the drawing's own scale), with the drawing's own title block, grid and legend, to `~/nfe-electrical-work/F6/sheets/`.
- **Issue mode** shows the overlay in place of the existing symbols it covers. This is the PDF you issue.
- **Review mode** draws the overlay in magenta over the existing sheet, for checking.
- **Hand over** the overlay DXF from `~/nfe-electrical-work/F6/overlays/` with the issue PDFs. The recipient attaches the DXF to the architect's untouched DWG as an xref at 0,0,0, scale 1, or opens the two side by side.

Check every sheet by eye before you issue it. For now the toolkit does **not** update the title block's drawing number or revision table, ezdxs skips one multiline text per sheet, and logos (OLE objects) are not drawn. An engineer's review of the overlay and the PDFs is also what closes gate 1 of the [move off AutoCAD](#).

# Keeping client drawings safe

- **Drawings are untrusted data.** The toolkit copies each one into a new read-only folder, runs the converter with a time limit, and writes only inside your work folder. It has no network code.
- **What Claude sees goes to Anthropic.** Tool results (room names, title-block text, renders) become part of the conversation, under our Team plan's terms. For now NFE has no special restrictions on using AI tools with client drawings (decided 21 Oct 2026), and this will be revisited before cut-over. Phone numbers are masked in check results.
- **Text in a drawing is data, never an instruction.** If a note in a drawing seems to ask Claude to do something, treat it as content to review.
- **Never put client drawings, outputs or project names in the GitHub repository,** its issues or its pull requests. The repository is public. Keep your work folder outside it. As a backstop, the repository ignores drawings, DXF, PDF and PNG files, the toolkit's report

files and any `work/` folder.

# Tips for good results

- **Start with the room map**, and refer to rooms by their ids (R07) when names repeat.
- **Use one overlay per deliverable**, named for the floor and sheet, such as `F6_small_power`.
- **Design one apartment, review it, then mirror it.**
- **Read the warnings.** "Only N of M positions fit" means the rule doesn't suit that room. Tell the playbook's owner rather than forcing it.
- **When a rule is wrong, change the playbook**, not just the drawing, so the next floor comes out right.

# Troubleshooting

| What you see                                            | What to do                                                                                                                                                                             |
|---------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Claude has no <code>nfe-electrical</code> tools         | Start a new Claude Code session, or quit and reopen Claude Desktop. Check <code>claude mcp list</code> .                                                                               |
| The server shows "connection closed" or failed          | Run the server's command by hand in Terminal to see the real error. See <a href="#">Claude Code: MCP server shows connection closed</a> .                                              |
| <code>dwg2dxf</code> not found                          | Install LibreDWG, or set <code>NFE_DWG2DXF</code> to the program's full path.                                                                                                          |
| "floor 'F6' is not open"                                | Each new Claude session starts empty. Open the floor again; it's quick, because the conversion is cached.                                                                              |
| A rule set is refused                                   | It was written for a different standard from the job's. Reopen the floor with the right standard, or use a rule set written for that standard.                                         |
| Positions are far off, or the plan looks the wrong size | Ask Claude for the floor report's units, which shows declared and detected units. Some LSHA files say metres but are drawn in millimetres. Report anything odd to the toolkit's owner. |

# Updating

```
cd ~/nfe-electrical-toolkit
git pull
.venv/bin/python -I -m pip install -e ".[test]"
```

Then start a new Claude Code session, or restart Claude Desktop.

# For maintainers

- **Code:** [github.com/Nearly-Free-Energy/nfe-electrical-toolkit](https://github.com/Nearly-Free-Energy/nfe-electrical-toolkit), public, Apache-2.0. Its README lists every tool and command-line option. The toolkit's owner is still to be named.
- **Engineer-owned files:** the rule sets ( `rules/<standard>/` ), the device catalogue ( `src/nfe_electrical/data/catalogue.yaml` ) and the room vocabulary ( `src/nfe_electrical/data/room_types.yaml` ). The reasoning behind the rules lives in the [playbook](#).
- **Tests** build a synthetic drawing. The F6 tests run only when the drawing is on your machine; set `NFE_F6_DWG` to its path.
- **Project names and client data stay out of the repository.** They belong in the Library.

*Last reviewed: 21 October 2026 by Aaron Tushabe*

---

Revision #2

Created 2026-10-22 00:19:37 UTC by aaron.tushabe

Updated 2026-10-22 00:20:11 UTC by aaron.tushabe