BATEM terminal (`baterm`) — User manual

Interactive shell for BATEM (`batem.baterm`). Three session entities — **context**, **building**, **pv** — share one hourly `DataProvider`.

First run: if `data/<location>.json` is missing, weather is downloaded — do not interrupt.
Short tour: baterm_demo.md · **Checklists:** Building_Data_Required.md · **In shell:** `help baterm` · `manual`

Table of contents

Start

```
baterm
# or from a source checkout:
python -m batem.baterm
```

```
help baterm | help context | help building | help pv
help naming          # prints doc/naming_convention.md
manual               # open this file
clear                # reset session memory (files kept)
```

Tab completion, `history` / `!N`, `set timing True|False` (on by default).

Session model

Entity	Type	Role
Context	`ContextData`	Site, weather period, distant masks → weather `DataProvider`
Building	`BuildingData` → `Building`	Geometry, envelope, HVAC; needs context to simulate
PV	`PVplantData` → `PVplant`	Array (+ embedded context); `pv simulate` writes on the same `DataProvider`

```
Context  ──required──►  Building
    └──required──►  PV
```

- Context alone → climate plots / heliodons.
- Loading a `.pv` adopts its embedded context if none is loaded yet; if a session

Files

Ext	Content	Load
`.context`	`ContextData`	`context load <path>` (not bare `context <path>`)
`.building`	`BuildingData`	`building load <path>`
`.pv`	`PVplantData` (+ nested context)	`pv load <path>` or `pv <path>`
`.sidemask`	one `SideMaskData`	`sidemask load <name>`
`.baterm`	command batch	`run <path>`

Extensions are auto-appended. Save with `context|building|pv save`. Forget memory: `… delete` or `clear`.

Context

Defines **where/when**: location, lat/lon, albedo, pollution, year/dates, distant `side_masks`.

Field	Notes
`location`	Weather file `data/<location>.json`
`latitude_north_deg` / `longitude_east_deg`	Needed for download + solar geometry
`albedo` / `pollution`	Solar model
`simulated_year`, dates, `side_masks`	Optional (defaults in form)

Python: `ContextData.help("location")`.

Command	Role
`context`	Summary

Building

Needs **context + building**. Live `Building` enables 3D, heliodons, simulate, variation.

Commands

Command	Role
`building`	Compact summary
`building new` / `edit`	GUI (list fields: type **inner** items; form adds `[]`)
`building load` / `save` / `delete`	File & memory
`building footprint` `3d` `heliodons`	Views
`building simulate [--quick]`	Hourly run (`--quick` = first 7 days)
`building results`	HVAC + comfort reprint

PV

Plant geometry + module model; embeds `ContextData`. `pv simulate` registers collector powers and `PV:power_W`.

Commands

Command	Role
`pv`	Summary
`pv new` / `edit`	GUI (needs context; opens context editor if missing)
`pv load` / `save` / `delete` / `dir`	Files
`pv 3d` / `heliodon` / `best angles`	Layout & sun
`pv simulate [--quick]`	Hourly production

DataProvider, plot, export

One hourly provider for weather + building `#sim` + PV + enercost.

Command	Scope
`vars` / `plot`	All origins
`context|building|pv vars` / `plot`	Filtered
`export myrun`	`myrun.xlsx` in cwd (all series)

Plots run in a **subprocess** (closing a window does not exit baterm).

Side masks

Distant shading (`SideMaskData`): in `ContextData.side_masks` and/or the session list (merged for 3D / heliodons).

Command	Role
`sidemask`	Table
`sidemask new` / `edit <name>`	GUI
`sidemask load` / `save` / `delete`	`.sidemask` files
`sidemask 3d [name …]`	Preview

Workspace

Command	Role
`cd` `pwd` `ls`	Navigate
`dir [folder]`	List project files
`status`	Session snapshot
`clear`	Forget context / building / PV / session masks
`export` `vars` `plot`	Data hand-off
`load <path>`	Remember `current_file` only
`workshops` [`list` | `stop` | `<id>`]	JupyterLab for `workshopXX_*.ipynb` (see below)
`lectures` [`list` | `<id>`]	List / open `lectures/slidesXX_*.pdf` (see below)

Electricity cost

Needs `PELEC:building#sim` and `PV:power_W`:

```
enercost
enercost hp_hc
enercost base --base 0.25 --export 0.12
```

Writes `ENERCOST:hour#sim`, `GRID_IMPORT:kWh#sim`, `GRID_EXPORT:kWh#sim`. Default = French HP/HC (HC 22:00–06:00).

Scripts

Command	Runs	Cwd
`run file.baterm`	Batch in current session (`#` comments)	—
`run file.sh`	Shell script	`batem/` package dir
`pyrun file.py [args]`	Same Python as baterm	`batem/` package dir

Aliases: `script` / `!` → `run` · `pyscript` → `pyrun`.

`pyrun` does **not** see in-memory objects — `export` an xlsx (absolute path) or reload files in the script.

```
building_simulate
```

Typical workflows

Site only

```
context load grenoble
context temperature
context heliodon
```

Building

```
context load grenoble
building load ref_house
building 3d
building simulate
building results
```

Roof example: `run examples/building_with_roof/run_hip_roof`.

Building + PV + bill

GUIs

Forms (`new` / `edit`), 3D (PyVista), climate figures (Matplotlib), and time-series plots (Plotly) run in **child processes** so the shell stays responsive. Needs **cmd2**, **Tk**, **PyVista**, **Plotly**, **Matplotlib**, **openpyxl**, **prettytable** · Python ≥ 3.10.

See also

Document	Content
baterm_demo.md	Short slide demo
QUICK_REFERENCE.md	Cheat sheet
Building_Data_Required.md	Simulate checklist
Building_User_Manual.md	Building physics
naming_convention.md	Variable names (`help naming`)
CONTROL_SYSTEMS_GUIDE.md	HVAC control
batem/baterm/README.md	Package layout
`batem/baterm/mixins/*.py`	Command source of truth

BATEM terminal — `batem.baterm` · `baterm` or `python -m batem.baterm`.