

# Configuration Reference

Generated on August 15, 2026

## application.properties vs application.config

- `application.properties` controls the process itself: listen port, runtime-level settings, database connection details where applicable. You edit this once per install and rarely touch it again.
- `application.config` (an entity, not a static file) controls product-level behaviour: locale, product name, and similar settings that are meant to be changed from the UI rather than by hand-editing a file on disk.

## `orchester.root` and directory layout

Every install has a single root directory (`orchester.root`, defaulting to `/opt/orchester` on Linux and Docker installs) containing:

| Directory              | Contents                                                                     |
|------------------------|------------------------------------------------------------------------------|
| <code>entities/</code> | Your configuration: peers, modules, variables, schemas — one file per entity |
| <code>logs/</code>     | Application logs                                                             |
| <code>loggers/</code>  | Time-series data recorded by logger modules                                  |
| <code>instance/</code> | The engine's own identity and licence state — not user-editable              |

An upgrade or reinstall must never touch these directories directly; see [Upgrading](#).

## Entity files

Each entity (a peer, a module, a variable definition) is stored as one file under `entities/`, named by its type and id. You will not normally hand-edit these — the UI is the supported way to change configuration — but knowing the layout helps when scripting backups or diffing configuration between environments.

## Where module-specific settings live

Configuration specific to a given module type (for example, connection parameters for a particular protocol) is documented alongside that module type in the UI itself, not duplicated here.

## Next steps

- [Upgrading](#)
- [Uninstalling](#)