

Movelt Pro 8.0 Cheat Sheet



CLI & Workflow Commands	What It Does / When to Use
<code>moveit_pro --help</code>	Lists all Movelt Pro CLI verbs and global options.
<code>moveit_pro run -c <config_package> [--no-drivers --only-drivers] [--headless] [--verbose] [--no-browser]</code>	Launches the Movelt Pro runtime and UI for a given robot configuration package. Us
<code>moveit_pro configure [-c <config_name>] [-w <workspace_path>]</code>	Set up or switch robot configuration package and user workspace.
<code>moveit_pro build user_workspace [--colcon-args]</code>	Build all ROS packages in your user workspace. Useful after making changes.
<code>moveit_pro test [--colcon-args "..."]</code>	Run unit/integration tests on your workspace. Helps ensure new code doesn't break existing tests.
<code>moveit_pro rviz</code>	Launch RViz with Movelt Pro's recommended configuration; good for deeper visual debugging.
<code>moveit_pro shell</code>	Drop into a shell in the running Movelt Pro container. Use for debugging, ROS node/topic introspection.
<code>moveit_pro dev</code>	Developer mode – mounts your workspace, useful for developing behaviors, testing builds locally.

Tips & Tricks

Verbose mode: Use `moveit_pro run -v` if moves, planning failures, or driver issues aren't obvious. Helps turn up more diagnostics.

No drivers / only drivers: If you want to split responsibilities (e.g. one PC running hardware drivers, another for sim or UI), use `--no-drivers` or `--only-drivers`.

Headless mode: For CI pipelines or programmatic usage (no UI), use `--headless` with `moveit_pro run`.

Browser suppression: If launching on machines without GUI or to avoid auto-opening browser, use `--no-browser`.

Selecting example config packages: Use `-c <example_config>` when starting so you can load sample environments (e.g. `lab_sim`, `hangar_sim`) to prototype quickly.

Use `--colcon-args` to target builds/tests to specific packages to save time when working in large workspaces.

Breakpoints in Behavior Trees: Add `BreakpointSubscriber` Behavior to Objective to pause execution; resume via UI or via publishing to a ROS topic if debugging.

config.yaml: Key central file in each robot config package. Defines hardware settings, simulation vs real, Movelt params, ROS2 control, Objectives / Behaviors, global overrides. Use `based_on_package` field for inheritance to reduce repetition.

Advanced Usage & Debugging Tools

Log Access: Logs are stored in the `~/moveit_pro/logs/` directory on the host. Use this to view startup sequences, runtime errors, or Behavior Tree issues after the container exits.

Behavior Tree Replay: Use the "Replay" mode in the UI to re-run previously executed Objectives for debugging without resending inputs.

Topic Echoing from Container: Use `moveit_pro shell` and then: `ros2 topic echo /your/topic/name`

Objective Watchdog: If Objectives fail silently or hang, check `/moveit/objective_server/status` and `/moveit/objective_server/heartbeat` to verify process health.

Rebuild Only a Single Behavior:
`moveit_pro build user_workspace --colcon-args="--packages-select=my_behavior_pkg"`

Run from External Launch: You can integrate with non-Movelt Pro launch stacks by calling `moveit_pro run --headless` and then launching external nodes alongside it.

Fleet Sim Testing: Use `moveit_pro run --no-drivers` on your dev machine to simulate multiple robots in parallel, especially helpful when testing multi-agent Objectives.