6.6 Troubleshooting Tool Issues

Go To Top

Return to Prerequisites and Tooling

The handbook should help developers resolve common tooling problems without turning every setup issue into an architecture discussion.

Troubleshooting guidance should cover predictable failures such as:

1. Git cannot access the Repository.
2. The shell cannot execute a script.
3. A script has incorrect line endings or permissions.
4. Python points to the wrong interpreter.
5. The virtual environment does not exist or contains missing packages.
6. DDS environment variables are missing.
7. The IDL compiler cannot find an input file.
8. Generated types appear in the wrong location.
9. The container runtime cannot build or run an image.
10. A required port, volume, network, or runtime path conflicts with a local setting.
11. VS Code cannot find recommended extensions or workspace tasks.

Troubleshooting guidance should direct developers back to the controlled Baseline. Developers should fix the environment or update the documented Baseline through review. They should not solve prerequisite problems by adding undocumented local workarounds that only work on one machine.

When a tool issue reveals a real gap in the handbook, scripts, Repository layout, or Baseline assumptions, the team should update the relevant material. A repeated troubleshooting problem usually signals that the Baseline needs clearer instructions, stronger validation, or a simpler toolchain.
