

## 7.3 Source File Naming

[Go To Top](#)

[Return to Naming and File Conventions](#)

Source file names should indicate the file's implementation responsibility. A developer should be able to distinguish [Node](#) implementation files, shared support files, [Control Plane](#) helpers, [Configuration](#) readers, status publishers, command handlers, and test support files from their names and locations.

The team should follow the naming conventions of the selected implementation language where those conventions are strong and widely understood. For example, Java classes, C++ source files, Python modules, and JavaScript or TypeScript modules may use different idiomatic file-naming conventions. The handbook should not impose a single style across all languages when the language ecosystem already provides a clear convention.

Source files should avoid generic names such as:

- helper
- utils
- misc
- test
- main
- new
- old
- final

If the team uses a generic support file, the name should still identify its role, such as:

- nodeStatusPublisher
- nodeCommandHandler
- configurationLoader
- controlPlaneTopicNames

Source file names should not include temporary version labels such as v2, new, old, backup, or final-final. Git provides version history. File names should identify the current purpose, not editing history.

---

© 2026 Dido Solutions, Inc. and Jackrabbit Consulting, Inc.

From:  
<https://wiki.didosolutions.com/> - **Dido Solutions, Inc Wiki**

Permanent link:  
<https://wiki.didosolutions.com/fxdemo/05-part/07-naming-and-file-conventions/07-3-source-file-naming/start>

Last update: **2026/08/09 17:22**

