Fortran Specialist Group
Conference

University College London
September 2022

Fortran-Lang and it's Projects

Laurence Kedward

Fortran-Lang

Presentation contents

I. History and Overview of Fortran-Lang
II. The Fortran Standard Library (stdlib)
III. The Fortran Package Manager (fpm)
IV. Demonstrations
V. Summary

I. History and Overview of Fortran-Lang

II. The Fortran Standard Library (stdlib)

III. The Fortran Package Manager (fpm)

IV. Live Demonstration

V. Summary

Formation of Fortran-Lang

- **August 2019**—conversation started on Twitter between **Ondřej Čertík**, **Milan Curcic** and Jacob Williams about perceived shortcomings in the Fortran ecosystem

Formation of Fortran-Lang

- **August 2019**—conversation started on Twitter between Ondřej Čertík, Milan Curcic and Jacob Williams about perceived shortcomings in the Fortran ecosystem

- **October 2019**—**Ondřej** creates the `j3-fortran/fortran_proposals` repository on GitHub to solicit suggestions and feedback directly from the community
  - A place to publicly suggest and discuss proposals for the Fortran standards committee
  - 90+ contributors to discussions 8 proposals drafted and submitted

Formation of Fortran-Lang

- **August 2019**—conversation started on Twitter between Ondřej Čertík, Milan Curcic and Jacob Williams about perceived shortcomings in the Fortran ecosystem

- **October 2019**—Ondřej creates the `j3-fortran/fortran_proposals` repository on GitHub to solicit suggestions and feedback directly from the community
- **November 2019—Milan** creates the `fortran-lang/stdlib` project on Github
  - A standard library with a wide scope for string handling, filesystem access, sorting, etc.

Formation of Fortran-Lang

- **August 2019**—conversation started on Twitter between Ondřej Čertík, Milan Curcic and Jacob Williams about perceived shortcomings in the Fortran ecosystem

- **October 2019**—Ondřej creates the `j3-fortran/fortran_proposals` repository on GitHub to solicit suggestions and feedback directly from the community
- **November 2019**—Milan creates the `fortran-lang/stdlib` project on Github
- **December 2019**—Discussion on the need for a dedicated Fortran package manager

Formation of Fortran-Lang

- **August 2019**—conversation started on Twitter between Ondřej Čertík, Milan Curcic and Jacob Williams about perceived shortcomings in the Fortran ecosystem

- **October 2019**—Ondřej creates the `j3-fortran/fortran_proposals` repository on GitHub to solicit suggestions and feedback directly from the community
- **November 2019**—Milan creates the `fortran-lang/stdlib` project on Github
- **December 2019**—Discussion on the need for a dedicated Fortran package manager

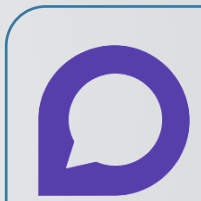
http://

- **April 2020**—Milan launches a new modern central website for Fortran
  - https://fortran-lang.org
- **May 2020**—Fortran-lang applies for a free Discourse instance:
  - https://fortran-lang.discourse.group

Fortran-Lang

Fortran Package Manager (fpm)

`stdlib` The Fortran Standard Library



Fortran-lang Discourse

Affiliated Projects

- Vscode-fortran-support
- Test-drive framework
- Homebrew-fortran
- Minpack
- Fftpack

`github.com/fortran-lang`

Lfortran compiler

Webpage

https://fortran-lang.org

I. History and Overview of Fortran-Lang

II. The Fortran Standard Library (stdlib)

III. The Fortran Package Manager (fpm)

IV. Live Demonstration

V. Summary

stdlib—Motivation

- Scope of intrinsics is limited
  - No widely-used data structures (linked-lists, dictionaries)
  - No routines for common tasks (string manipulation, error handling, high-level IO, statistical methods)
- Many common methods are coded in-house
  - Duplicate effort
  - Varying correctness and rigor
  - Prone to inefficiency or errors
- Discourages the adoption of Fortran for new projects
  - Quicker and easier to get going in other languages

The Fortran standard library aims to provide a community-developed, robust and reusable reference implementation of commonly-used procedures in a versioned and well-tested library

stdlib—Scope and Development

The Fortran standard library aims to provide a community-developed, robust and reusable reference implementation of commonly-used procedures in a versioned and well-tested library

Open source: MIT License

Github: https://github.com/fortran-lang/stdlib

API: https://stdlib.fortran-lang.org

Scope

- **Algorithms**—sorting
- **Programming**—Strings, containers, file io, testing, logging
- **Mathematics**—linear algebra, statistics, integration, root-finding, special functions

stdlib—Current status

38 Committers

100+ contributors to discussions

20+ modules:

- `array`
- `ascii`
- `bitsets`
- `error`
- `hash`
- `hasmaps`
- `io`
- `io_numpy`
- `kinds`
- `linalg`
- `logger`
- `math`
- `optval`
- `quadrature`
- `random`
- `selection`
- `sorting`
- `specialfunctions`
- `stats`
- `stats_distribution`
- `string_type`
- `stringlist`
- `strings`

stdlib—Current status

38 Committers

100+ contributors to discussions

20+ modules:

- `array`
- `ascii`
- `bitsets`
- `error`
- `hash`
- `hasmaps`
- `io`
- `io_numpy`
- `kinds`
- `linalg`
- `logger`
- `math`
- `optval`
- `quadrature`
- `random`
- `selection`
- `sorting`
- `specialfunctions`
- `stats`
- `stats_distribution`
- `string_type`
- `stringlist`
- `strings`

stdlib—Current status

38 Committers

100+ contributors to discussions

20+ modules:

- `array`
- `ascii`
- `bitsets`
- `error`
- `hash`
- `hasmaps`
- `io`
- `io_numpy`
- `kinds`
- `linalg`
- `logger`
- `math`
- `optval`
- `quadrature`
- `random`
- `selection`
- `sorting`
- `specialfunctions`
- `stats`
- `stats_distribution`
- `string_type`
- `stringlist`
- `strings`

stdlib—Compatibility

Dependencies

- Fortran compiler supporting F2008
- Cmake OR fpm
- Fypp preprocessor

Officially Supported Platforms (tested regularly in CI)

OS	Compiler
MacOS Catalina 10.15 (x86_64)	GCC Fortran 9,10,11 Intel oneAPI classic 2021.1
Ubuntu 20.04 (x86_64)	GCC Fortran 9, 10, 11 Intel oneAPI classic 2021.1
Windows Server 2019 (x86_64)	GCC Fortran (MSYS or MinGW_w64) 10

I. History and Overview of Fortran-Lang

II. The Fortran Standard Library (stdlib)

III. The Fortran Package Manager (fpm)

IV. Live Demonstration

V. Summary

Fpm—Motivation

Existing build systems for Fortran suffer from some drawbacks:

- Poor portability (e.g. makefiles)
- Complex, poor robustness
- Steep learning curve (e.g. Cmake)
- Difficult to maintain
- No easy way to incorporate existing software libraries into new projects

Fpm aims to be a Fortran-specific build system and package manager to reduce the learning curve for starting new Fortran projects and depending on other Fortran libraries.

Partly inspired by similar solutions for Rust (cargo), Python (pip) and Julia (pkg)

Fpm—Development

Fpm aims to be a Fortran-specific build system and package manager to reduce the learning curve for starting new Fortran projects and depending on other Fortran libraries.

Functionality

- Fpm can scan module/submodule dependencies and build a wide-variety of projects
- Fpm will compile Fortran, C and C++ sources
- Fpm supports incremental and parallel builds
- Fpm compatible libraries can easily be specified as project dependencies
  - Fpm will automatically download and incorporate the dependency into the local project

Open source: MIT License

Github: https://github.com/fortran-lang/fpm

Documentation: https://fpm.fortran-lang.org

27 code contributors

Implemented in **Fortran**

170+ fpm-compatible packages on Github and Gitlab

Fpm—Command-line Interface

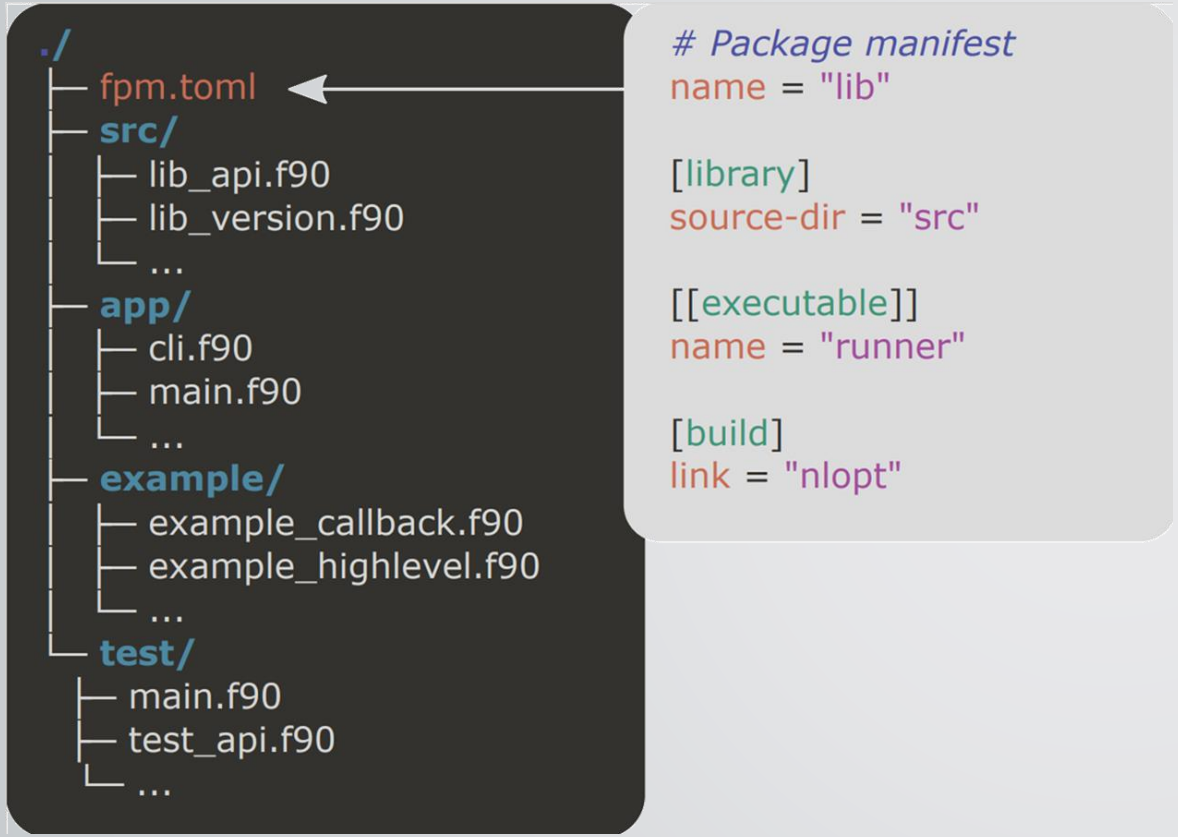
Fpm is a command-line application with a subcommand interface (like git):

```
$ fpm <subcommand>
```

Common subcommands:

- **`fpm new`** – initialize a new fpm project
- **`fpm build`** – fetch dependencies and compile libraries and executables
- **`fpm run`** – `build` + run executable(s)
- **`fpm test`** – `build` + run tests
- **`fpm install`** – `build` + install executable(s) locally
- **`fpm update`** – update project dependencies
- **`fpm clean`** – remove all generated output objects
- **`fpm help`** – view help pages on subcommands

Fpm—Project structure



- Project is described by a package manifest file: fpm.toml
- Minimal metadata required: simple and easy to use
- Default folder structure allows fpm to find source files automatically
- Can specify third-party libraries as dependencies

I. History and Overview of Fortran-Lang

II. The Fortran Standard Library (stdlib)

III. The Fortran Package Manager (fpm)

IV. Live Demonstration

V. Summary

Summary

- Fortran-Lang is an active and growing community looking to improve the Fortran ecosystem for new and existing developers
- Fortran-Lang leads development of:
  - The Fortran Standard Library (stdlib)
  - The Fortran Package Manager (fpm)
  - Other Fortran packages and tooling
- Fortran-Lang has an online presence:
  - Fortran-Lang Discourse https://fortran-lang.discourse.group/
  - Fortran-Lang website https://fortran-lang.org/en/

Read more in:

L. J. Kedward et al., "The State of Fortran"

in Computing in Science & Engineering, vol. 24, no. 2, 1 March-April 2022

doi: 10.1109/MCSE.2022.3159862.

Acknowledgments

Bálint Aradi • Ondřej Čertík • Milan Curcic • Sebastian Ehlert • Philipp Engel • Rohit Goswami • Michael Hirsch • Asdrubal Lozada-Blanco • Vincent Magnin • Arjen Markus • Emanuele Pagone • Ivan Pribec • Brad Richardson • Harris Snyder • John Urban • Jérémie Vandenplas

And all contributors to Fortran-Lang projects and discussions

Thank you for Listening!