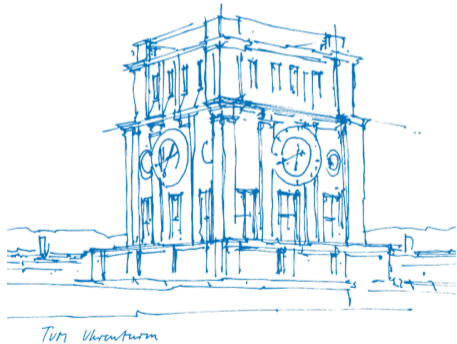


Continuous integration and documentation automation in preCICE


17th OpenFOAM Workshop
Cambridge, UK

Gerasimos Chourdakis
Technical University of Munich

July 11, 2022



A quality dashboard for your project


README.md 

preCICE



Communication

chat [on gitter](#) | discourse QA [4.1k posts](#) | mailing list [subscribe](#) | twitter [@preCICE_org](#) | website [up](#)

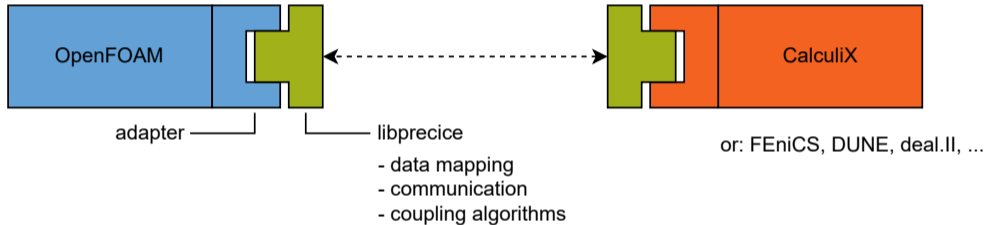
Project Status

release [v2.4.0](#) | doi [10.18419/darus-2613](#) |  Build and Test [passing](#) | system tests [check](#)

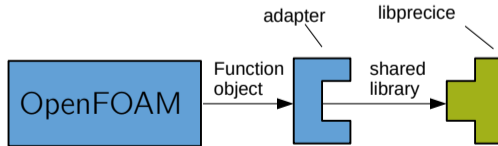
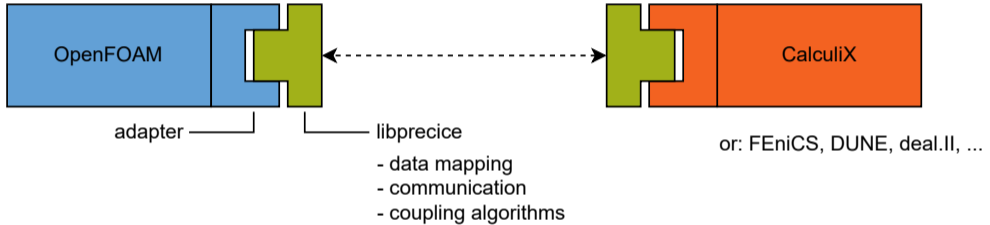
Project Quality

xSDK [member](#) | openssf best practices [passing](#) | codefactor [A](#) |  lgtm:C++ [A+](#) |  codecov [90%](#)

preCICE and its OpenFOAM adapter



preCICE and its OpenFOAM adapter



Continuous Integration tools



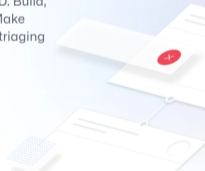
GitHub Actions

Automate your workflow from idea to production

GitHub Actions makes it easy to automate all your software workflows, now with world-class CI/CD. Build, test, and deploy your code right from GitHub. Make code reviews, branch management, and issue triaging work the way you want.

[Get started with Actions >](#)

Questions? [Contact Sales >](#)



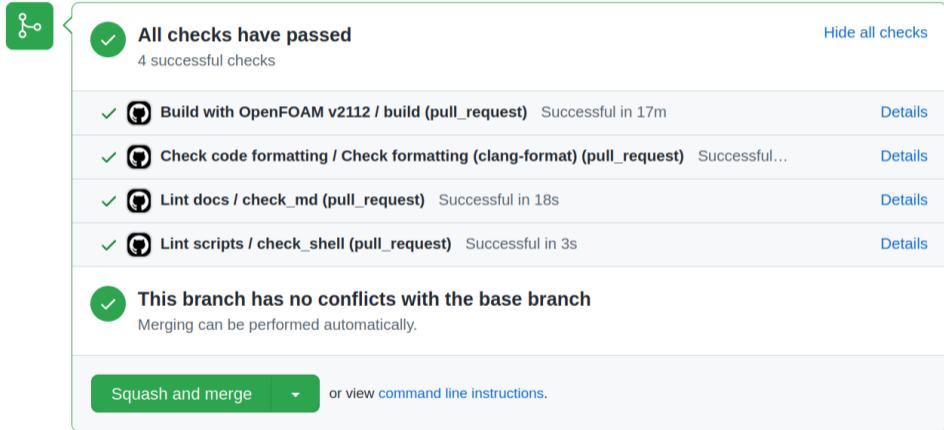
GitLab Continuous Integration (CI)

Outstanding source code exists. For teams and projects big and small, the proof is in the pipeline.


[Get to know CI/CD](#)









⋮


Safely contributing to the OpenFOAM adapter



A screenshot of a GitHub pull request status card. On the left is a green icon with a white branching diagram. The main area is a light blue rounded rectangle with a green border. At the top left is a green checkmark icon. The main heading is "All checks have passed" in bold black text, with "4 successful checks" below it. To the right is a link "Hide all checks". Below this are four rows, each with a green checkmark, a GitHub Actions icon, a job name, a status, and a "Details" link. The jobs are: "Build with OpenFOAM v2112 / build (pull_request) Successful in 17m", "Check code formatting / Check formatting (clang-format) (pull_request) Successful...", "Lint docs / check_md (pull_request) Successful in 18s", and "Lint scripts / check_shell (pull_request) Successful in 3s". At the bottom is a green checkmark icon, the heading "This branch has no conflicts with the base branch", and the text "Merging can be performed automatically.". At the very bottom is a green button "Squash and merge" with a dropdown arrow, followed by the text "or view command line instructions.".

 **All checks have passed** [Hide all checks](#)
4 successful checks

  Build with OpenFOAM v2112 / build (pull_request) Successful in 17m Details
  Check code formatting / Check formatting (clang-format) (pull_request) Successful... Details
  Lint docs / check_md (pull_request) Successful in 18s Details
  Lint scripts / check_shell (pull_request) Successful in 3s Details

 **This branch has no conflicts with the base branch**
Merging can be performed automatically.

[Squash and merge](#) or view [command line instructions](#).

GitHub Actions for the OpenFOAM adapter

Workflows

[New workflow](#)

All workflows

- Build and store a Docker ima...
- Build with OpenFOAM v2112
- Check code formatting
- Check links (manual)
- Custom build (manual)
- Lint docs
- Lint scripts
- Lint scripts and docs

All workflows

Showing runs from all workflows

X

742 workflow run results		Event ▾	Status ▾	Branch ▾	Actor ▾
	Extend Allwmake suggestions for possible problems Lint scripts #374: Pull request #220 opened by MakisH	MakisH:extend-allwmake-su...	📅 3 days ago 🕒 19s	...	
	Extend Allwmake suggestions for possible problems Lint docs #368: Pull request #220 opened by MakisH	MakisH:extend-allwmake-su...	📅 3 days ago 🕒 29s	...	
	Extend Allwmake suggestions for possible problems Check code formatting #382: Pull request #220 opened by MakisH	MakisH:extend-allwmake-su...	📅 3 days ago 🕒 46s	...	
	Extend Allwmake suggestions for possible problems Build with OpenFOAM v2112 #438: Pull request #220 opened by MakisH	MakisH:extend-allwmake-su...	📅 3 days ago 🕒 14m 6s	...	
	Update master (v1.1.0) Lint scripts #354: Pull request #186 synchronize by MakisH	develop	📅 12 days ago 🕒 21s	...	

Automatic format check with clang-format

13 lines (13 sloc) | 369 Bytes

```
1 name: Check code formatting
2 on: [push, pull_request]
3 jobs:
4   formatting-check:
5     name: Check formatting (clang-format)
6     runs-on: ubuntu-latest
7     steps:
8     - uses: actions/checkout@v2
9     - name: Run clang-format style check for C/C++ programs.
10       uses: jidicula/clang-format-action@main
11       with:
12         clang-format-version: '11'
13         check-path: '.'
```


Automatic format check with clang-format

```
13 lines (13 sloc) | 369 Bytes ...  
1 name: Check code formatting  
2 on: [push, pull_request]  
3 jobs:  
4   formatting-check:  
5     name: Check formatting (clang-format)  
6     runs-on: ubuntu-latest  
7     steps:  
8     - uses: actions/checkout@v2  
9     - name: Run clang-format style check for C/C++ programs.  
10       uses: jidicula/clang-format-action@main  
11       with:  
12         clang-format-version: '11'  
13         check-path: '.'
```

By the way: we need an official style file

Development > openfoam > Issues > #1634

Open

Created 2 years ago by  Gerasimos Chourdakis

Close issue



Style formatting config needed (e.g. clang-format)

Functionality to add/problem to solve

Add a [clang-format](#) or similar system config file in the code repository, so that contributions to the repository or derived projects can follow the [Coding Style Guide](#).

Clang-format is an established formatting tool and integration is already provided in many editors and IDEs, including [Emacs](#), [Vim](#), and [CLion](#).

Target audience

- Regular developers of OpenFOAM
- External contributors to OpenFOAM
- Contributors to third-party OpenFOAM projects (e.g. function objects, such as the [preCICE OpenFOAM adapter](#))

Proposal

1. Create a `.clang-format` file, e.g. in the root directory of the repository, or any other directory preferred for developers' tools.
 - For example, with `clang-format -style=llvm -dump-config > .clang-format`
2. Define the preferred rules following the [Clang-Format Style Options](#). Beware that the rules may depend on the Clang-Format version.

Discover issues in shell scripts with shellcheck

```

-$ shellcheck /usr/lib/openfoam/openfoam2206/bin/tools/RunFunctions

In /usr/lib/openfoam/openfoam2206/bin/tools/RunFunctions line 1:
#-----* sh -*-----
^-- SC2148 (error): Tips depend on target shell and yours is unknown. Add a shebang or a 'shell' directive.

In /usr/lib/openfoam/openfoam2206/bin/tools/RunFunctions line 119:
    value="$(foamDictionary -value $@ 2>/dev/null)" || return 2
                ^-- SC2068 (error): Double quote array expansions to avoid re-splitting elements.

In /usr/lib/openfoam/openfoam2206/bin/tools/RunFunctions line 160:
    if [ -n "nFaces" ]
        ^----^ SC2157 (error): Argument to -n is always true due to literal strings.

In /usr/lib/openfoam/openfoam2206/bin/tools/RunFunctions line 184:
    set -- $(foamDictionary -entry numberOfSubdomains -value "$dict" 2>/dev/null)
        ^-- SC2046 (warning): Quote this to prevent word splitting.

    ...

In /usr/lib/openfoam/openfoam2206/bin/tools/RunFunctions line 484:
    for file in $(grep -l "#include" * 2>/dev/null)
                ^-- SC2013 (info): To read lines rather than words, pipe/redirect to a 'while read' loop.
                    ^-- SC2035 (info): Use ./glob* or -- glob* so names with dashes won't become options.

```

For more information:

- <https://www.shellcheck.net/wiki/SC2068> -- Double quote array expansions to ...
- <https://www.shellcheck.net/wiki/SC2145> -- Argument mixes string and array. ...
- <https://www.shellcheck.net/wiki/SC2148> -- Tips depend on target shell and y...

Run shellcheck on GitHub Actions

10 lines (10 sloc) | 265 Bytes

```
1 name: Lint scripts
2 on: [push, pull_request]
3 jobs:
4   check_shell:
5     runs-on: ubuntu-latest
6     steps:
7       - name: Check out repository
8         uses: actions/checkout@v2
9       - name: Lint shell scripts (shellcheck)
10        uses: ludeeus/action-shellcheck@master
```

More Actions: check URLs

14 lines (14 sloc)

445 Bytes

...

```
1 name: Check links (manual)
2 on: workflow_dispatch
3 jobs:
4   check_links:
5     runs-on: ubuntu-latest
6     steps:
7       - name: Check out repository
8         uses: actions/checkout@v2
9       - name: Check links in markdown files (markdown-link-check)
10        uses: gaurav-nelson/github-action-markdown-link-check@v1
11     with:
12       use-quiet-mode: 'yes'
13       use-verbose-mode: 'no'
14       config-file: '.markdown-link-check-config.json'
```

GitHub Actions: Manual triggering

Run workflow

Use workflow from

Branch: develop

Virtual Environment *

ubuntu-18.04

Ref (branch/tag/commit) of the OpenFOAM adapter to build *

develop

Version of OpenFOAM to build with *

OpenFOAMv2112

Version of preCICE to build with *

2.3.0

Run tutorial flow-over-heated-plate

Run tutorial quickstart

Run tutorial partitioned-pipe

Branch of the tutorials to use *

master

Run workflow

GitHub Actions: Manual triggering (config)

```
# File .github/workflows/build-custom.yml
name: Custom build (manual)
on:
  workflow_dispatch:
    inputs:
      # ...
    versionOpenFOAM:
      type: choice
      options:
        - OpenFOAMv2206
        - OpenFOAM9
  jobs:
    build:
      runs-on: ubuntu:latest
      steps:
        - name: Run something
          run: ./run.sh
```

Workflow logs

[precice / tutorials](#) Public
Unpin
Unwatch 8
Fork 55
Starred 51

[Code](#)
[Issues 22](#)
[Pull requests 3](#)
[Discussions](#)
[Actions](#)
[Projects](#)
[Security](#)
[Insights](#)
[Settings](#)

✓ Add a cleanup step Run tutorials with docker-compose #19
 Re-run all jobs
...

[Summary](#)

Jobs

- ✓ run-flow-over-heated-plate

run-flow-over-heated-plate Search logs ⚙️

succeeded 12 days ago in 58s

- > ✓ Set up job 1s
- > ✓ Checkout repository 2s
- > ✓ Run docker-compose 49s
- > ✓ Archive logs 1s
- > ✓ Archive case files 5s
- > ✓ Post Checkout repository 0s
- > ✓ Complete job 0s

Workflow artifacts

precice / tutorials Public

Unpin Unwatch 8 Fork 55 Starred 51

<> Code Issues 22 Pull requests 3 Discussions Actions Projects Security Insights Settings

✓ Add a cleanup step Run tutorials with docker-compose #19 Re-run all jobs ⋮

Summary

Jobs

- run-flow-over-heated-plate

Triggered via push 12 days ago

MakisH pushed → 8a77be3 [add-nightly-tests](#)

Status	Total duration	Artifacts
Success	1m 10s	2

run-tutorials-compose.yml
on: push

- run-flow-over-heated-plate 51s

Artifacts
Produced during runtime

Name	Size
case-files	6.63 MB
logs	155 KB

Documentation automation

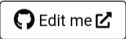
adapter repository: docs/config.md

```

389 lines (292 sloc) | 15.2 KB
1 ---
2 title: Configure the OpenFOAM adapter
3 permalink: adapter-openfoam-config.html
4 keywords: adapter, openfoam, configuration, preciceDict, controlDict
5 summary: "Write a system/preciceDict, set compatible boundary conditions, and activate the adapter in your system/controlDict"
6 ---
7
8 In order to run a coupled simulation, you need to:
9
10 1. prepare a preCICE configuration file (described in the [preCICE configuration](https://www.precice.org/configuration))
11 2. prepare an adapter's configuration file,
12 3. set the coupling boundaries in the OpenFOAM case,
13 4. load the adapter, and
14 5. start all the solvers normally, from the same directory, e.g. in two different terminals.
15
16 If you prefer, you may find an already prepared case in our [tutorial for CHT: Flow over a heated plate](https://precice.org/tutorial-3-1-cht-flow-over-a-heated-plate)
17
18 You may skip the section "_Advanced configuration_" in the beginning, as it only concerns special cases. You may also
19
20 ## The adapter's configuration file
21 ~
  
```

website rendering:

The screenshot shows the preCICE website interface. At the top, there is a navigation bar with links for 'Quickstart', 'Docs', 'Tutorials', 'Community', 'Blog', and 'About'. A search bar is also present. The main content area is titled 'Configure the OpenFOAM adapter' and includes a summary: 'Write a system/preciceDict, set compatible boundary conditions, and activate the adapter in your system/controlDict.' Below the summary is a 'Table of Contents' section with a list of links: 'The adapter's configuration file', 'Configuration of the OpenFOAM case' (with sub-links for 'Boundary conditions', 'CHT', 'FI', and 'Load the adapter'), 'Advanced configuration' (with sub-links for 'Nearest-projection mapping', 'Adapter implementation', 'Additional properties for some solvers', 'Coupled heat transfer', and 'Fluid-structure interaction'), 'Additional parameters in the adapter's configuration file' (with sub-links for 'User-defined solver type', 'Parameters and fields with different names', and 'Debugging'), and 'Coupling OpenFOAM with 2D solvers'. An 'Edit me' button is located in the top right corner of the content area.



Documentation automation: setup

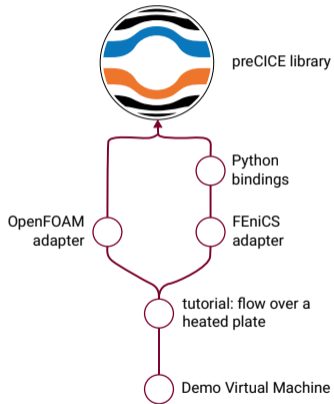
website repository: Git modules

```
0 lines (0 sloc) | 255 Bytes ...
1 [submodule "tutorials"]
2   path = imported/tutorials
3   url = https://github.com/precice/tutorials.git
4   branch = develop
5 [submodule "openfoam-adapter"]
6   path = imported/openfoam-adapter
7   url = https://github.com/precice/openfoam-adapter.git
8   branch = develop
```

adapter repository: Trigger update

```
22 lines (22 sloc) | 598 Bytes Raw Blame Edit ...
1 name: Update website
2 on:
3   push:
4     branches:
5       - 'develop'
6     paths:
7       - 'docs/**'
8   jobs:
9     trigger:
10    runs-on: ubuntu-latest
11    env:
12      WORKFLOW_FILENAME: update-submodules.yml
13    steps:
14      - name: Trigger workflow
15        run: |
16          curl \
17            --request POST \
18            --url https://api.github.com/repos/precice/precice.github.io/actions/workflows/$WORKFLOW_FILENAME/dispatches \
19            --header "authorization: token ${ secrets.WORKFLOW_DISPATCH_TOKEN }" \
20            --header "Accept: application/vnd.github.v3+json" \
21            --data '{"ref":"master"}' \
22            --fail
```

The multi-component preCICE ecosystem



preCICE	Section 5.1	Section 5.4	Section 5.1	Section 5.1	Section 5.1	Section 5.4
	0110010 warning 01error	<ul style="list-style-type: none"> formatting clang-format tests coverage lcov, CodeCov code analysis codacy, codefactor coverity-scan, lgtm 				
	GCC, Clang		Boost.test	Boost.test + MPI	solver dummies	CPack, lintian (+ Spack)
	building	quality assurance	unit tests	integration tests	smoke tests	packaging

Python bindings	Section 5.2	Section 5.2	Section 5.2	Section 5.2	Section 5.2	Section 5.2
						<ul style="list-style-type: none"> formatting PEP8, markdownlint publishing Docker, twine, PyPI
	unittest	fake preCICE lib	unittest	MagicMock	shell	more
	unit tests	integration tests	unit tests	integration tests	system tests	

MATLAB bindings (under construction)	OpenFOAM adapter build, format, lint scripts	deal.II adapter build	CalculiX adapter (under construction)	SU2 adapter (under construction)	code_aster adapter (under construction)
---	---	--------------------------	--	-------------------------------------	--

Tutorials	Section 5.3					Section 5.4
						<ul style="list-style-type: none"> linting shellcheck formatting PEP8, markdownlint links check markdown-link-check
	flow over heated plate	perpendicular flap	Turek-Hron FSI3	partitioned heat eq.	partitioned pipe	more
	system / regression tests					

vm	ch-img	doxy	.org
Section 5.4 lint scripts, build, package	Section 5.4 Docker images for ubuntu, centos, arch	Section 5.4 publish Doxygen docs	Section 5.4 format, build, deploy

Key reference (fresh!)



Search

Research and Innovation

Open Research Europe

SUBMIT YOUR RESEARCH

Browse
Gateways & Collections
How to Publish ▾
About ▾
Blog
Sign in

26 Views
12 Downloads
1 Citations

“ Cite
⬇ Download ▾
➔ Export ▾
↔ Share ▾
📊 Track

[Home](#) > [Articles](#) > [preCICE v2: A sustainable and user-friendly coupling library](#)

SOFTWARE TOOL ARTICLE 

preCICE v2: A sustainable and user-friendly coupling library [version 1; peer review: 2 approved]

Gerasimos Chourdakis , Kyle Davis , Benjamin Rodenberg , Miriam Schulte , Frédéric Simonis , Benjamin Uekermann   , Georg Abrams, Hans-Joachim Bungartz, Lucia Cheung Yau, Ishaan Desai , Konrad Eder, Richard Hertrich, Florian Lindner , Alexander Rusch , Dmytro Sashko, David Schneider , Amin Totounferoush , Dominik Volland, Peter Vollmer , Oguz Ziya Koseomur

This article is included in Excellent Science gateway


Open Peer Review

Approval Status ✔✔ 

	1	2
Version 1		
29 Apr 22	✔ view	✔ view

1. **Axelle Viré**, Delft University of Technology, Delft, The Netherlands
2. **Garth Wells** , University of Cambridge, Cambridge, UK

Comments on this article

Summary

- CI: execute workflows automatically
- GitHub Actions: easy and powerful
- Workflow: a YAML file under `.github/`
- preCICE: Many workflows in different repositories

Read more in the preCICE v2 paper (new)

Slides & feedback: go.tum.de/613308



`gerasimos.chourdakis@tum.de` (@MakisH)

(Note: looking for a research stay abroad in 2023)

Extras

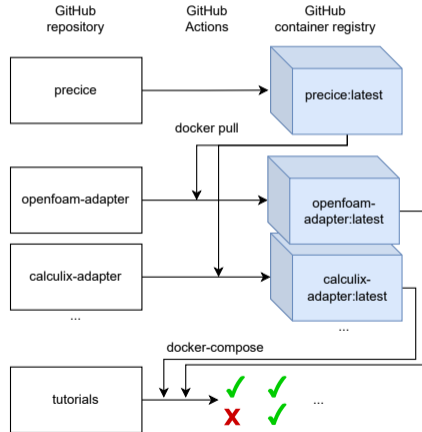
Prototype: System tests for the complete ecosystem

Testing the complete system for regressions:

- Using multiple layers together
- Running complete examples
- Comparing results


(regression) testing \neq validation!

System tests architecture (prototype)



GitHub Container Registry

openfoam-adapter

 Install from the command line:

[Learn more](#)

```
$ docker pull ghcr.io/precice/openfoam-adapter:latest
```

Recent tagged image versions

latest

↓ 8

Published 12 days ago · Digest [...](#)

[View and manage all versions](#)

Previous approach to system tests

Travis CI [Dashboard](#) [Changelog](#) [Documentation](#) [Help](#)

We are unable to start your build at this time. You exceeded the number of users allowed for your plan. Please review your plan details and follow the steps to resolution.

Search all repositories

My Repositories **Running (0/0)** +

- preCICE/systemtests** # 3411
 - Duration: 1 hr 14 min 1 sec
 - Finished: 8 months ago
- preCICE/python-bindings** # 824
 - Duration: 1 min 37 sec
 - Finished: 10 months ago
- preCICE/mpi-adapter** # 231
 - Duration: 12 min 56 sec
 - Finished: 10 months ago
- preCICE/calculix-adapter** # 105
 - Duration: 4 min 53 sec
 - Finished: 10 months ago
- preCICE/ferics-adapter** # 840
 - Duration: 6 min 1 sec
 - Finished: 10 months ago
- preCICE/openfoam-adapter** # 266
 - Duration: 13 min 51 sec
 - Finished: 10 months ago

preCICE / systemtests **Build**

Current Branches Build History Pull Requests More options

develop **CRON** Generate results with preCICE 2.2.0 and PETSc RBF Mapping. **#3411 failed** Restart build

→ Commit 8eabc93
 ↳ Branch devvs1ep
 BenjaminRodenberg authored and committed

Build jobs View config

Building preCICE 26 min 54 sec

X # 3411.1	AMD64	Bionic	Arch Linux	48 sec
✓ # 3411.2	AMD64	Bionic	Ubuntu 18.04 home [PETSc from source]	26 min 53 sec
✓ # 3411.3	AMD64	Bionic	Ubuntu 18.04.package	25 min 21 sec
✓ # 3411.4	AMD64	Bionic	Ubuntu 20.04.package [PETSc from APT]	20 min 59 sec

Building adapters

# 3411.5	AMD64	Bionic	[18.04] SU2 adapter	-
# 3411.6	AMD64	Bionic	[20.04] SU2 adapter [PETSc]	-

Manual testing to the rescue

Merged **Release v2.3.0** #1095
 fsmionis merged 313 commits into `master` from `release-v2.3.0` on Oct 6, 2021

Regression Tests

Assign each point below to a responsible person, before you continue. Use `@member`.

Run all these tests manually on your system. If you succeed, please write a comment with the revisions of the components that you used below. Example: `#507 (comment)` and update the table.

State	Success	Failure	Skipped
Write	<code>:o:</code>	<code>:x:</code>	<code>:fast_forward:</code>
Read	○	✖	⏸

State	Tester	Test
○	@IshaanDesai	perpendicular-flap fluid-nutils - solid-calculix
○	@DavidSCN	perpendicular-flap fluid-openfoam - solid-dealii
○	@IshaanDesai	perpendicular-flap fluid-su2 - solid-fenics
○	@DavidSCN	multiple-perpendicular-flaps fluid-openfoam - solid-(left+right)-dealii
○	@MakisH	flow-over-heated-plate fluid-openfoam - solid-openfoam serial + parallel
○	@IshaanDesai	flow-over-heated-plate fluid-openfoam - solid-fenics serial + parallel
○	@IshaanDesai	flow-over-heated-plate fluid-openfoam - solid-nutils
○	@MakisH	flow-over-heated-plate-nearest-projection fluid-openfoam - solid-openfoam
○	@IshaanDesai	flow-over-heated-plate-steady-state fluid-openfoam - solid-codeaster
○	@MakisH	heat-exchanger fluid-(inner+outer)-openfoam - solid-calculix

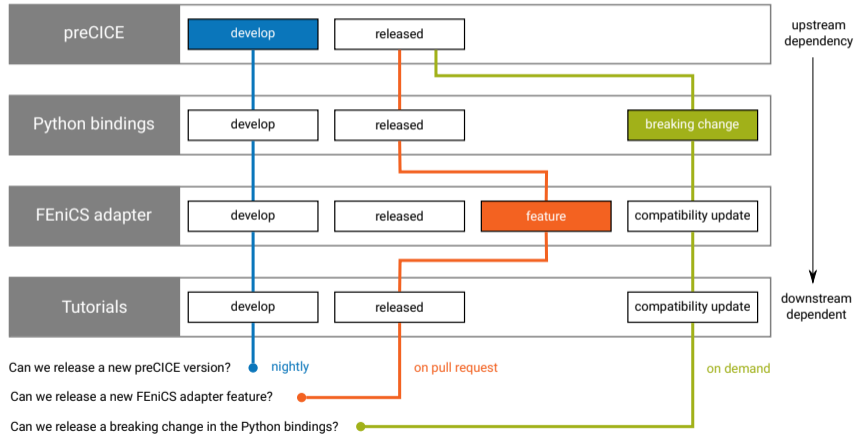
Docker Compose: Running

```
cd tutorials/flow-over-heated-plate/tests
```

```
export TAG_OPENFOAM_ADAPTER=latest
```

```
MY_UID="$(id -u)" MY_GID="$(id -g)" docker-compose up
```

Different perspectives



Docker Compose

```
# File: docker-compose.yml
services:
  # ...
  fluid-openfoam:
    image: "ghcr.io/precice/of-adapter:
           ${TAG_OPENFOAM_ADAPTER}"
    user: "${MY_UID}:${MY_GID}"
    volumes:
      - /etc/passwd:/etc/passwd:ro
      - /etc/group:/etc/group:ro
      - ../../:/tests
    command: >
             /bin/bash -c "openfoam2112 ./run.sh"
```


Docker Compose

```
# File: docker-compose.yml
services:
  # ...
  fluid-openfoam:
    image: "ghcr.io/precice/of-adapter:
           ${TAG_OPENFOAM_ADAPTER}"
    user: ${MY_UID}:${MY_GID}
    volumes:
      - /etc/passwd:/etc/passwd:ro
      - /etc/group:/etc/group:ro
      - ../../:/tests
    command: >
             /bin/bash -c "openfoam2112 ./run.sh"

  solid-openfoam:
    image: # same as fluid-openfoam
    user: # same
    volumes:
      - # same
    command: # same, different directory
```

What to compare to?

Reference data:

- Solver logs → often not identical
- Solver results → various formats, too much
- preCICE exports → same format, no time-related noise, enough

(demonstrated in TUM FSI Seminar paper 2020 by Mohamad Kanj)

What to compare to?

Reference data:

- Solver logs → often not identical
- Solver results → various formats, too much
- preCICE exports → same format, no time-related noise, enough

(demonstrated in TUM FSI Seminar paper 2020 by Mohamad Kanj)

Tutorials structure extension:

- flow-over-heated-plate/
 - fluid-openfoam/
 - solid-openfoam/
 - precice-config.xml
 - reference-data/
 - fluid-openfoam_solid-openfoam/
 - tests/
 - docker-compose.yml
- tools/
 - run-tests.sh

BSSW.io article: Unit and integration tests in preCICE



The screenshot shows the top navigation bar of the BSSW.io website, which is dark blue with white text. The logo 'better scientific software' is on the left, and navigation links for 'Resources', 'Blog', 'Events', and 'About' are on the right. Below the navigation bar is a breadcrumb trail: 'HOME > BLOG > Overcoming Complexity in Testing Multiphysics Coupling...'. The main heading of the article is 'Overcoming Complexity in Testing Multiphysics Coupling Software'. Below the heading are social media share icons for LinkedIn, Facebook, Twitter, and a general share icon. The introductory paragraph states: 'Testing complex software can easily get out of hand, especially when your product is a multiphysics coupling library. Fortunately, we've found some strategies that have helped tame the nightmare.' At the bottom, there are metadata sections: 'PUBLISHED FEB 07, 2022', 'AUTHORS FRÉDÉRIC SIMONIS, GERASIMOS CHOURDAKIS, AND BENJAMIN UEKERMANN', and 'TOPICS BETTER PLANNING BETTER RELIABILITY BETTER DEVELOPMENT', each with external links to related content.

better scientific software

Resources ▾ Blog Events About ▾ 🔍

HOME > BLOG > Overcoming Complexity in Testing Multiphysics Coupling...

Overcoming Complexity in Testing Multiphysics Coupling Software

SHARE in f t ↻

Testing complex software can easily get out of hand, especially when your product is a multiphysics coupling library. Fortunately, we've found some strategies that have helped tame the nightmare.

PUBLISHED FEB 07, 2022

AUTHORS **FRÉDÉRIC SIMONIS, GERASIMOS CHOURDAKIS, AND BENJAMIN UEKERMANN**

TOPICS **BETTER PLANNING** ↪ SOFTWARE INTEROPERABILITY
BETTER RELIABILITY ↪ TESTING
BETTER DEVELOPMENT ↪ DEVELOPMENT TOOLS