

Code management for a reproducible science with Git, GitFlow, GitHub and Zenodo

Nicolas Barrier¹
nicolas.barrier@ird.fr

Criscely Luján^{1,2}
criscely.lujan@ird.fr

¹IRD, UMR-MARBEC

²Université Paris-Sud

2021-06-23



Introduction

To reproduce an ecosystem modelling study, you will need:

- The model parameters (time-step, mortality rates)
- The forcing files (temperature, oxygen, plankton, etc.)
- **The numerical model** (source or executable)
- **The analysis scripts** (R, Python, Matlab)
- The external libraries used (graphical libraries, mathematical ones, etc.)

Therefore, a proper management of codes (model + scripts) is necessary.

This can be achieved by version control

Version control

- Tracks changes (*commits*) over time
- Saves snapshots of a project (*tags*)
- Creates derivatives of a project (*branches*)
- Facilitates collaboration among multiple users



Version control softwares

V · T · E		Version control software		[hide]
Years, where available, indicate the date of first stable release. Systems with names <i>in italics</i> are no longer maintained or have planned end-of-life dates.				
Local only	Free/open-source	RCS (1982) · SCCS (1972)		
	Proprietary	PVCS (1985) · QVCS (1991)		
Client–server	Free/open-source	CVS (1986, 1990 in C) · CVSNT (1998) · QVCS Enterprise (1998) · Subversion (2000)		
	Proprietary	AccuRev SCM (2002) · ClearCase (1992) · CMVC (1994) · Dimensions CM (1980s) · DSEE (1984) · Endeavor (1980s) · Integrity (2001) · Panvalet (1970s) · Perforce Helix (1995) · SCLM (1980s?) · Software Change Manager (1970s) · StarTeam (1995) · Surround SCM (2002) · Synergy (1990) · Team Concert (2008) · Team Foundation Server (2005) · Visual Studio Team Services (2014) · Vault (2003) · Visual SourceSafe (1994)		
Distributed	Free/open-source	ArX (2003) · BitKeeper (2000) · Codeville (2005) · Darcs (2002) · DCVS (2002) · Fossil (2007) · Git (2005) · GNU arch (2001) · GNU Bazaar (2005) · Mercurial (2005) · Monotone (2003) · Pijul (2015) · SVK (2003) · Veracity (2010)		
	Proprietary	TeamWare (1990s?) · Code Co-op (1997) · Plastic SCM (2006) · Team Foundation Server (2013) · Visual Studio Team Services (2014)		
Concepts	Baseline · Branch · Changeset · Commit · Data comparison · Delta compression · Fork (Gated commit) · Interleaved deltas · Merge · Repository · Tag · Trunk			
Category · Comparison · List				



Source: https://en.wikipedia.org/wiki/List_of_version-control_software



- **Popular and successful**

- Active development + multiple extensions

- **Tracks any type of files**

- Works best with ASCII files (.html, .tex, .R, .c, .f90)
- Large binary files with [Git-LFS](#) (.nc, .Rdata, .mat, .csv)

- **Branching**

- Light branch management
- Smarter merges
- Easy code management with [Git-Flow](#)

Several web-based hosting services are available:

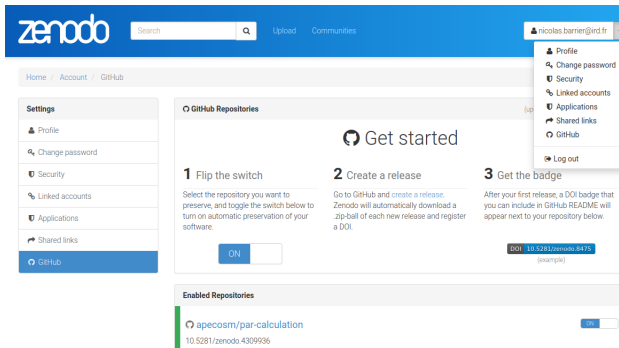
- **GitHub**, Gitlab
- Institutional: [GitLab Bioinfo](#), [GitLab UM](#), [SourceSup](#), [Ifremer Forge](#)

Why use  ?

- Popular (largest host of source code in the world)
- Facilitates worldwide collaboration
- Free (and improved accounts for education)
- Works with [Git-LFS](#)
- Continuous integration ([GitHub Actions](#))
- Association with **Zenodo**

Zenodo = *a catch-all repository for EC funded research*, based on FAIR principles (**F**indable, **A**ccessible, **I**nteroperable, **R**eusable).

Automatic generation of DOIs when a new release is drafted.



The screenshot displays the Zenodo user interface for a user named 'nicolas.barrier@ird.fr'. The main content area is titled 'Get started' and provides a three-step guide for integrating GitHub:

- 1 Flip the switch**: Select the repository you want to preserve, and toggle the switch below to turn on automatic preservation of your software. A toggle switch is shown in the 'ON' position.
- 2 Create a release**: Go to GitHub and [create a release](#). Zenodo will automatically download a .zip-ball of each new release and register a DOI.
- 3 Get the badge**: After your first release, a DOI badge that you can include in GitHub README will appear next to your repository below. An example badge is shown: [DOI 10.5281/zenodo.5475](#) (example).

On the left, a 'Settings' sidebar lists options: Profile, Change password, Security, Linked accounts, Applications, Shared links, and GitHub (which is currently selected). Below the 'Get started' guide, an 'Enabled Repositories' section shows the repository 'apcosm/par-calculation' with its DOI '10.5281/zenodo.4309936' and a toggle switch in the 'ON' position.

Good practices

Main branch (`main/master`)

- Not updated too often
- Each change is identified (`tag` and `release`)

Development branch (`develop`).

- Independent developments in separated branches

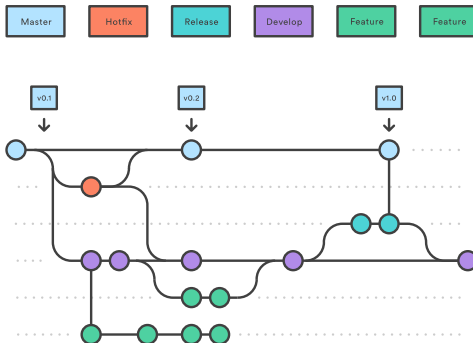
Bug fix:

- In `master` with a new version number
- In `develop`

Seems complicated? Not at all, this is GitFlow!

Git extension to manipulate these branches.

```
git flow TYPE start NAME
git flow TYPE publish NAME
git flow TYPE finish NAME
TYPE=branch type (feature, release, hotfix)
NAME=branch name (feature name, version number)
```



Example: Ichthyop

For an example, visit <https://github.com/ichthyop/ichthyop>



ICHTHYOP

license [GPL-3.0](#) DOI [10.5281/zenodo.4707436](#) release [v3.3.8](#) [java build](#) [passing](#) [issues](#) [4 open](#)

Publication date:

April 21, 2021

DOI:

[DOI: 10.5281/zenodo.4707436](#)

Related identifiers:

Supplement to

<https://github.com/ichthyop/ichthyop/tree/3.3.8>

License (for files):

[G](#) Other (Open)

Versions

Version 3.3.8	Apr 21, 2021
10.5281/zenodo.4707436	
Version 3.3.7	Apr 19, 2021
10.5281/zenodo.4706683	
Version 3.3.6	Nov 4, 2020
10.5281/zenodo.4244484	
Version 3.3.5	Nov 4, 2020
10.5281/zenodo.4243817	
Version 3.3.4	Nov 4, 2020
10.5281/zenodo.4243814	

Cite all versions? You can cite all versions by using the DOI [10.5281/zenodo.4243813](#). This DOI represents all versions, and will always resolve to the latest one. [Read more.](#)

```
🔗 develop Adding formatter
correct launch.json
Remove building of exe file
Merge branch 'feature/correct-multiple-opening' into develop
Update release notes
Closing file after getData method
Merge branch 'develop' of github.com:ichthyop/ichthyop-private into develop
Merge branch 'feature/template-com' into develop
Update release notes
Update symphone templates
Update config for 3d roms
Update roms config + add time unit
Remove test on file existence (fails for openlap)
Adding units attributes
Correct template for mercator to make it work
Adding dataset for mercator test (based on HEMO)
remove possible "C" string in time units
Update README.md
Update URL for DOI
Adding automatic compilation script for debug
Update settings.json
Merge branch 'feature/new-test-resource' into develop
Update release notes
Replace name for properties
Change path for resources
Remove classpath
add resource encoding
add arguments to debug
Move resources to the proper folder
Merge tag '3.3.8' into develop
🔗 master 3.3.8 Merge branch 'hotfix/3.3.8'
Correct bug for multiple outputs
Merge tag '3.3.7' into develop
🔗 3.3.7 Merge branch 'release/3.3.7'
Correction of shoote action (see localtime)
```