

## Using git

For information on how to use git see [https://code.mpimet.mpg.de/projects/icon/wiki/Notes\\_on\\_Git](https://code.mpimet.mpg.de/projects/icon/wiki/Notes_on_Git)

Further down on this page you'll find a more general introduction

[https://code.mpimet.mpg.de/projects/icon/wiki/Notes\\_on\\_Git#Information-on-general-Git-and-specific-ICON-Git-usage](https://code.mpimet.mpg.de/projects/icon/wiki/Notes_on_Git#Information-on-general-Git-and-specific-ICON-Git-usage)

Some reference to information on git, e.g. <https://git-scm.com/>

## Useful git-tools

### gitk

### tig:

Path on mistral: /sw/rhel6-x64/vcs/tig-2.2.1/bin/tig

Path on workstations: /usr/bin/tig

short blog entry: <https://www.atlassian.com/blog/git/git-tig>

cheat sheet: <http://ricostacruz.com/cheatsheets/tig.html>

manual: <http://jonas.nitro.dk/tig/manual.html>

(Thanks to Ralf Mueller)

## Useful git commands

Cloning the repository, including initialization of submodules:

```
git clone --recursive git@gitlab.dkrz.de:icon/icon-mpim.git
```

Check your status

```
git status
```

Check the latest commits

```
git log
```

List all available branches in your repository

```
git branch -a
```

Checkout an existing branch

```
git checkout <branch-name>
```

Checkout a new branch (take care for the naming convention of branches)

```
git checkout -b <branch-name>
```

Update your (local) repository

```
git fetch --all
```

Commit your changes - check your status and add your changes

```
git add my-file  
git add my-modified-file
```

- commit your changes including your commit message

```
git commit -m "[icon-mpim:mybranch] merge my latest changes into mybranch"
```

- commit your changes, followed by opening an editor for your commit message

```
git commit
```

- publish your changes

```
git push
```

Checkout a version with a special hash tag:

```
git checkout 89aa38fc
```

You may have to “git submodule update” again. Looking at the status the first line of the message will be

```
HEAD detached at 89aa38f
```

This, of course, only works for hash tag inside your cloned repository, e.g. icon-mpim.git.

Checkout a new branch and branch off from a special hash tag (take care for the naming convention of branches):

```
git checkout -b my_new_branch 89aa38fc
```

Delete a (local) branch

```
git branch -d my_new_branch
```

Cherry-picking a range of commits

```
git cherry-pick 3020e317..9b9d68c8 origin/icon-mpim-other-branch
```

1. - *Monika Esch* 2022/11/09 10:05

From:

<https://wiki.mpimet.mpg.de/> - **MPI Wiki**

Permanent link:

[https://wiki.mpimet.mpg.de/doku.php?id=models:icon:model\\_codes:git\\_commands](https://wiki.mpimet.mpg.de/doku.php?id=models:icon:model_codes:git_commands)

Last update: **2022/11/09 10:05**

