

ICON for NWP Developers

Marek Jacob

2021-05-26

ICON is the ICOSahedral Nonhydrostatic global forecast model suite, which was initiated by the German Weather Service (DWD) and the Max-Planck-Institute for Meteorology (MPI-M) Hamburg and is now developed jointly at DWD, MPI-M, German Climate Computing Center (DKRZ), Karlsruhe Institute of Technology (KIT) and the Center for Climate Systems Modeling (C2SM) at the ETH Zürich. Other partners also contribute to the development such as COSMO (Consortium for Small-Scale Modeling), CLM (Climate Limited-Area Modeling Community) and the Swiss National Supercomputing Centre (CSCS). This document summarizes the first steps to join the common git development work flow.

1 GitLab registration and first steps

ICON is collaboratively developed with `git`. A GitLab server by the German Climate Computing Center (DKRZ) is used as centralized exchange platform. "In order to use the GitLab resource, project administrators need to activate the "GitLab" permission per user on <https://luv.dkrz.de> before a user can login to <https://gitlab.dkrz.de> with his/her LDAP password." ¹

This means:

1. One needs to register in luv: <https://luv.dkrz.de/projects/newuser/>

The image shows two side-by-side screenshots of the DKRZ web interface. The left screenshot is the login page, featuring the DKRZ logo at the top, followed by input fields for 'Username' and 'Password', a 'Remember me' checkbox, and a blue 'Sign in' button. Below the login form are links for 'Register' and 'Forgot password?'. A green arrow points to the 'Register' link. The right screenshot is the 'Create new account' page, which includes fields for 'E-mail address*', 'Password*', 'Password confirmation*', 'First name*', and 'Last name*'. It also contains a checkbox for 'I am citizen of an EU/EEA member state and I also reside in an EU/EEA member state.*' and radio buttons for 'True' and 'False'.

¹<https://www.dkrz.de/up/systems/gitlab-git-repository-manager>

2. Once your luv account is activated, you have to “[Join existing project](#)” number 1125 *ICON-development*. Fill the message field with an appropriate note, e.g. “icon cosmo” (cosmo partners) or “icon nwp” (DWD) or similar, to be associated with the respective subproject.

Join existing project

You can ask the administrators of existing projects to let you join their project. Please contact a representative of the project you intend to work for beforehand.

By clicking **send**, your contact information including your nationality and county of residence will be submitted to the administrators of the selected project. For more information review our [privacy statement](#).

Project*

1125: ICON-development

Message

icon iwip

Send

3. Now, wait for admission by a project administrator.
4. Visit <https://gitlab.dkrz.de/icon> and checkout which repositories [you can access](#) and where you are a “Developer”.

There are several individual icon repositories, that represent the development process in the different teams contributing to icon. The different repositories are also called “flavors”. As of August 2021 there are these main flavors:

icon Common releases are coordinated through this repository and the other repositories merge from here regularly.

icon-aes, icon-les, icon-oes Used by atmosphere, land, and ocean departments at the MPI-M in Hamburg

icon-cimd Computational Infrastructure and Model Development group at the MPI-M.

icon-cscs ICON on GPU development by MeteoSwiss, ETH Zürich and CSCS. (Might be dropped in favor of nwp)

icon-dkrz Development at the DKRZ (Infrastructure and IO)

icon-kit Development at the KIT (related to ICON-ART and other)

icon-nwp Repository used by DWD, COSMO and CLM

The main/master branch often has individual names. For example, the main branch in icon-nwp is called “master” (it was “icon-nwp/icon-nwp-dev” before July 2021). You should start your developments from such a master branch. Try to merge new commits on the master branch regularly to your development branch to avoid merge conflicts later on. If possible, try to merge your changes back to a master as soon as possible.

1.1 First git checkout

Before you can clone the repository from GitLab, you have to register a [SSH key in your GitLab profile](#). The procedure is described in the [GitLab README](#) and the key commands are summarized in the following,

1. Generate a ED25519 ssh key. Follow the [README](#). For security reasons your ssh keys should be secured with a passphrase. E.g.:

```
ssh-keygen -t ed25519 -f "$HOME/.ssh/id_ed25519_gitlab.dkrz.de"
```

2. You can [setup and configure ssh-agent](#) on your machine in order to unlock your ssh passphrases just once per computer session.
3. It [is possible](#) to use different ssh keys for different services. This becomes handy for services with short expire limits. As a starting point, connect your ssh key and the `gitlab.dkrz.de` in your `~/.ssh/config` as follows.

```
echo -e "\n#GitLab DKRZ" >> ~/.ssh/config
```

```
echo "Host gitlab.dkrz.de" >> ~/.ssh/config
```

```
echo " PreferredAuthentications publickey" >> ~/.ssh/config
```

```
echo " IdentityFile ~/.ssh/id_ed25519_gitlab.dkrz.de" >> ~/.ssh/config
```

4. After creating the ssh key, you have to copy-paste the content of the new `*.pub` file to the [SSH settings in GitLab](#).

```
cat "$HOME/.ssh/id_ed25519_gitlab.dkrz.de.pub"
```

This is also described [in the README](#).

The screenshot shows the GitLab User Settings page for 'SSH Keys'. The left sidebar has 'SSH Keys' highlighted with a red arrow and the number '3'. The main content area has a search bar and a section titled 'SSH Keys' with a sub-section 'Add an SSH key'. Below this, there is a text area for pasting a public SSH key, with a red arrow and the number '4' pointing to it. The key text is: `ssh-ed25519 AA1IZD11NTE5CN3LAAAACIY5MjjVMD9BqAAC3Nza2B/fyBkpRwm9Z7Xo+ndovpfbtM majacob@omlws82`. Below the text area are fields for 'Title' (containing 'majacob@omlws82') and 'Expires at' (empty). A red arrow and the number '5' point to the 'Add key' button. In the top right corner, a user profile dropdown menu is visible with a red arrow and the number '1' pointing to it, and another red arrow and the number '2' pointing to the 'Edit profile' option.

5. [Verify that you can connect](#) by running:

```
ssh -T git@gitlab.dkrz.de
```

Now you can clone a repository. For example to start with master do

```
git clone --recursive git@gitlab.dkrz.de:icon/icon-nwp.git
# checkout the master branch
git checkout master
# create your own feature branch
git checkout -b icon-nwp/title-of-my-new-feature
# now you can work on your branch.
```

See also the “Step 1 Working on the feature branch” in the [Overview of global ICON workflow in the wiki](#).

As a “Developer” in the “nwp” group, you can create and push your own branches to icon-nwp. However, you can not push to the main branch, the master. In order to get your changes upstream, you have to propose a “Merge Request” in GitLab and notify a “Gatekeeper”, a person who is a GitLab project “Owner”. See https://gitlab.dkrz.de/groups/icon/nwp/-/group_members?sort=access_level_desc.

Remark: Make sure that you have used `--recursive` option with `git clone`. Otherwise do:

```
git submodule update --init --recursive
```

2 Compiling ICON

Visit any ICON repository on GitLab to find a README.md describing how to build ICON in general. Note that institute/machine specific configure “wrappers can be found in the respective subdirectories of the `./config` directory.”

The “out-of-source” build is the recommended way of building to keep the source directory clean. In this way, you can easily clean-up one build or work with several builds (e.g. different compilers) at the same time. For example:

```
mkdir build-linuxWS.gcc && cd build-linuxWS.gcc
/path/to/icon-nwp/config/dwd/linuxWS.gcc
make -j 8
```

Or you can make a `build` directory within the ICON source directory.

```
cd /path/to/icon-nwp/
mkdir build && cd build
../config/dwd/linuxWS.gcc
make -j 8
```

3 Further reading

See the [ICON Tutorial](#)² for further reading on:

- Installation
- Input data
- Description of the dynamic core, the physics-dynamics coupling and the physical parameterizations
- Simulation of idealized test cases
- Global and regional weather forecasts
- Parallelization and output
- Programming in ICON
- Visualization
- ICON's data assimilation system

Selected topics are also covered in the “[Reports on ICON](#)”.

The [ICON Developer Wiki](#) includes a section on technical topics related to the **GPU programming** of ICON.

Also note the presence of **Buildbot**. Buildbot is a software development tool which automates compile and test cycles of ICON required to validate changes to the project code base. If you want to merge new code you should check that all buildbot tests are green for your feature branch. The buildbot procedure is [described in the ICON developer wiki](#). Use our DKRZ credentials to log in into <https://buildbot.dkrz.de/>. The buildbot runs take some time (roughly 30 minutes each) but you have to wait for free resources. Check the grid view <https://buildbot.dkrz.de/grid?width=10> and note that you can use the `width=10` parameter to change the number of columns.

Please follow the guide in <run/checksuite.rcnl.dwd.de/README.TXT> for equivalent tests on the DWD NEC Aurora HPC.

For people working the with **SX-AURORA DWD HPC**, there is “A Short User Guide for DWD's new HPC System” that can help you with your first steps.

Finally, there is an **outdated ICON programming style guide** included in the ICON repository: icon-2.6.3-rc/doc/style/icon_standard.pdf.

²[DOI:10.5676/DWD_pub/nwv/icon_tutorial2020](https://doi.org/10.5676/DWD_pub/nwv/icon_tutorial2020)