

# PANDAROOT – DOCKER

BATTERIES INCLUDED

ROMAN KLASSEN

RUHR UNIVERSITY BOCHUM

26.10.2021



# THE GOAL

Use PandaRoot on any machine (with docker) without installing FairSoft/FairRoot.  
Avoid dependency/compiler hell.

- MC Data Generation on HPC
- Analysis on Workstation
- Software Development
- Probably more, add your own

Preferred workflow: simple docker pull.

Advantage: easy setup and clean separation from host system!

Problem: people have different requirements

- Image should be easily malleable (un-Docker-ish)
- Image should be ready out of the box (large file size)
- Image should be small (not self contained)
- Image may use existing FairSoft/FairRoot

Can't be all at once: prepare different images.

## EXAMPLE 1: DEV WORK

I've prepared a Docker image that includes:

- FairSoft (master branch)
- FairRoot (master branch)
- All necessary packages for PandaRoot

Use PandaRoot source code from Host computer. Edit/develop on host, compile/use in container.

FairSoft/FairRoot image is on Docker hub, but that's not possible for PandaRoot (yet) due to licensing concerns.

## TRY IT!

On your host machine (laptop, workstation):

Clone the PandaRoot repo

```
git clone git@git.panda.gsi.de:PandaRootGroup/PandaRoot.git
```

Pull docker image, pass PandaRoot dir and run

```
docker run -it -v $(pwd)/PandaRoot:/mnt/work/PandaRoot rklasen/fairroot:latest
```

You're now inside the docker container:

Compile PandaRoot

```
mkdir PandaRoot/build && cd PandaRoot/build && cmake ../  
make -j8
```

Note: better use bind mount and uid mappings, see GitLab issue.

And that's already it. No package installation, no version incompatibilities, no fuss.

```
run test simulation
```

```
cd macro/run  
root -l -q -b sim_complete.C
```

**But:** the docker image is huge (40GB). Docker image is currently uploading to docker hub, but should be done sometime today. Then you can try it.

## USE EXTERNAL FAIRSOFT/FAIRROOT

GSI supplies us with pre-built FairSoft and FairRoot as cvmfs mount. So we don't actually have to d/l this huge docker image, we can just use their artefacts. There are 3 different ways this is done:

1. Mount on your host system (many HPC systems)
2. Mount in workspace container
3. Mount via dedicated container and share to host / other containers

# 1: USE HOST CVMFS

Mount cvmfs via host and pass it to the docker container:

magic cvmfs mount commands

didn't succeed yet

Pass to docker container

```
docker run -it -v $(pwd)/PandaRoot:/mnt/work/PandaRoot -v /cvmfs:/cvmfs  
rklasen/pandaroot:cvmfs
```

**Pro:** well suited for HPC systems like HIMSTER 2

**Cons:** no clean host/container separation



## 2: MOUNT CVMFS IN WORKSPACE CONTAINER

Mount cvmfs inside workspace container instead of host. Include in FairSoft/FairRoot image.

**Pro:** self contained, clean separation

**Cons:** requires "--privileged" flag for docker container to FUSE mount

So that's not happening on a HPC system, and for local workstations there is no real advantage to the first method

### 3: MOUNT VIA DEDICATED CONTAINER

Cern provides a container image that mounts any cvmfs share to the host mount.

- run Cern Container, creates /cvmfs on host
- pass host /cvmfs to PandaRoot container

Very elegant, but same con as above:

**requires `-privileged` flag**

So unless there is strong demand, I'm not pursuing this.

## JOIN THE DISCUSSION

If you have a different requirement or can help, join the Discussion:

GitLab Issue

<https://git.panda.gsi.de/PandaRootGroup/PandaRoot/-/issues/262>

**THANKS FOR YOUR ATTENTION!**