



# PALM steering

---

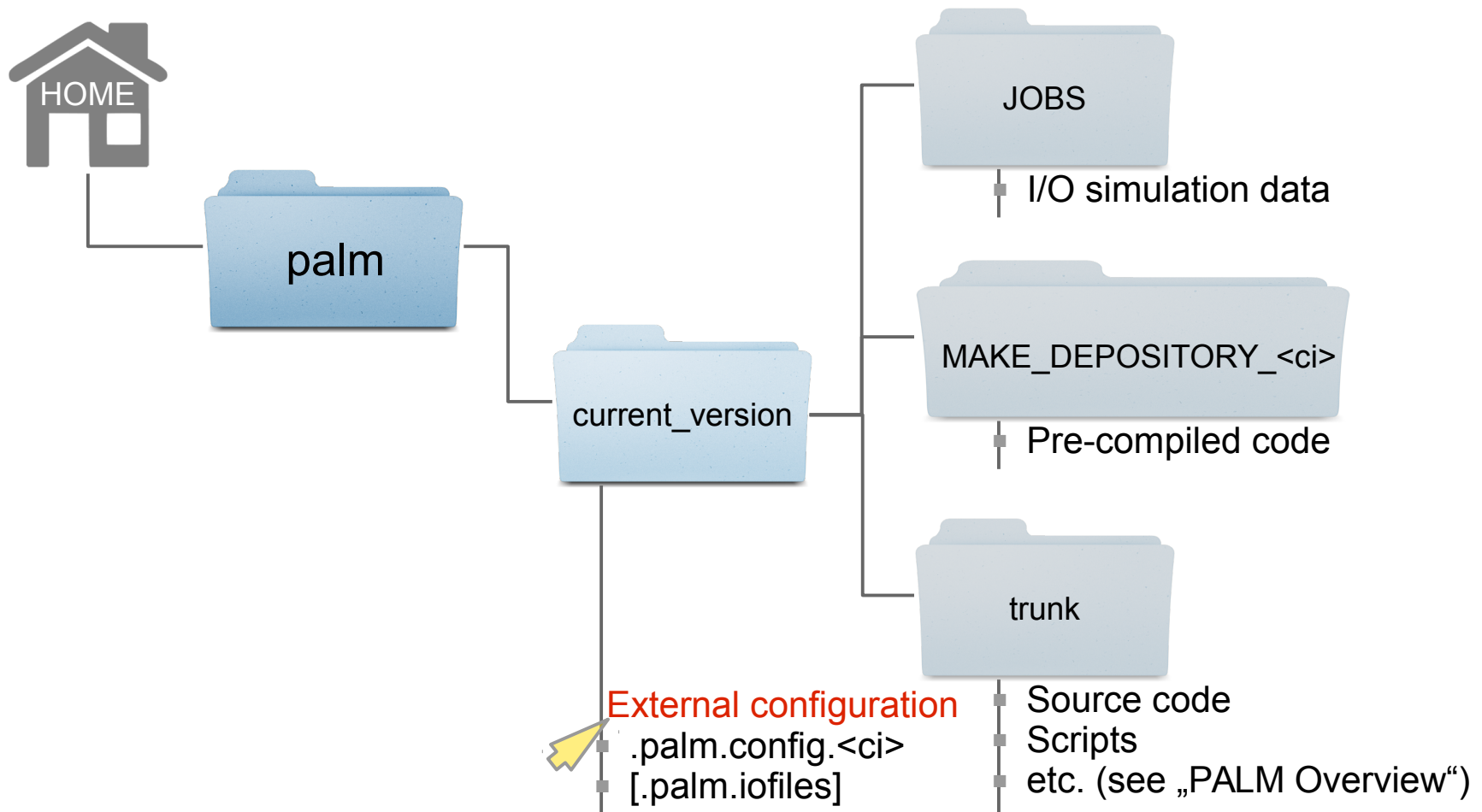
last update: 26. Feb 2018



Institute of Meteorology and Climatology, Leibniz Universität Hannover

## PALM directory structure

How to configure PALM I/O and environment?



## External configuration – Overview (I)

Which steps need to be taken care of to run PALM?

- (1) Compile default PALM source code (script **palmbuild**)
  - MAKE\_DEPOSITORY\_<configuration\_identifier> created, containing pre-compiled source code
  - Done by automatic installer
  - **palmbuild** has to be called manually after each code update!
- (2) Compile PALM user-interface code, if existing
- (3) Execute PALM (a.out)
  - PALM is executed in a temporary directory **<jobname>.<randomnumber>**, which should reside on a fast file system (see variable **%fast\_io\_catalog** on next slide).
- (4) Copy/transfer input & output files
  - Copy input files to **<jobname>.<randomnumber>**, from there, copy output files to destination.

## External configuration – Overview (II)

Which steps need to be taken care of to run PALM?

(5) Cycle numbering of output files in order to maintain output in case running the same job multiple times (e.g. restarts).

**Otherwise, output would be overwritten!**

(6) By default, directory `<jobname>.<randomnumber>` is deleted at the end

The bash script **palmsrun** takes care of steps 3-6, and it calls **palmbuild** in case that `MAKE_DEPOSITORY_<ci>` doesn't exist (step1, new installation) or user-defined code shall be used (step 2)!

Required information for **palmsrun** must be specified by the user in PALM's configuration file (see next slides).

## External configuration – PALM environment configuration file (I)

current\_version

`.palm.config.<configuration identifier>`

```

# $Id: .palm.config.imuk 2411 2017-09-06 08:28:02Z raasch $
# column 1          column 2
# name of variable  value of variable (~ must not be used)
#-----
%base_directory     $HOME/palm/current_version
%base_data          ~/palm/current_version/JOBS
%source_path        $HOME/palm/current_version/trunk/SOURCE
%user_source_path   $base_directory/JOBS/$fname/USER_CODE
%fast_io_catalog    <your_path_to_fast_file_system>
#
%local_ip           <ip_address_of_your_local_pc>
%local_username     <your_username_on_local_pc>
#
%compiler_name      mpif90
%compiler_name_ser  ifort
#
%cpp_options        -cpp -D_parallel -DMPI_REAL=MPI_DOUBLE_PRECISION
                   -DMPI_2REAL=MPI_2DOUBLE_PRECISION -D_fftw -D_netcdf
#
%compiler_options   -openmp -fpe0 -O3 -xHost -fp-model source -ftz -fno-alias ...
                   -I /<path_to>/fftw/3.3.4/include ...
                   -L/<path_to>/fftw/3.3.4/lib64 -lfftw3 ...
%linker_options     -openmp -fpe0 -O3 -xHost -fp-model source -ftz -fno-alias ...
                   -I /<path_to>/fftw/3.3.4/include ...
                   -L/<path_to>/fftw/3.3.4/lib64 -lfftw3 ...
#
%hostfile           auto
%execute_command    mpiexec -machinefile hostfile -n {{mpi_tasks}} palm
#
#-----
# INPUT-commands, executed before running PALM - lines must start with "IC:"
#-----
# IC:
#-----
# ERROR-commands - executed when program terminates abnormally
#-----
EC:[[ \${locat} = execution ]]  && cat  RUN_CONTROL
EC:[[ \${locat} = execution ]]  && cat  PARTICLE_INFOS/*
#-----
# OUTPUT-commands - executed when program terminates normally
#-----
#
# Email notification in case of finished PALM run
OC:mail username@hostname "PALM run with job id {{job_id}} finished"

```

- Created during automatic installation
- Computer- and software-specific configuration
- Variables are interpreted by **palmlrun** script
- `<configuration identifier>` (short `<ci>`) can be an arbitrary string
- One individually named file per environment in that PALM shall be executed (compiler options, computer, software)
- Detailed documentation available soon:  
<https://palm.muk.uni-hannover.de/trac/wiki/doc/app/palmlrun>

## External configuration – PALM environment configuration file (II)

current\_version

`.palm.config.<configuration_identifier>`

```

# $Id: .palm.config.imuk 2411 2017-09-06 08:28:02Z raasch $
# column 1          column 2
# name of variable  value of variable (~ must not be used)
#-----
%base_directory    $HOME/palm/current_version
%base_data         ~/palm/current_version/JOBS
%source_path       $HOME/palm/current_version/trunk/SOURCE
%user_source_path  $base_directory/JOBS/$fname/USER_CODE
%fast_io_catalog   <your_path_to_fast_file_system>
#
%local_ip          <ip_address_of_your_local_pc>
%local_username    <your_username_on_local_pc>
#
%compiler_name     mpif90
%compiler_name_ser ifort
#
%cpp_options       -cpp -D_parallel -DMPI_REAL=MPI_DOUBLE_PRECISION
                  -DMPI_2REAL=MPI_2DOUBLE_PRECISION -D_fftw -D_netcdf
#
%compiler_options  -openmp -fpe0 -O3 -xHost -fp-model source -ftz -fno-alias ...
                  -I /<path_to>/fftw/3.3.4/include ...
                  -L/<path_to>/fftw/3.3.4/lib64 -lfftw3 ...
%linker_options    -openmp -fpe0 -O3 -xHost -fp-model source -ftz -fno-alias ...
                  -I /<path_to>/fftw/3.3.4/include ...
                  -L/<path_to>/fftw/3.3.4/lib64 -lfftw3 ...
#
%hostfile          auto
%execute_command   mpiexec -machinefile hostfile -n {{mpi_tasks}} palm
#
#-----
# INPUT-commands, executed before running PALM - lines must start with "IC:"
#-----
# IC:
#-----
# ERROR-commands - executed when program terminates abnormally
#-----
EC:[[ \${locat} = execution ]] && cat RUN_CONTROL
EC:[[ \${locat} = execution ]] && cat PARTICLE_INFOS/*
#-----
# OUTPUT-commands - executed when program terminates normally
#-----
#
# Email notification in case of finished PALM run
OC:mail username@hostname "PALM run with job id {{job_id}} finished"

```

Header

Path variables

Information about local computer

Compiler names (parallel and serial)

Pre-processor directives

Compiler and linker options, including software library paths

Execute command  
(depends on MPI library that is used)

UNIX commands to be executed...

- before PALM execution
- in case of an error during execution
- after PALM execution

**.palm.config.<ci> must be adapted for using batch mode and submission of batch jobs from a local to a remote computer!**

## External configuration – PALM environment configuration file (III)

current\_version

`.palm.config.<configuration_identifier>`

```
#$Id: .palm.config.imuk 2411 2017-09-06 08:28:02Z raasch $
#column 1          column 2
#name of variable  value of variable (~ must not be used)
#-----
%base_directory    $HOME/palm/current_version
%base_data         ~/palm/current_version/JOBS
%source_path       $HOME/palm/current_version/trunk/SOURCE
%user_source_path  $base_directory/JOBS/$jobname/USER_CODE
%fast_io_catalog   <your_path_to_fast_file_system>
%local_jobcatalog  <your_path_to_jobcatalog_on_local_PC>
%remote_jobcatalog <your_path_to_jobcatalog_on_remote_PC>
#
%local_ip          <ip_address_of_your_local_pc>
%local_username    <your_username_on_local_pc>
#
%remote_ip         <ip_address_of_your_remote_pc>
%remote_username   <your_username_on_remote_pc>
%remote_loginnode  <hostname_of_login_node_on_remote_pc>
#
%defaultqueue     workq
%submit_command   /opt/pbspro/default/bin/qsub
... ..
#
%execute_command  mpirun -machinefile hostfile -n {{mpi_tasks}} palm
#
#
# BATCH-directives to be used for batch jobs.
BD:#!/bin/bash
BD:#PBS -N{{job_id}}
BD:#PBS -l walltime={{cpu_hours}}:{{cpu_minutes}}:{{cpu_seconds}}
BD:#PBS -l ncpus={{cores}}
BD:#PBS -o{{job_protocol_file}}
BD:#PBS -j oe
BD:#PBS -q{{queue}}
#
# BATCH-directives for batch jobs used to send back the jobfile from a remote to a local
host
BDT:#!/bin/bash
BDT:#PBS -N job_protocol_transfer
BDT:#PBS -l walltime=00:30:00
BDT:#PBS -l ncpus=1
BDT:#PBS -o{{job_transfer_protocol_file}}
BDT:#PBS -j oe
BDT:#PBS -q workq
#
#-----
# INPUT-commands, executed before running PALM - lines must start with "IC:"
#-----
... ..
```

Path to job protocol files

Information about local PC

Information about remote PC

Default batch queue & job submit cmd

Execute command

(depends on MPI library that is used)

Batch directives for submitting a

PALM run as batch job

(depends on batch system, here PBS)

Batch directives for submitting a

single-processor job to send back the

job protocol file from remote to local PC

## External configuration – PALM I/O configuration file

### .palm.iofiles

```

#-----
# List of input-files
#-----
PARIN          in:tr      d3r      $base_data/$jobname/INPUT      _p3d*
PARIN          in:tr      d3r      $base_data/$jobname/INPUT      _p3dr*
PARIN          in:tr      pcr      $base_data/$jobname/INPUT      _pcr*
TOPOGRAPHY_DATA inopt:tr   d3#:d3r  $base_data/$jobname/INPUT      _topo*
PIDS_STATIC    inopt:tr   d3#:d3r  $base_data/$jobname/INPUT      _static*
PIDS_DYNAMIC   inopt:tr   d3#:d3r  $base_data/$jobname/INPUT      _dynamic*
...
BININ          in:lnpe    d3r:rec  $fast_io_catalog/$jobname/RESTART _d3d*
PARTICLE_RESTART_DATA_IN in:lnpe    prtr     $fast_io_catalog/$jobname/RESTART _rprtr
DATA_1D_PR_NETCDF inopt      prr      $base_data/$jobname/OUTPUT      _pr*      nc
...
DATA_2D_YZ_NETCDF inopt      yzr      $base_data/$jobname/OUTPUT      _yz*      nc
DATA_2D_YZ_AV_NETCDF inopt      yzr      $base_data/$jobname/OUTPUT      _av_yz*   nc
DATA_3D_NETCDF    inopt      3dr      $base_data/$jobname/OUTPUT      _3d*      nc
DATA_3D_AV_NETCDF inopt      3dr      $base_data/$jobname/OUTPUT      _av_3d*   nc
...
DATA_PRT_NETCDF   inopt:pe   prtr     $base_data/$jobname/OUTPUT      _prt*
#-----
# List of output-files
#-----
BINOUT*        out:lnpe   restart  $fast_io_catalog/$jobname/RESTART _d3d
PARTICLE_RESTART_DATA_OUT* out:lnpe   prt#:prt $fast_io_catalog/$jobname/RESTART _rprtr
#
RUN_CONTROL*    out:tr     d3#:pcr  $base_data/$jobname/MONITORING   _rc
RUN_CONTROL*    out:tra    d3r      $base_data/$jobname/MONITORING   _rc
HEADER*         out:tr     d3#:pcr  $base_data/$jobname/MONITORING   _header
HEADER*         out:tra    d3r      $base_data/$jobname/MONITORING   _header
CPU_MEASURES*   out:tr     d3#:pcr  $base_data/$jobname/MONITORING   _cpu
CPU_MEASURES*   out:tra    d3r      $base_data/$jobname/MONITORING   _cpu
PARTICLE_INFO*  out:tr     pt#      $base_data/$jobname/MONITORING   _prt_info
PARTICLE_INFO*  out:tra    ptr      $base_data/$jobname/MONITORING   _prt_info
LIST_PROFIL*    out:tr     d3#      $base_data/$jobname/MONITORING   _list_pr
#
DATA_1D_PR_NETCDF* out:tr     *        $base_data/$jobname/OUTPUT      _pr      nc
...
DATA_2D_YZ_NETCDF* out:tr     *        $base_data/$jobname/OUTPUT      _yz      nc
DATA_2D_YZ_AV_NETCDF* out:tr     *        $base_data/$jobname/OUTPUT      _av_yz   nc
DATA_3D_NETCDF*   out:tr     *        $base_data/$jobname/OUTPUT      _3d      nc
DATA_3D_AV_NETCDF* out:tr     *        $base_data/$jobname/OUTPUT      _av_3d   nc
...
DATA_PRT_NETCDF*  out:pe     *        $base_data/$jobname/OUTPUT      _prt
DATA_PRT_NETCDF*  out:trpe   *        $base_data/$jobname/OUTPUT      _prt
PARTICLE_DATA*    out:lnpe   *        $base_data/$jobname/OUTPUT      _prt_dat
...

```

- Default file is at ...trunk/SCRIPTS, and used from there by palmrun. If user needs to modify it, .palm.iofiles must be copied to current\_version folder.
- Definition of **input/output files and paths**, interpreted by **palmrun** script
- Definition of **activation strings**
- While user could have multiple .palm.config.<ci> files with different configurations (e.g. different compiler options), only one .palm.iofiles is required, since path variables **base\_data** and **fast\_io\_catalog** are set in .palm.config.<ci>, and **jobname** equals the job name (see next slide)



## Starting a run – palmrun options

### **palmrun ...**

(important options for interactive run on local PC)

- a „d3# ...“  
(**activation strings**, possible strings listed in .palm.iofiles)
- d <jobname>  
(taken as basefile/directory name of all data files of this run)
- h <configuration\_identifier>  
(as in configuration file name .palm.config.<ci>)
- X <number of processor cores (PEs) to be used>
- T <number of MPI tasks to be started per computer node>

(for batch mode and remote jobs, additional options are available, see palmrun documentation)

- b  
(run in batch mode)
- q <batch-job queue to be used>
- t <CPU time (in seconds)>

**palmrun ?** lists all available options

## Starting a run – palmrun execution

Directly type **palmrun** command-line into UNIX terminal

```
ostria:~/palm/current_version[1117]$ palmrun -d example_cbl -a "d3#" -h "imukpalm4u" -X "4" -T "4"

*** palmrun 1.0 Rev: 2789 $
    will be executed.      Please wait ...

    Reading the configuration file...
    Reading the I/O files...

*** INFORMATIVE: additional source code directory
    "/home/kanani/palm/current_version/JOBS/example_cbl/USER_CODE"
    does not exist or is not a directory.
    No source code will be used from this directory!

#-----#
| palmrun 1.0 Rev: 2789 $                               Thu Feb  8 13:29:18 CET 2018 |
|-----|
| called on:                ostria                       |
| execution on:             imukpalm4u (IP:130.75.105.106) |
| running in:               interactive run mode         |
| number of cores:         4                             |
| tasks per node:         4 (number of nodes: 1)        |
|-----|
| cpp directives:          -cpp -D__parallel -DMPI_REAL=MPI_DOUBLE_PRECI |
|                          SION -DMPI_2REAL=MPI_2DOUBLE_PRECISION -D__ff |
|                          tw -D__netcdf -D__netcdf4     |
| compiler options:       -fpe0 -O3 -xHost -fp-model source -ftz -fno-a |
|                          lias -no-prec-div -no-prec-sqrt -ip -nbs -I / |
|                          muksoft/packages/fftw/3.3.4/include -L/muksof |
|                          t/packages/fftw/3.3.4/lib64 -lfftw3 -I /mukso |
|                          ft/packages/netcdf/4_intel/include -L/muksoft |
|                          /packages/netcdf/4_intel/lib -lnetcdf -lnetcd |
|                          ff                             |
| linker options:         -fpe0 -O3 -xHost -fp-model source -ftz -fno-a |
|                          lias -no-prec-div -no-prec-sqrt -ip -nbs -I / |
|                          muksoft/packages/fftw/3.3.4/include -L/muksof |
|                          t/packages/fftw/3.3.4/lib64 -lfftw3 -I /mukso |
|                          ft/packages/netcdf/4_intel/include -L/muksoft |
|                          /packages/netcdf/4_intel/lib -lnetcdf -lnetcd |
|                          ff                             |
| base name of files:      example_cbl                   |
| activation string list:  d3#                           |
|-----#

>>> everything o.k. (y/n) ? y
```

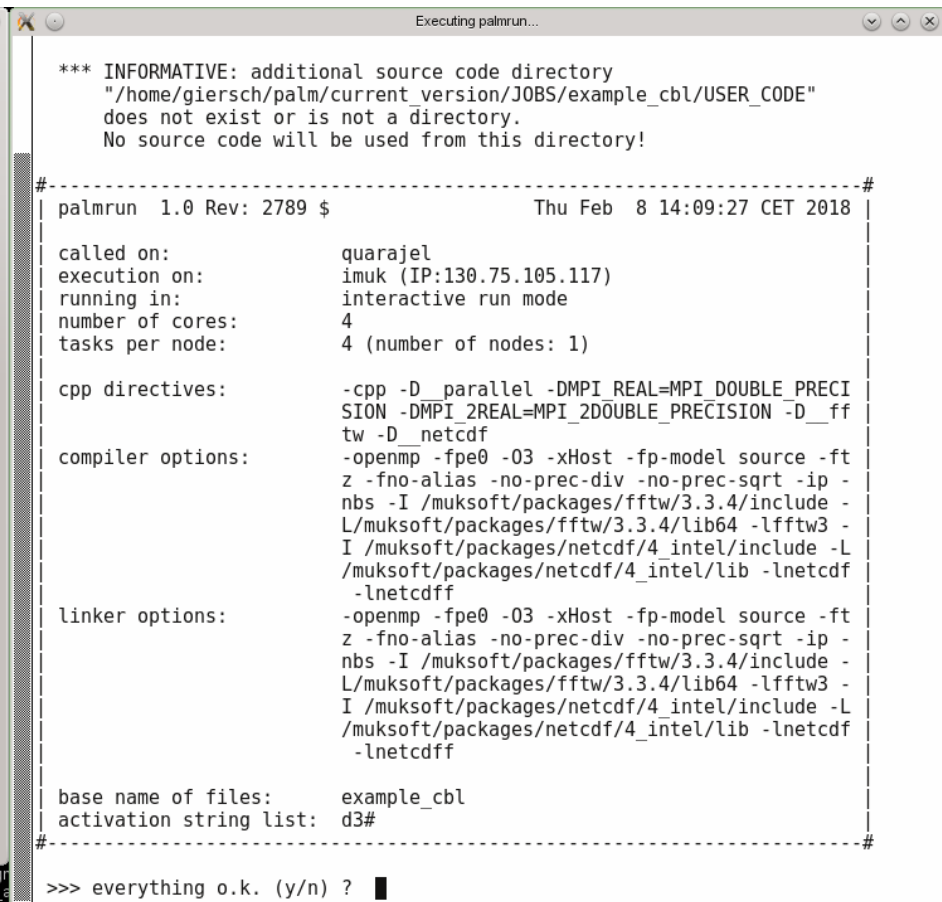
## Starting a run – palmrun execution with GUI

Generate **palmrun** UNIX command-line via graphical user interface (gui)

(<https://palm.muk.uni-hannover.de/trac/wiki/doc/app/palmrun>)

- Invoke **palmrungui** in UNIX terminal

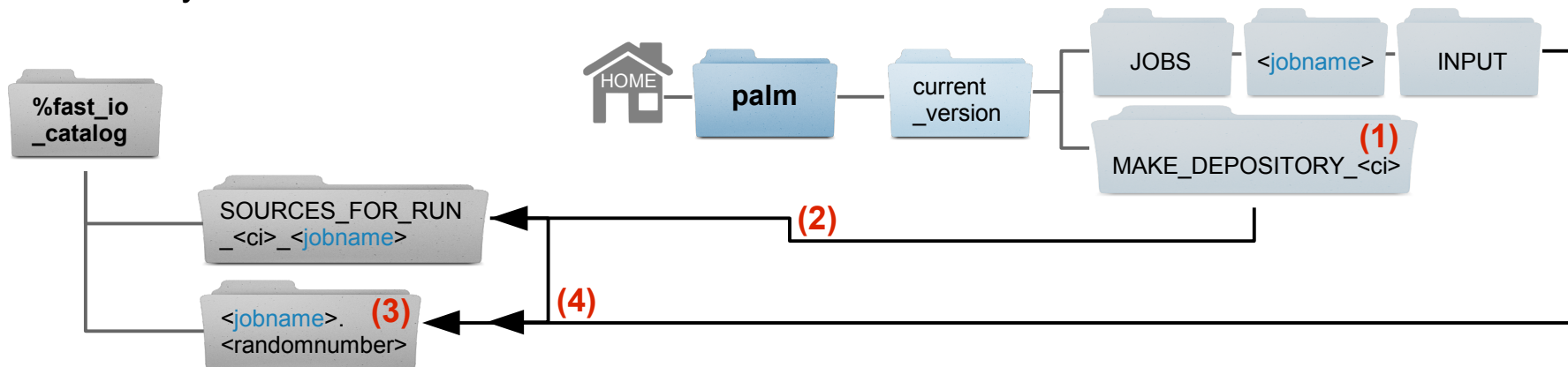
```
ostria:~/palm/current_version[1008]$ palmrungui
```



## Overview of palmrun's workflow (I)

```
palmrun -d <jobname> -a „d3# ... “ -h <ci> ...
```

- (1) Check if `SOURCE` code has been pre-compiled, i.e. does directory `MAKE_DEPOSITORY_<ci>` exist?  
If not, `palmrun` calls `palmbuild` to compile the code according to settings in `.palm.config.<ci>`
- (2) For every manual call of `palmrun`, copy all required sources for the job to `SOURCES_FOR_RUN_<ci>_<jobname>` in folder `%fast_io_catalog`. These sources are also used by automatic restarts.
- (3) Create temporary working directory `<jobname>.<randomnumber>` under path given by `%fast_io_catalog` in `.palm.config.<ci>`
- (4) Copy `INPUT` files and sources from `SOURCES_FOR_RUN_<ci>_<jobname>` to temporary directory.

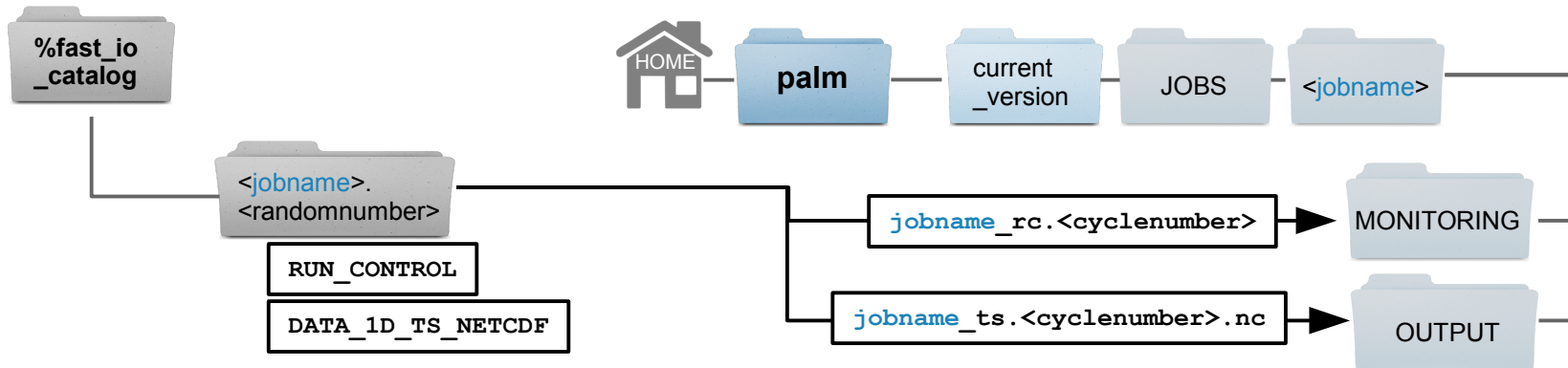
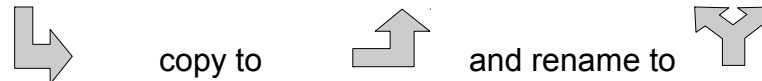


## Overview of palmrun's workflow (II)

```
palmrun -d <jobname> -a „d3# ... “ -h <ci> ...
```

- (5) Execute PALM in temporary directory
- (6) Copy **MONITORING** and **OUTPUT** files from temporary directory to their destination given in **.palm.iofiles**

```
#-----  
# List of output-files  
#-----  
RUN_CONTROL      out:tr  d3#:pcr  $base_data/$jobname/MONITORING  _rc  
...  
DATA 1D TS NETCDF out:tr  *      $base_data/$jobname/OUTPUT      ts  nc
```



- (7) Delete temporary directory (prevented by **palmrun** option „-B“). **SOURCES\_FOR\_RUN** directory is not deleted, remaining available for informational purposes and for restart run.