PALM steering

Institute of Meteorology and Climatology, Leibniz Universität Hannover

- This lecture gives instructions on how to run PALM locally in **interactive mode**, i.e. on your local PC/notebook or on any machine where you are logged in (called **local host**). PALM immediately starts to execute.

- Running PALM in **batch mode** (job-queuing/batch system) on a **local** or **remote** (super) computer/host is not part of this seminar. You can find further documentation about how to run PALM in batch mode via link:

  https://palm.muk.uni-hannover.de/trac/wiki/doc/app/palmrun

- Information on how to stop and continue a run – the so-called **restart mechanism** – will follow on Wednesday.

Definitions

- **Local host:** The computer that you are currently sitting at or are logged in via your terminal (ssh).
- **Remote host:** Any computer with a batch system, that you have ssh access to, but are not logged in now.
- **Batch mode:** Automatic processing of commands contained in a batch file, not interactively.

PALM folder structure

Where to find PALM after downloading and installing the PALM release?
(*download location="~/palm/current_version"*)
(*install_prefix="~/palm/current_version"*)

Working directory of PALM user

In case of several PALM versions that are used simultaneously it is beneficial to use the release number, e.g., 23.04

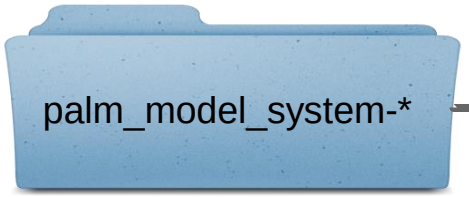
Pre-compiled RRTMG code

Must be created after installation

Files for steering, job monitoring, data output, user code

Pre-compiled code generated by **`palmbuild`**

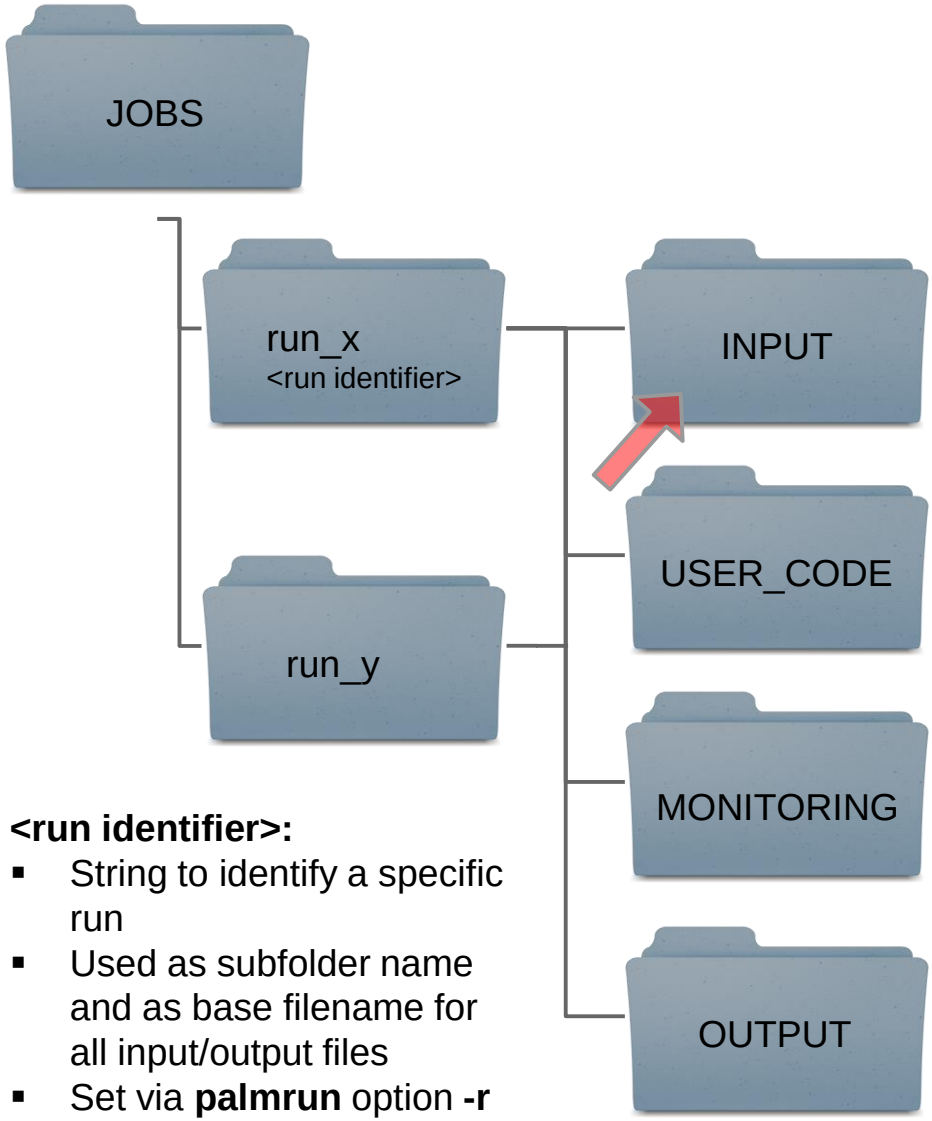
Scripts for execution



Download folder: Source code, scripts, etc. (see lecture „PALM Overview“)

PALM steering

PALM input/output overview



ASCII-Input | ASCII-Output | NetCDF

- Parameter file(s) for steering
- Simple topography data
- Driver files (e.g. topography data, soil properties, see lecture „static & dynamic drivers“)

- User-defined source code

(see lecture „User-defined code“)

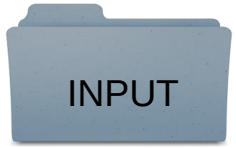
- Header file (parameter settings)
- Run control file (parameter settings, timestep information)
- CPU file (computing time performance measurements)

- 1D profiles (xy- & time-averaged)
- 2D sections & 3D data (instantaneous & time-averaged)

<run identifier>:

- String to identify a specific run
- Used as subfolder name and as base filename for all input/output files
- Set via **palmrun** option **-r**

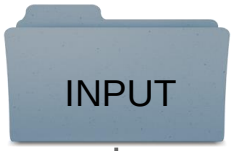
Model steering – Parameters overview



- Grid parameters (computational grid / processor grid):
  - grid spacing, number of grid points, etc.
- Numerical parameters & boundary conditions:
  - timestep scheme, advection scheme, etc.
  - horizontal (cyclic/non-cyclic),
  - vertical (constant-flux layer, free-slip condition, etc.)
- Initialization parameters:
  - initial profiles, restart run, main run (following pre-run)
- Output parameters:
  - 1d, 2d, 3d output, output intervals, etc.
- Physical parameters:
  - latitude, angular velocity of the earth, etc.
- Parameters for embedded modules (wind turbine model, land surface model, Lagrangian particle model, etc.)

PALM steering

Model steering – Input parameter file



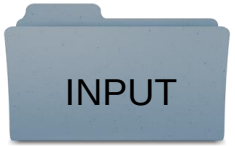
run_x_p3d (<run identifier>_p3d)

```
 &initiaization_parameters nx = 39, ny = 39, nz = 40,
          dx = 50.0, dy = 50.0, dz = 50.0,
          dz_stretch_level = 1200.0,

          fft_method = 'temperton-algorithm',

          initializing_actions = 'set_constant_profiles',
          ug_surface = 0.0, vg_surface = 0.0,
          pt_vertical_gradient       = 0.0, 1.0,
          pt_vertical_gradient_level = 0.0, 800.0,
          surface_heatflux = 0.1, bc_pt_b = 'neumann',

          topography = 'read_from_file', /

 &runtime_parameters  end_time = 3600.0,

          create_disturbances = .T.,
          dt_disturb = 150.0, disturbance_energy_limit = 0.01,

          dt_run_control = 0.0,

          data_output = 'w_xy', 'w_xz', 'w_xz_av', 'theta_xy', ,theta_xz',

          dt_data_output = 900.0,
          dt_data_output_av = 1800.0,
          averaging_interval = 900.0,
          dt_averaging_input = 10.0,

          section_xy = 2, 10, section_xz = 20,

          data_output_2d_on_each_pe = .F.,

          dt_dopr = 900.0,
          averaging_interval_pr = 600.0, dt_averaging_input_pr = 10.0,
          data_output_pr = '#theta', 'w"theta"', 'w*theta*', 'wtheta',
                           'w*2', ,theta*2', /
```

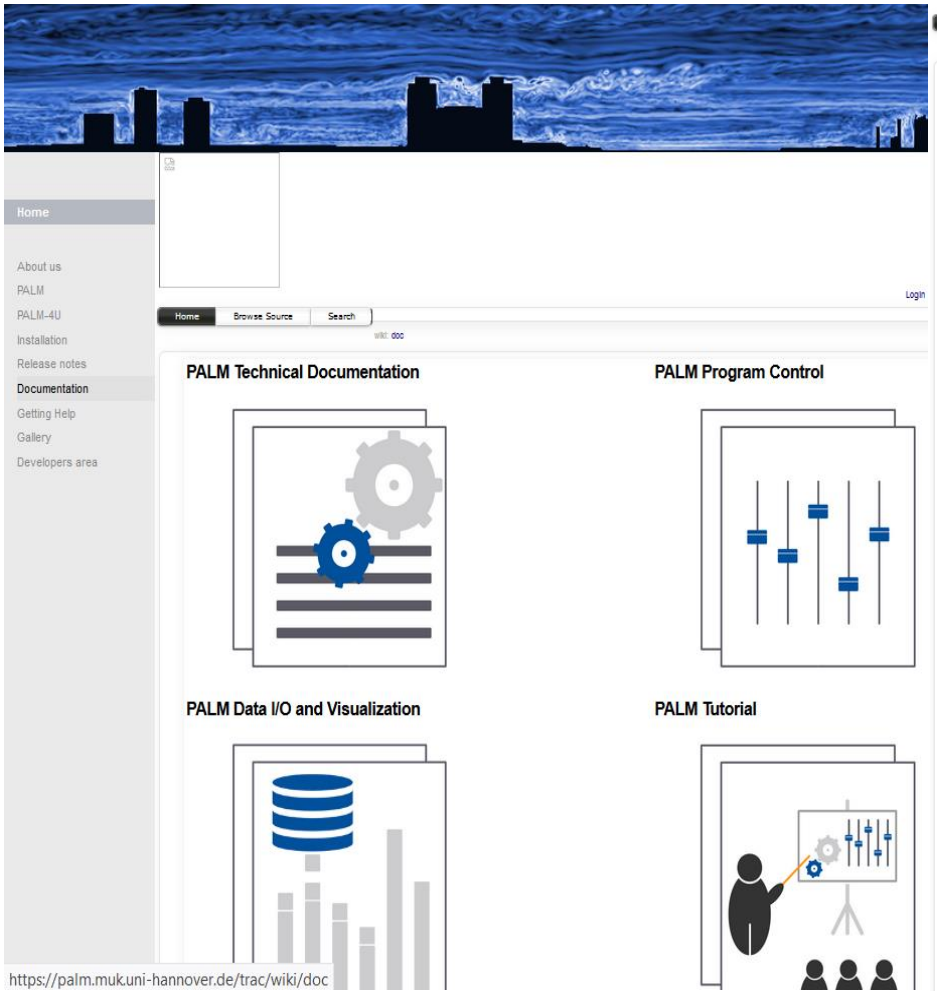
- Written in FORTRAN-NAMELIST syntax
- Read in by PALM
- Two NAMELIST groups mandatory:
  `&initialization_parameters`
  and
  `&runtime_parameters`
- Further NAMELISTs for specific modules of PALM, e.g.
  `&wind_turbine_parameters`
- Exact formatting required, e.g.:
  - NAMELIST starts with **&<name>** and ends with slash **/**
  - Parameters are separated by **,**
  - Strings placed between **'…'**
- Never use TAB's for alignment

Otherwise strange errors may appear!

Model steering – Parameter‘s documentation

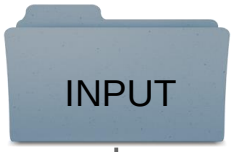


https://palm.muk.uni-hannover.de/trac/wiki/doc

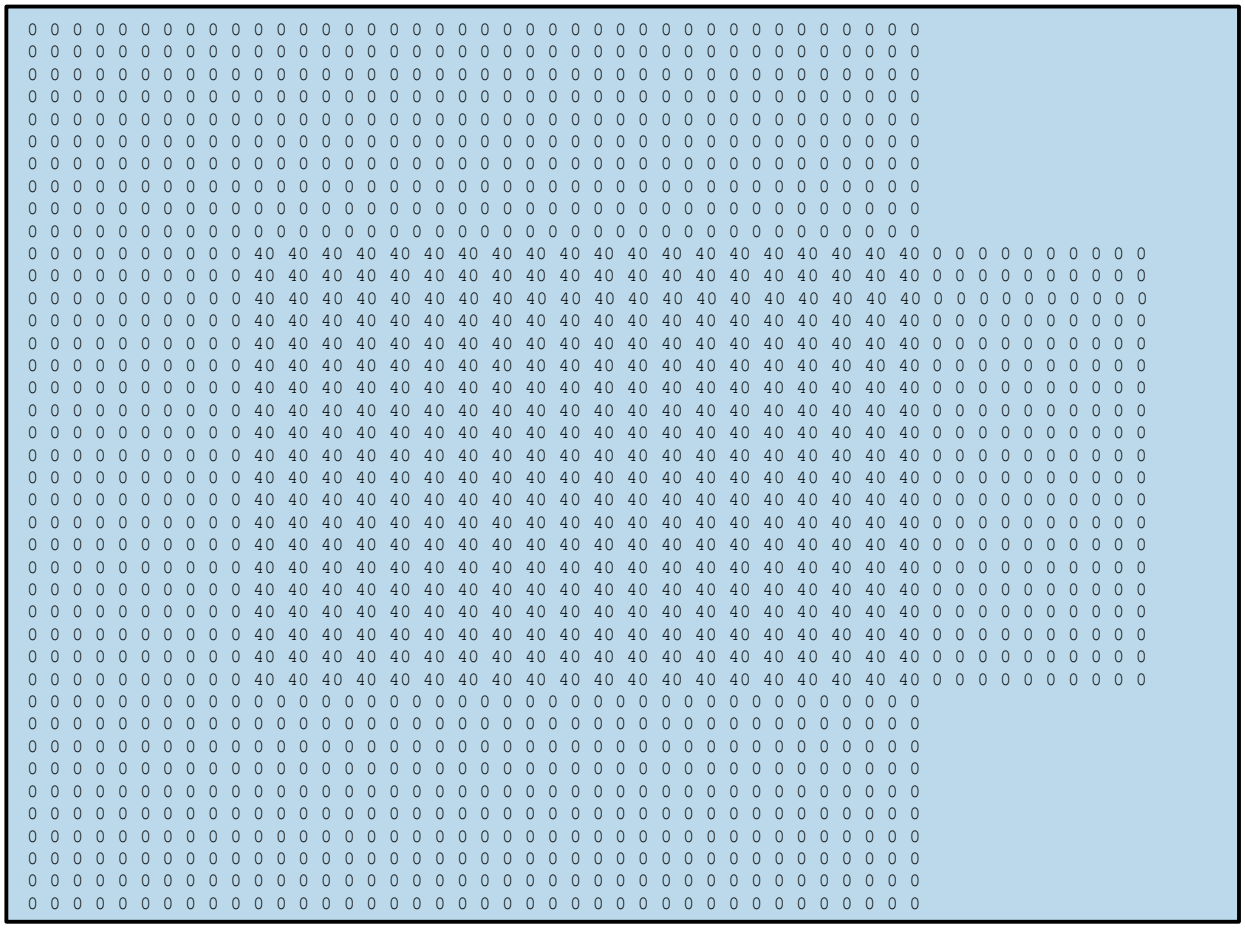


PALM steering

Model steering – Other input files, e.g.

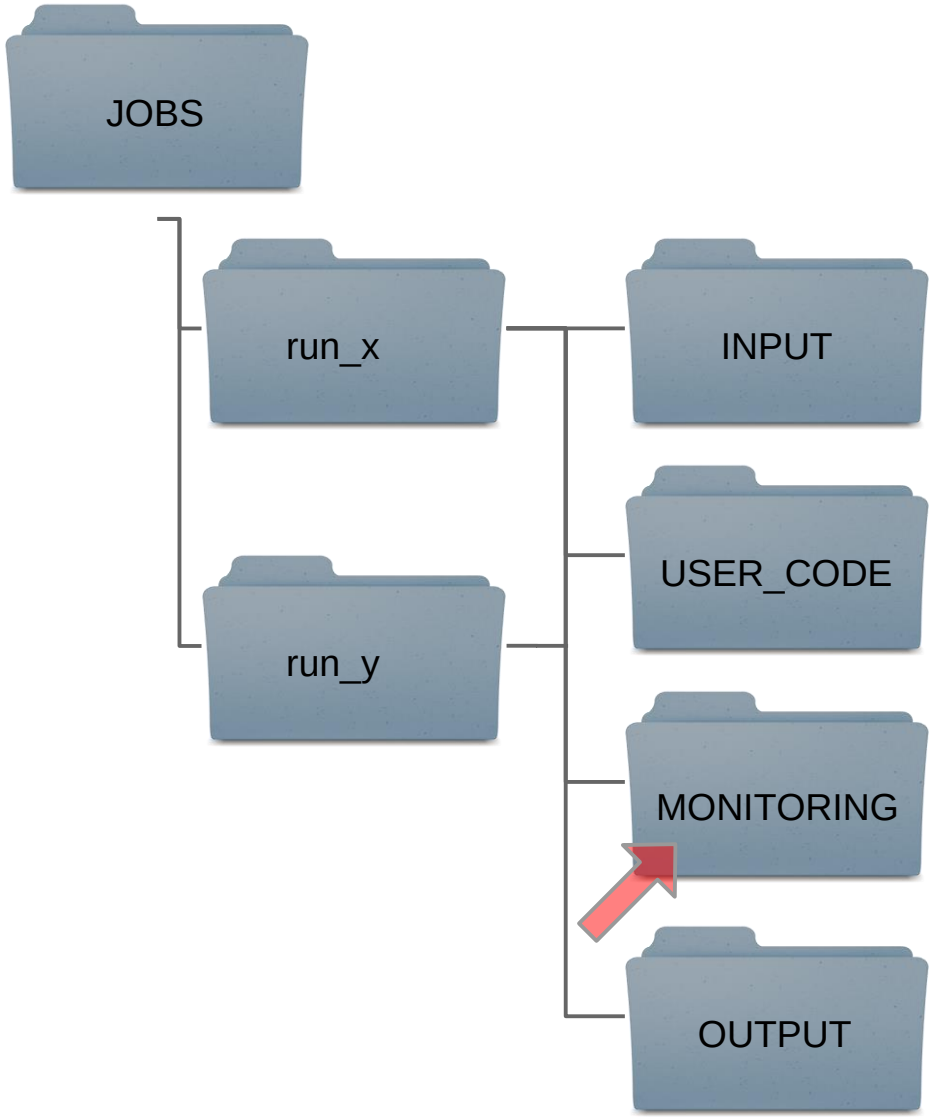


run_x_topo (<run_identifier>_topo)



- ASCII file with 2D topography height information (in meters above ground).
- Such a file will be used e.g. in the exercise E3 on Wednesday.
- Not suitable for complex topography. Instead, a static driver file in NetCDF format must be used (see lecture „static & dynamic driver” and E3).

PALM input/output overview



ASCII-Input | ASCII-Output | NetCDF

- Parameter file(s) for steering
- Simple topography data
- Driver files (e.g. topography data, soil properties, see lecture „static & dynamic drivers“)

- User-defined source code

(see lecture „User-defined code“)

- Header file (parameter settings)
- Run control file (parameter settings, timestep information)
- CPU file (computing time performance measurements)

- 1D profiles (xy- & time-averaged)
- 2D sections & 3D data (instantaneous & time-averaged)

PALM steering

Job monitoring – Header

run_x_header (<run identifier>_header.*, * = three digit number starting at 000)

```
******************************    ------------------------------------------
* PALM 23.04                 *    atmosphere - run without 1D - prerun
******************************    ------------------------------------------

Date:                31-07-23    Run:       example_cbl
Time:                14:56:25    Run-No.:   00
Run on host:          deafult
Number of PEs:             4    Processor grid (x,y): (   2,   2) calculated
--------------------------------------------------------------------------------

Numerical Schemes:
------------------

--> Use the boussinesq approximation for the model equations.
--> Solve perturbation pressure via FFT using temperton-algorithm routines
    perturbation pressure is calculated at every Runge-Kutta step
--> Momentum advection via Wicker-Skamarock-Scheme 5th order
--> Scalar advection via Wicker-Skamarock-Scheme 5th order
--> Loop optimization method: cache
--> Time differencing scheme: runge-kutta-3
--------------------------------------------------------------------------------


Run time and time step information:
----------------------------------

Timestep:             variable     maximum value: 20.000 s    CFL-factor: 0.90
Start time:              0.000 s
End time:             3600.000 s

Time reached:         3607.000 s
CPU-time used:           8.033 s     per timestep:                      0.031 s
                                     per second of simulated time:      0.002 s


Computational grid and domain size:
----------------------------------

Grid length:      dx =     50.000 m    dy =     50.000 m    dz =     50.000 m
Domain size:       x =   2000.000 m     y =   2000.000 m  z(u) =   2862.511 m
```

- Gives general information about the run.
- Summary/Information about the selected model parameters (physical and numerical values).
- File is generated for **every** run (initial as well as restart runs).

Job monitoring – Run control

- For initial runs, the information of the HEADER file is printed at the top of the RUN_CONTROL file.
- File is generated for **every** run, but the HEADER information is only printed for the initial run of a job chain.
- Subsequent to HEADER information, values of specific model variables follow, printed at certain timesteps of the simulation (one line for each timestep, output interval can be controlled by runtime parameter **`dt_run_control`**)
- **Contents of this timestep output should be carefully viewed after each run, as it allows a first check, if the model ran correctly!**

run_x_rc (<run identifier>_rc .*, * = three digit number starting at 000)

```
******************************    ------------------------------------------
* PALM 23.04                 *    atmosphere - run without 1D - prerun
******************************    ------------------------------------------

Date:                31-07-23    Run:       example_cbl
Time:                14:56:25    Run-No.:   00
Run on host:            lcmuk
Number of PEs:               4    Processor grid (x,y): (   2,   2) calculated
--------------------------------------------------------------------------------
...
Run-control output:
------------------

RUN  ITER. HH:MM:SS.SS    DT(E)     UMAX     VMAX     WMAX     U*    W*      THETA*     Z_I     ENERG.   DISTENERG    DIVOLD     DIVNEW     UMAX(KJ
------------------------------------------------------------------------------------------------------------------------------------------------
  0      0 00:00:00.00  20.0000A  -0.2131D -0.2288D  0.1182  0.000  1.38   0.000E+00    800.  0.104E-02  0.103E-02  0.597E-03  0.867E-13    7   7
  0      1 00:00:20.00  20.0000A  -0.2125  -0.2273   0.1172  0.009  0.69  -0.131E+02    100.  0.103E-02  0.102E-02  0.570E-05  0.411E-15    7   7
  0      2 00:00:40.00  20.0000A  -0.2117  -0.2252   0.1162  0.009  0.69  -0.132E+02    100.  0.102E-02  0.102E-02  0.939E-05  0.413E-15    7   7
  0      3 00:01:00.00  20.0000A  -0.2106  -0.2227   0.1152  0.009  0.69  -0.132E+02    100.  0.101E-02  0.101E-02  0.131E-04  0.417E-15    7   7
...
```

HEADER

PALM steering

Job monitoring – CPU measurements

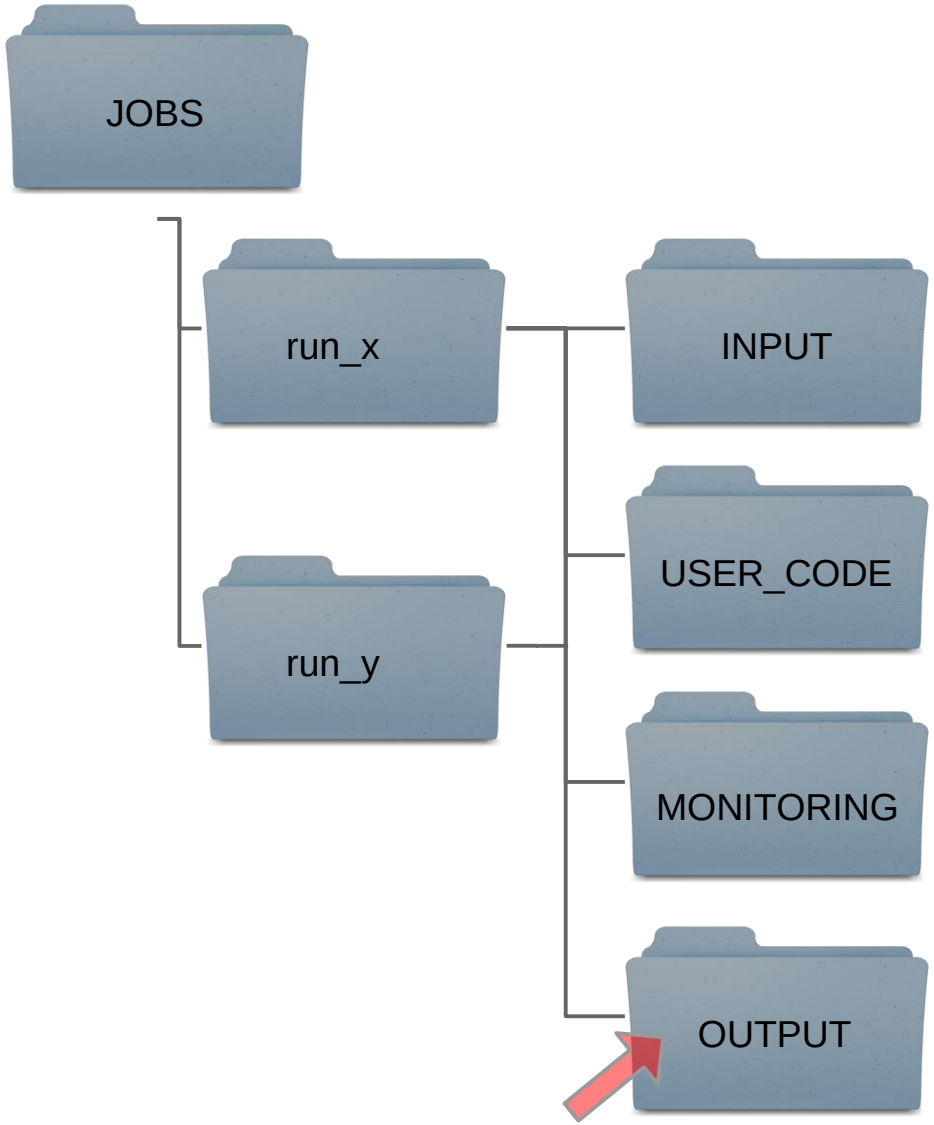
run_x_cpu (<run identifier>_cpu .*, * = three digit number starting at 000)


```
PALM 23.04   run: example_cbl.00  host: default  31-07-23 14:56:25
-----------

CPU measures for     4 PEs (    2(x) *     2(y) tasks *    1 threads):

gridpoints (x/y/z):                          40 *    40 *    40
nr of timesteps:                            263
cpu time per grid point and timestep:        0.45511 * 10**-6 s
-------------------------------------------------------------------

place:                              mean        counts      min       max       rms
                                  sec.      %               sec.      sec.      sec.
-------------------------------------------------------------------------------------
total                            8.043   100.00        1    8.043     8.043     0.000
all progn.equations              5.177    64.37      789    5.054     5.462     0.167
pres                             1.153    14.33      790    1.152     1.154     0.001
exchange-horiz-progn             0.607     7.55      789    0.304     0.738     0.177
surface_layer_fluxes             0.358     4.45      789    0.351     0.364     0.005
diffusivities                    0.322     4.00      789    0.316     0.328     0.004
flow_statistics                  0.239     2.97      264    0.239     0.240     0.001
calculate_timestep               0.059     0.73      264    0.051     0.062     0.005
sum_up_3d_data                   0.013     0.16      147    0.013     0.013     0.000
initialisation                   0.008     0.10        1    0.008     0.008     0.000
data_output_tseries              0.003     0.04      263    0.000     0.000     0.005
disturb_field                    0.003     0.04       10    0.003     0.003     0.000
data_output_2d                   0.002     0.03       10    0.002     0.002     0.000
run_control                      0.002     0.02      264    0.000     0.000     0.003
swap_timelevel                   0.001     0.01      789    0.001     0.001     0.000
last actions                     0.000     0.00        1    0.000     0.000     0.000
user_actions                     0.000     0.00      789    0.000     0.000     0.000
data_output_profiles             0.000     0.00        4    0.000     0.000     0.000
average_3d_data                  0.000     0.00        2    0.000     0.000     0.000

special measures:
-------------------------------------------------------------------------------------
timesteps                        8.033    99.87      263    8.033     8.033     0.000
exchange_horiz                   0.784     9.75     7136    0.483     0.918     0.177
poisfft                          0.647     8.04      790    0.646     0.647     0.001
fft_x                            0.187     2.33      790    0.187     0.188     0.000
fft_y                            0.187     2.33      790    0.186     0.188     0.001
transpo forward                  0.140     1.74      790    0.139     0.142     0.001
divergence                       0.137     1.70     1054    0.136     0.137     0.000
mpi_alltoall                     0.102     1.26     4740    0.101     0.102     0.000
transpo invers                   0.080     0.99      790    0.078     0.080     0.001
tridia                           0.051     0.63      790    0.050     0.051     0.000
```

- Contains information about the CPU requirements of single parts of the program.
- **all progn. equations** and **pres solver** should be the main consumers.
- For larger grids (e.g., 1024^3 points), **pres** may need up to 50% of the total time.
- Time needed for communication (sum of **exchange_horiz** and **mpi_alltoall**) should not exceed ~20-30% of the total time.
- The load balance among the processor cores can be inspected via **min**, **max** and **rms** (root-mean-square) values → They may indicate network problems!

PALM steering

PALM input/output overview



ASCII-Input | ASCII-Output | NetCDF

- Parameter file(s) for steering
- Simple topography data
- Driver files (e.g. topography data, soil properties, see lecture „static & dynamic drivers“)

- User-defined source code

(see lecture „User-defined code“)

- Header file (parameter settings)
- Run control file (parameter settings, timestep information)
- CPU file (computing time performance measurements)

- 1D profiles (xy- & time-averaged)
- 2D sections & 3D data (instantaneous & time-averaged)

PALM steering

Data output – Available standard output

Type	Possibilities	Local file name generated by PALM run_x_<...>	Filename within temporary working directory and source code
time series	instantaneous	ts.*.nc	DATA_1D_TS_NETCDF
mean vertical profiles	horizontally-averaged & instantaneous or time-av.	pr.*.nc	DATA_1D_PR_NETCDF
2D cross sections (xy, xz, yz)	instantaneous or time-av., possible average along 3rd dimension	xy.*.nc, av_xy.*.nc xz.*.nc, av_xz.*.nc yz.*.nc, av_yz.*.nc	DATA_2D_XY_NETCDF DATA_2D_XY_AV_NETCDF etc.
3D data set	instantaneous or time-av.	3d.*.nc, av_3d.*.nc	DATA_3D_NETCDF DATA_3D_AV_NETCDF
1/2/3D data subsets (masked output)	instantaneous/ time-averaged	masked_M01.*.nc, av_masked_M01.*.nc, masked_M02.*.nc, etc.	DATA_MASK_NETCDF_M01 DATA_MASK_AV_NETCDF_M01 etc.

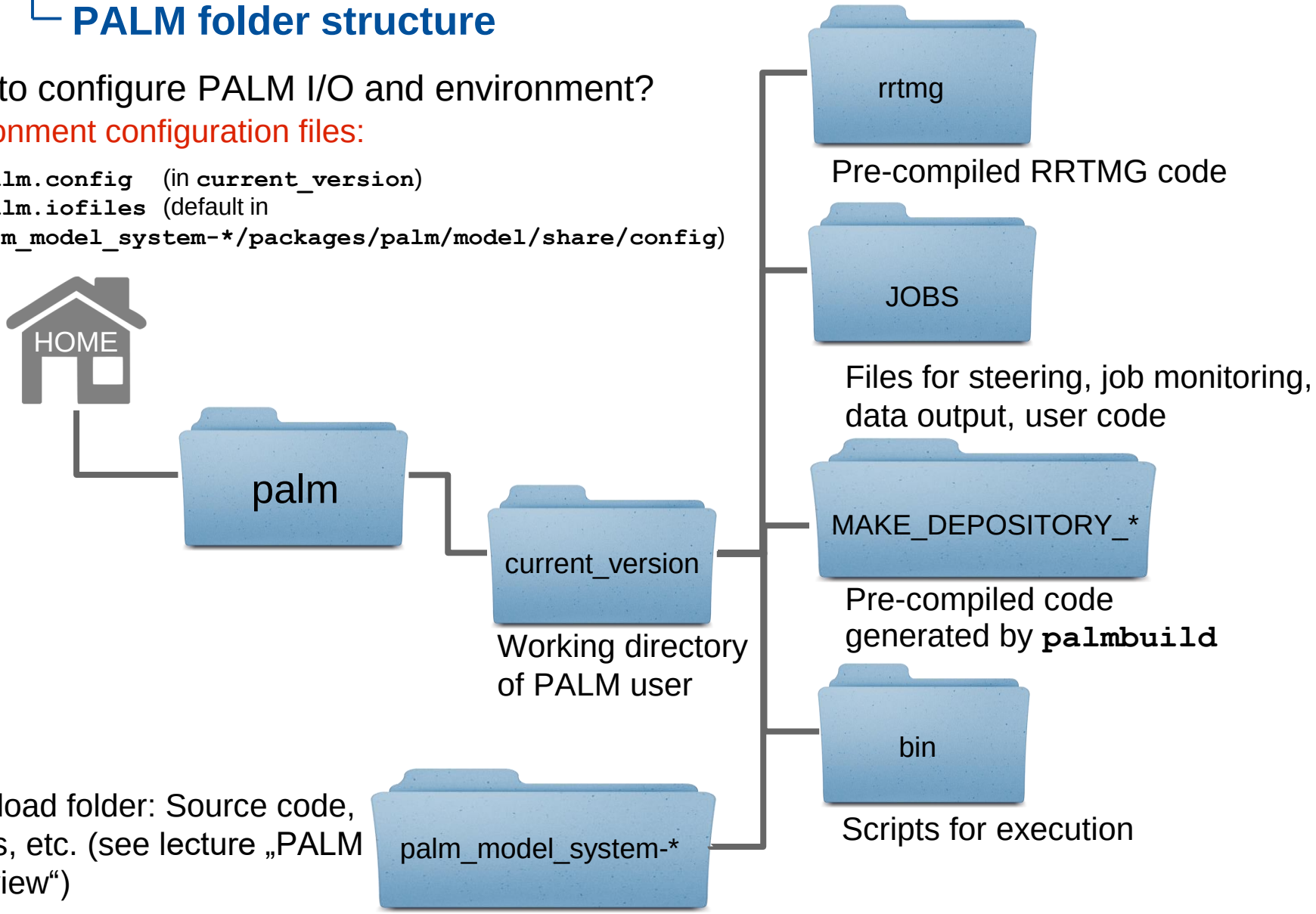
- NetCDF data format: http://www.unidata.ucar.edu/software/netcdf/
- PALM output steering parameters: https://palm.muk.uni-hannover.de/trac/wiki/doc/app/runtime_parameters#Dataoutput
- Hints for plotting NetCDF data will be provided in the lecture „Data visualization“

PALM folder structure

How to configure PALM I/O and environment?

Environment configuration files:

- `.palm.config` (in `current_version`)
- `.palm.iofiles` (default in `palm_model_system-*/packages/palm/model/share/config`)



Configuration of Environment – Overview

Which steps are taken care of to run PALM?

palmbuild

(1) Compile PALM source code
- **MAKE_DEPOSITORY_<configuration_identifier>** (short **<ci>,** typically **<ci>, <ci>** = default) created, containing pre-compiled source code
- Already done, if automatic installer has been used
- **palmbuild** has to be called manually after each code update!

palmrun

(2) Execute PALM
- **PALM is executed in a temporary directory** **<run_identifier>.<randomnumber>** (short **<ri>**), which should reside on a fast file system, given by variable **%fast_io_catalog** in the configuration file (next slide).

(3) Copy/transfer input & output files
- Copy files to **<ri>.<randomnumber>** from input folders, output files to output folders. Cyclic numbering of output files in order to maintain them in case of running the same/similar jobs multiple times. Otherwise, output would be overwritten!

(4) By default, directory **<ri>.<randomnumber>** is deleted at the end.

palmrun calls **palmbuild** (step 1) in case that MAKE_DEPOSITORY_<ci> doesn't exist (new installation) or user-defined code shall be used!

Required information for **palmrun / palmbuild** must be specified by the user in PALM's configuration files (see next slides).

Configuration of Environment – Configuration File

.palm.config.<ci>

```
#
%base_directory      /home/raasch/palm/current_version
%base_data           /home/raasch/palm/current_version/JOBS
%source_path         /home/raasch/palm/current_version/trunk/packages/palm/model/src
%user_source_path    /home/raasch/palm/current_version/JOBS/$run_identifier/USER_CODE
%fast_io_catalog     /localdata/raasch
%restart_data_path   /localdata/raasch
%output_data_path    /home/raasch/palm/current_version/JOBS
%local_jobcatalog    /home/raasch/job_queue
#%remote_jobcatalog   <path/to/directory>
#
%local_ip            127.0.0.1
%local_username      raasch
#
%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
%make_options        -j 4
%compiler_options    -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
#------------------------------------------------------------------------------
#
# Combine 1D- and 3D-profile output (these files are not usable for plotting)
OC:[[ -f LIST_PROFIL_1D     ]]  &&  cat  LIST_PROFIL_1D  >>  LIST_PROFILE
OC:[[ -f LIST_PROFIL        ]]  &&  cat  LIST_PROFIL     >>  LIST_PROFILE
#
# Combine all particle information files
OC:[[ -f PARTICLE_INFOS/_0000 ]]  &&  cat  PARTICLE_INFOS/* >> PARTICLE_INFO
```

- Created during automatic installation.
- Computer- and software-specific configuration.
- Variables are interpreted by **palmrun** script.
- **<ci>** can be an arbitrary string.
- One individually named file per environment in that PALM shall be executed (compiler options, computer, software).
- Documentation available at:

https://palm.muk.uni-hannover.de/trac/wiki/doc/app/palm_config

Configuration of Environment – Configuration File

.palm.config.<ci>

```
#
%base_directory      /home/raasch/palm/current_version
%base_data           /home/raasch/palm/current_version/JOBS
%source_path         /home/raasch/palm/current_version/trunk/packages/palm/model/src
%user_source_path    /home/raasch/palm/current_version/JOBS/$run_identifier/USER_CODE
%fast_io_catalog     /localdata/raasch
%restart_data_path   /localdata/raasch
%output_data_path    /home/raasch/palm/current_version/JOBS
%local_jobcatalog    /home/raasch/job_queue
#%remote_jobcatalog   <path/to/directory>
#
%local_ip            127.0.0.1
%local_username      raasch
#
%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
%make_options        -j 4
%compiler_options    -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
#-------------------------------------------------------------------------------
#
# Combine 1D- and 3D-profile output (these files are not usable for plotting)
OC:[[ -f LIST_PROFIL_1D     ]]  &&  cat  LIST_PROFIL_1D  >>  LIST_PROFILE
OC:[[ -f LIST_PROFIL        ]]  &&  cat  LIST_PROFIL     >>  LIST_PROFILE
#
# Combine all particle information files
OC:[[ -f PARTICLE_INFOS/_0000 ]]  &&  cat  PARTICLE_INFOS/* >> PARTICLE_INFO
```

Header

Path variables

Information about local computer

Compiler names (parallel and serial)

Preprocessor directives

Compiler and linker options, including software library paths

UNIX commands to be executed...

- before PALM code execution
- in case of an error during execution
- after PALM code execution
- **Configuration files must be adapted for batch mode and submission of jobs from a local to a remote computer!**

PALM steering

Configuration of Environment– PALM I/O configuration file

`.palm.iofiles`

```
#
#------------------------------------------------------------------------------------------------
# List of input-files
#------------------------------------------------------------------------------------------------
PARIN                     in:tr      d3#      $base_data/$run_identifier/INPUT          _p3d*
PARIN                     in:tr      d3r      $base_data/$run_identifier/INPUT          _p3dr*
PARIN                     in:tr      pcr      $base_data/$run_identifier/INPUT          _pcr*
NAVIGATION_DATA           inopt:tr   d3#:d3r   $base_data/$run_identifier/INPUT          _nav*
TOPOGRAPHY_DATA           inopt:tr   d3#:d3r   $base_data/$run_identifier/INPUT          _topo*
...
DATA_1D_PR_NETCDF         inopt      prr      $output_data_path/$run_identifier/OUTPUT  _pr*       nc
DATA_1D_TS_NETCDF         inopt      tsr      $output_data_path/$run_identifier/OUTPUT  _ts*       nc
DATA_1D_PTS_NETCDF        inopt      ptsr     $output_data_path/$run_identifier/OUTPUT  _pts*      nc
DATA_2D_XY_NETCDF         inopt      xyr      $output_data_path/$run_identifier/OUTPUT  _xy*       nc
DATA_2D_XY_NETCDF         inopt      xyr      $output_data_path/$run_identifier/OUTPUT  _xy*       nc
DATA_2D_XY_AV_NETCDF      inopt      xyr      $output_data_path/$run_identifier/OUTPUT  _av_xy*    nc
...
#
#------------------------------------------------------------------------------------------------
# List of output-files
#------------------------------------------------------------------------------------------------
BINOUT*                   out:lnpe restart   $restart_data_path/$run_identifier/RESTART  _d3d
PARTICLE_RESTART_DATA_OUT* out:lnpe prt#:prtr $restart_data_path/$run_identifier/RESTART  _rprt
SVFOUT*                   out:lnpe svfout    $restart_data_path/$run_identifier/SVF      _svf
#
RUN_CONTROL*              out:tr   d3#:pcr   $output_data_path/$run_identifier/MONITORING  _rc
RUN_CONTROL*              out:tra  d3r       $output_data_path/$run_identifier/MONITORING  _rc
HEADER*                   out:tr   d3#:pcr   $output_data_path/$run_identifier/MONITORING  _header
HEADER*                   out:tra  d3r       $output_data_path/$run_identifier/MONITORING  _header
CPU_MEASURES*             out:tr   d3#:pcr   $output_data_path/$run_identifier/MONITORING  _cpu
CPU_MEASURES*             out:tra  d3r       $output_data_path/$run_identifier/MONITORING  _cpu
PARTICLE_INFO*            out:tr   pt#       $output_data_path/$run_identifier/MONITORING  _prt_info
PARTICLE_INFO*            out:tra  ptr       $output_data_path/$run_identifier/MONITORING  _prt_info
LIST_PROFIL*              out:tr   *         $output_data_path/$run_identifier/MONITORING  _list_pr
#
DATA_1D_PR_NETCDF*        out:tr   *         $output_data_path/$run_identifier/OUTPUT  _pr        nc
DATA_1D_SP_NETCDF*        out:tr   *         $output_data_path/$run_identifier/OUTPUT  _sp        nc
DATA_1D_TS_NETCDF*        out:tr   *         $output_data_path/$run_identifier/OUTPUT  _ts        nc
DATA_1D_PTS_NETCDF        out:tr   *         $output_data_path/$run_identifier/OUTPUT  _pts       nc
DATA_2D_XY_NETCDF*        out:tr   *         $output_data_path/$run_identifier/OUTPUT  _xy        nc
DATA_2D_XY_AV_NETCDF*     out:tr   *         $output_data_path/$run_identifier/OUTPUT  _av_xy     nc
...
```

Documentation available at: https://palm.muk.uni-hannover.de/trac/wiki/doc/app/palm_iofiles

- A default file in folder **`palm_model_system-*/packages/palm/model/share/config`** is used (contains a maximum of six columns)
- Definition of input/output files and paths, interpreted by **`palmrun`** script (column 1, 4-6)
- **Activation strings** (column 3) are given with **`palmrun`** option **`-a`**
- Only one **`.palm.iofiles`** file required. Path variable **`base_data`** can be individually set in **`.palm.config.<ci>`**, and **`run_identifier`** equals the run-identifier name given with **`palmrun`** option **`-r`** (next slide)

Starting a run – palmrun options

- **`palmrun ...`**
- Important options for interactive run on local PC:
  - **`-a`** `"d3#"`
    (activation string, possible strings listed in **`.palm.iofiles`**)
  - **`-r`** <run_identifier>
    (e.g. run_x, taken as base file/directory name of all data files of this run)
  - **`-c`** <configuration_identifier>
    (as in configuration file name **`.palm.config.<ci>`**)
  - **`-X`** <number of processor cores (PEs) to be used>
- For batch mode and remote jobs, additional options are available, see palmrun documentation:
  - **`-b`** (run in batch mode)
  - **`-q`** <name of batch-job queue>
  - **`-t`** <CPU time (in seconds)>
  - **`-T`** <number of mpi tasks to be started on one node of the computer>

`palmrun ?` lists all available options

Starting a run – palmrun execution

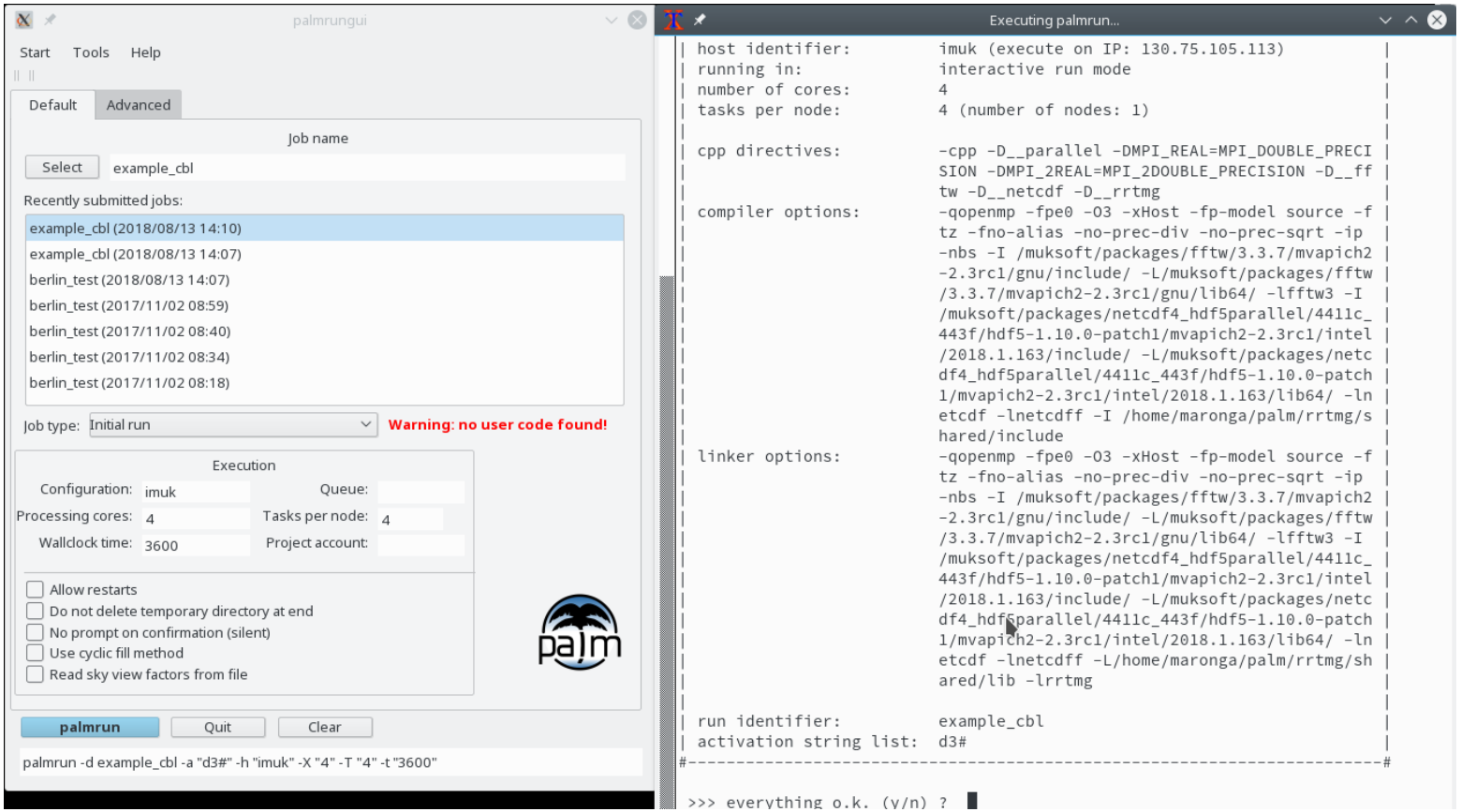
Directly type **`palmrun`** command-line into UNIX terminal

```
(base) sebastian@sebastian-ThinkPad-T495s:~/palm/current_version$ palmrun -r example_cbl -c default -a "d3#" -X 4

*** palmrun
    will be executed.     Please wait ...

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

  *** INFORMATIVE: additional source code directory
      "/home/sebastian/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                                  Fr 21. Jul 14:36:16 CEST 2023 |
| Version: PALM 23.04-rc.1                                               |
|                                                                        |
| called on:               sebastian-ThinkPad-T495s                      |
| config. identifier:      default (execute on IP: 127.0.0.1)            |
| running in:              interactive run mode                          |
| number of cores:         4                                             |
| tasks per node:          4 (number of nodes: 1)                        |
|                                                                        |
| cpp directives:          -cpp -D__gfortran -D__parallel -DMPI_REAL=MPI |
|                          _DOUBLE_PRECISION -DMPI_2REAL=MPI_2DOUBLE_PRE |
|                          CISION -D__netcdf -D__fftw -D__rrtmg          |
| compiler options:        -Ofast -ffree-line-length-none -I /usr/includ |
|                          e -I /usr/include -I /home/sebastian/palm/cur |
|                          rent_version/rrtmg/include                    |
| linker options:          -Ofast -ffree-line-length-none /usr/lib/x86_6 |
|                          4-linux-gnu/libnetcdff.so /usr/lib/x86_64-lin |
|                          ux-gnu/libfftw3.so /home/sebastian/palm/curre |
|                          nt_version/rrtmg/lib/librrtmg.so              |
|                                                                        |
| run identifier:          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/palmrungui)
- Invoke **`palmrungui`** in UNIX terminal

```
sno:~/palm/current_version$ palmrungui
```



Summary of palmrun's workflow

Minimum command for executing PALM with more than one core:

```
palmrun -r <run_identifier> -a <activation_string>
        -X <total number of cores>
```

example: **palmrun -r *run_x* -a "*d3#*" -X *4* [-c *default*]**

1. Checks, if source code has been pre-compiled, i.e. does directory **MAKE_DEPOSITORY_default** exist? If not, **palmrun** calls the **palmbuild** script to compile the code according to settings in **.palm.config.default**
2. Temporary working directory **run_x.<randomnumber>** under path given by **%fast_io_catalog** in **.palm.config.default** is created
3. INPUT files and **MAKE_DEPOSITORY_default** content are copied to temporary directory. For every manual execution of **palmrun -r run_x** …, a folder **SOURCES_FOR_RUN_run_x** is created under path **%fast_io_catalog**. It contains the compiled code, **palmrun** script, configuration files, etc.
4. PALM is executed in the temporary directory
5. MONITORING and OUTPUT files are copied from temporary directory to their destinations given in **.palm.iofiles**

```
#------------------------------------------------------------------------------
# List of output-files
#------------------------------------------------------------------------------
RUN_CONTROL*         out:tr    d3#:pcr    $base_data/$run_identifier/MONITORING  _rc
...
DATA_1D_TS_NETCDF*   out:tr          *    $base_data/$run_identifier/OUTPUT      _ts     nc
```

run_x_ts.<cyclenumber>.nc
where <cyclenumber> is a 3 digit number

Final filename (e.g.): ~/palm/current_version/JOBS/run_x/OUTPUT/run_x_ts.000.nc

6. Delete temporary directory (prevented by **palmrun** option **"-B"**).
   Folder **SOURCES_FOR_RUN_run_x** is not deleted, remains available for restart runs.

Monitoring of a run

(1) Watch the informative messages and progress bar in terminal (interactive run only).

```
palmrun -d run_x -a "d3#" -h <ci> -X 4 -T 4
...
...
*** execution starts in directory
    "/home/sebastian/palm/current_version/tmp/example_cbl.14917"
--------------------------------------------------------------------------------
*** running on: mypc mypc mypc mypc
               ... reading environment parameters from ENVPAR --- finished
               ... reading NAMELIST parameters from PARIN --- finished
               ... creating virtual PE grids + MPI derived data types --- finished
               ... checking parameters --- finished
               ... allocating arrays --- finished
               ... initializing with constant profiles --- finished
               ... initializing statistics , boundary conditions , etc. --- finished
               ... creating initial disturbances --- finished
               ... calling pressure solver --- finished
               --- leaving init_3d_model
               --- start with time-stepping
[XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX______________________________] 53.3 left
```

(2) Watch file RUN_CONTROL file in the temporary working directory:

- switch to temporary directory of the specific run
- command **`tail -f RUN_CONTROL`** get's you the consecutive PALM output into this file on the terminal

```
host:...$ cd <path_to_temp_directory>
host:/path_to_temp_directory$ tail -f RUN_CONTROL
0       2 00:00:40.00   20.0000A  -0.2117  -0.2252   0.1162  0.009  0.69  -0.132E+02    100.  0.102E-02  0.102E-02  0.939E-05
0.413E-15    7   7  11   11   7  19    5  26   7    0.000    0.000       0
0       3 00:01:00.00   20.0000A  -0.2106  -0.2227   0.1152  0.009  0.69  -0.132E+02    100.  0.101E-02  0.101E-02  0.131E-04
0.417E-15    7   7  11   11   7  19    5  26   7    0.000    0.000       0
```

PALM online:
https://palm.muk.uni-hannover.de

Our YouTube channel:
https://youtube.com/user/palmhannover