Installation of OpenPALM on a Linux station of the formation room at CERFACS

By The OpenPALM Team†

1. Download of the latest stable OpenPALM distribution

The latest stable release of OpenPALM is downloadable from the OpenPALM web site (Fig. 1): http://www.cerfacs.fr/globc/PALM_WEB/
The instructions to download OpenPALM are accessible from the page *Become a user*. In order to inform the OpenPALM Team of your download, you have to fill a form (Fig. 2).

2. Introduction

In the OpenPALM distribution you'll find the source codes of the OpenPALM library, of its interface and of all the sessions of the training. The first thing to do you have to decompress the gzipped tar archive of the distribution:

```
> tar -xvfz distrib.tgz
```

Two directories are created: PrePALM_MP and PALM_MP. The first one contains the graphical user interface PrePALM, the second one the OpenPALM library. An archive that contains CWIPI is also available (cwipi_*version*.tgz).

The libraries that composed OpenPALM have to be installed: PALM and CWIPI.

3. Installation of the PrePALM graphical user interface

3.1. *Pre-requirements*

The graphical interface PrePALM is written in Tcl/Tk with some C. Therefore you need these two environments on the machines where PrePALM has to run. The Tcl/Tk version has to be at least 8.3

A small C program is used to interpret the STEPLANG language: it is therefore necessary to compile this component. A pre-compiled version working on i386 to i686 and x86_64 platforms is provided with the OpenPALM distribution. The most widespread public domain algebra libraries (such as BLAS, LAPACK, ScaLAPACK) interfaced in the OpenPALM algebra toolbox are not provided with the OpenPALM distribution and should be installed (if they are not already pre-installed) on the machines where the final application has to be compiled and executed. On the contrary, the geophysical interpolation library based on the OASIS coupler and on the SCRIP algorithms is provided with the OpenPALM distribution.

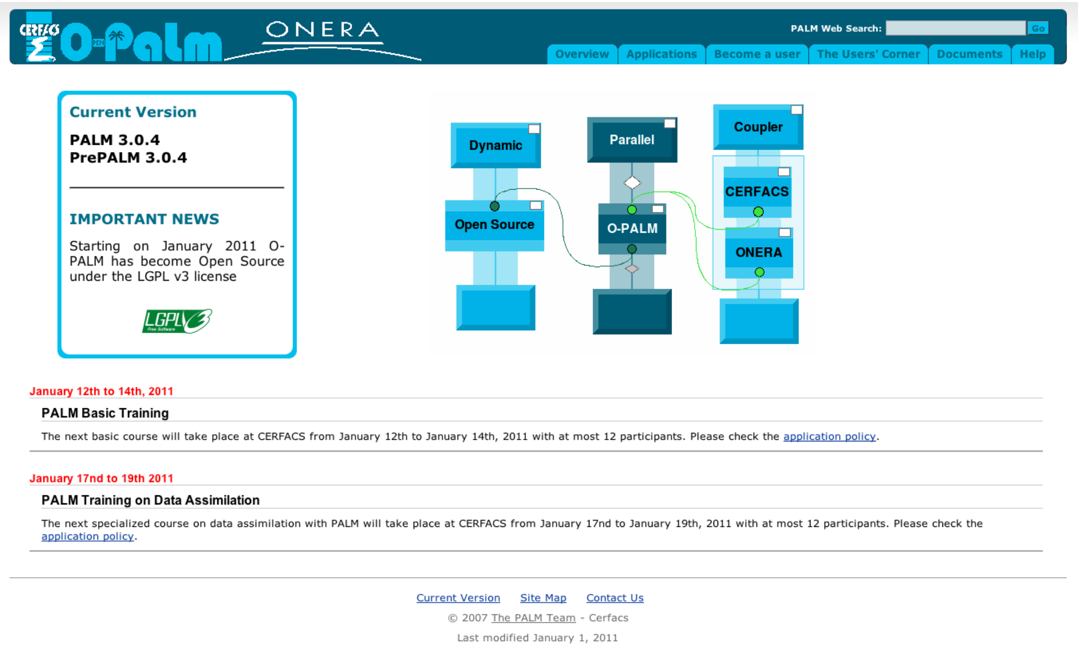
3.2. *PrePALM command definition*

The graphical user interface is written in Tcl/Tk which is an interpreted language. Therefore there is no need of compilation. Nevertheless every user has to set an environment variable containing the installation path and an alias as a shortcut for the GUI. Accordingly to the preferred shell you should add to the .cshrc or .bashrc or .rc file:

csh, tcsh:

```
> setenv PREPALMMPDIR path_to_PrePALM
> alias prepalm $PREPALMMPDIR/prepalm_MP.tcl \!* &
```

† CERFACS, 42 avenue G. Coriolis, 31 057 Toulouse Cedex 01, France



Current Version

PALM 3.0.4
PrePALM 3.0.4

IMPORTANT NEWS

Starting on January 2011 O-PALM has become Open Source under the LGPL v3 license

January 12th to 14th, 2011

PALM Basic Training

The next basic course will take place at CERFACS from January 12th to January 14th, 2011 with at most 12 participants. Please check the application policy.

January 17nd to 19th 2011

PALM Training on Data Assimilation

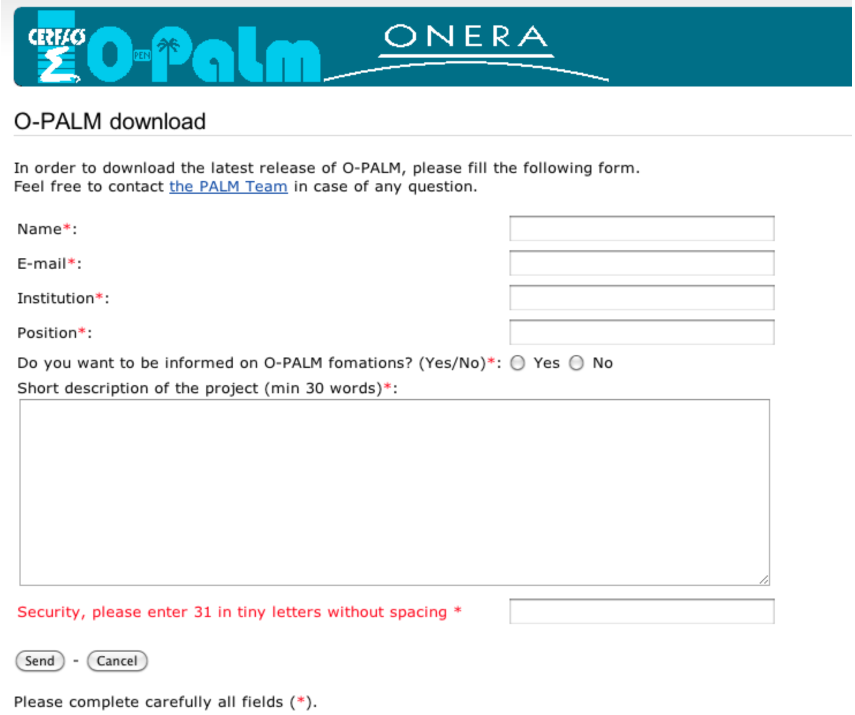
The next specialized course on data assimilation with PALM will take place at CERFACS from January 17nd to January 19th, 2011 with at most 12 participants. Please check the application policy.

Current Version Site Map Contact Us

© 2007 The PALM Team - Cerfacs

Last modified January 1, 2011

FIGURE 1. OpenPALM web site.



O-PALM download

In order to download the latest release of O-PALM, please fill the following form.
Feel free to contact the PALM Team in case of any question.

Name*:

E-mail*:

Institution*:

Position*:

Do you want to be informed on O-PALM fomations? (Yes/No)*: ○ Yes ○ No

Short description of the project (min 30 words)*:

Security, please enter 31 in tiny letters without spacing *

Please complete carefully all fields (*).

FIGURE 2. OpenPALM download page. Anti-spam = *thirtyone*.

sh, bash:

```
function prepalm {
export PREPALMMPDIR=path_to_PrePALM
$PREPALMMPDIR/prepalm_MP.tcl $* &
}
```

Optionally you can set the PREPALMEDITOR environment variable pointing to your preferred editor that PrePALM will start every time it proposes to access an external editor. The default is vi. If you rather prefer emacs you should add to the shell configuration file:

csh, tcsh:

```
> setenv PREPALMEDITOR emacs
```

sh, bash:

```
> export PREPALMEDITOR=emacs
```

3.3. *STEPLANG interpreter installation*

Steplang is the command language used to describe the event driven actions manipulating the objects stored in the buffer. If you need to recompile its interpreter, enter STEPLANG the directory

```
> cd PrePALM_MP/STEPLANG/
```

Modify, if needed, the simple Makefile

```
CC=gcc
ARCH=i386
```

and issue:

```
> make clean
> make
```

If everything go right, you should end up with the steplang-i386 executable.

4. Installation of the PALM library

4.1. *Pre-requirements*

The OpenPALM library includes the objects used to generate the OpenPALM driver (palm_main) and the user defined entities (units and blocks). This library has to compiled on the platform where the PALM application will eventually run. The installation procedure is based on the automatic configuration tool autoconf. Remember that OpenPALM has been implemented in FORTRAN 90 and C.

To install PALM it is therefore necessary to have access to:

- A FORTRAN 90 and a C compiler. They have to be compatible. The best idea is to use the two compilers from a same distribution and at the same version
- An MPI library that implements the MPI-2 standard (one does not need MPI-2 if he is only going to work in MPI-1 mode. Cf. Chapter 15). The MPI library has to compiled with the same compiler as in the previous item.

Optionally, depending on the OpenPALM features you are going to use, you may need

- the standard scientific libraries BLAS and LAPACK (possibly optimised by the manufacturer).
- the parallel algebra libraries PBLAS and SCALAPACK
- the NetCDF I/O library
- the sources of the minimisers of which the interface is available in the PALM algebra toolbox

Remark: you do not need superuser rights to install OpenPALM on your machine.

4.2. *Installation*

You install PALM with simply three commands from the PALM_MP directory of the distribution:

```
> ./configure [OPTION]... [VAR=VALUE]...
> make
> make install
```

The only step requiring some attention is the first one, for you have to choose the proper options for the configuration. They depend on your compilers, on the platform, on the MPI distribution and, finally, on the flavor of OpenPALM (single proc, MPI-1, MPI-2) that you are going to install.

You can obtain a summary of the available options with the command ./configure help that will answer: ./configure [OPTION]... [VAR=VALUE]...

To assign environment variables (e.g., CC, CFLAGS...), specify them as VAR=VALUE. See below for descriptions of some of the useful variables. Defaults for the options are specified in brackets.

```
Configuration:
  -h, --help                display this help and exit
      --help=short          display options specific to this package
      --help=recursive      display the short help of all the included packages
  -V, --version             display version information and exit
  -q, --quiet, --silent     do not print `checking...' messages
      --cache-file=FILE     cache test results in FILE [disabled]
  -C, --config-cache        alias for `--cache-file=config.cache'
  -n, --no-create           do not create output files
      --srcdir=DIR          find the sources in DIR [configure dir or `..']

Installation directories:
  --prefix=PREFIX           install architecture-independent files in PREFIX
                            [NONE]
  --exec-prefix=EPREFIX     install architecture-dependent files in EPREFIX
                            [PREFIX]

By default, `make install' will install all the files in
`NONE/bin', `NONE/lib' etc. You can specify
an installation prefix other than `NONE' using `--prefix',
for instance `--prefix=$HOME'.

For better control, use the options below.

Fine tuning of the installation directories:
  --bindir=DIR             user executables [EPREFIX/bin]
  --sbindir=DIR            system admin executables [EPREFIX/sbin]
  --libexecdir=DIR         program executables [EPREFIX/libexec]
  --datadir=DIR            read-only architecture-independent data [PREFIX/share]
  --sysconfdir=DIR         read-only single-machine data [PREFIX/etc]
  --sharedstatedir=DIR     modifiable architecture-independent data [PREFIX/com]
  --localstatedir=DIR      modifiable single-machine data [PREFIX/var]
  --libdir=DIR             object code libraries [EPREFIX/lib]
  --includedir=DIR         C header files [PREFIX/include]
  --oldincludedir=DIR      C header files for non-gcc [/usr/include]
  --infodir=DIR            info documentation [PREFIX/info]
  --mandir=DIR             man documentation [PREFIX/man]

System types:
  --build=BUILD      configure for building on BUILD [guessed]
  --host=HOST        cross-compile to build programs to run on HOST [BUILD]

Optional Features:
  --disable-FEATURE        do not include FEATURE (same as --enable-FEATURE=no)
  --enable-FEATURE[=ARG]   include FEATURE [ARG=yes]
  --enable-64bits          Use 64 bits addressing (default on sgi and fujitsu)
  --enable-promote-real    Promote REAL fortran data type to DOUBLE PRECISION
  --enable-blasopti        Use BLAS optimization (default on scalar computers)
  --enable-mpi_softwait    Use non CPU hogging mpi_wait (default on sgi, sun, nec, linux)

Optional Packages:
  --with-PACKAGE[=ARG]     use PACKAGE [ARG=yes]
  --without-PACKAGE        do not use PACKAGE (same as --with-PACKAGE=no)
  --without-mpi            Use Monoprocessing without MPI
  --with-mpich=MPICH_ROOT mpich for MPI (default=no)
  --with-lam=LAMMPI_ROOT  lam for MPI (default=no)
  --with-openmpi=OPENMPI_ROOT  OpenMPI for MPI (default=no)
  --with-mpi_path=path     Path of the MPI implementation
  --with-F90=F90           F90 compiler
  --with-CC=CC             C compiler
  --with-fopt=OPT          Option for Fortran Compiler
  --with-copt=OPT          Options for C compiler
  --with-debug=EXTRA_FLAGS     enable debugging (default debug flag  is -g)
  --with-fortran_underscore    Underscore at end of fortran functions
  --with-fortran_main=MAIN     internal name of main FORTRAN routine
                               (default value depends on system type)
  --with-roundtrip-delay=roundtrip-delay        *100 MPI_Iprobes (default~100)
  --with-mpi_comm_free=mpi_comm
  --with-leak_mem_ctl       To detect memory leak
  --with-shared_lib       Compile shared libraries
  --with-mpi1mode           using mpi1 mode (no spawn)
  --with-mpi2win              using mpi2 windows

Some influential environment variables:
  CC          C compiler command
  CFLAGS      C compiler flags
  LDFLAGS     linker flags, e.g. -L          if you have libraries in a
              nonstandard directory
  CPPFLAGS    C/C++ preprocessor flags, e.g. -I             if you have
              headers in a nonstandard directory
  CPP         C preprocessor
```

Use these variables to override the choices made by 'configure' or to help it to find libraries and programs with nonstandard names/locations.

For normal usage, you have to concentrate on the bold blue options only. The remaining options are dedicated to the OpenPALM developers. In any case we suggest to explicitly choose the FORTRAN 90 and C compiler.

It is absolutely mandatory that the C and FORTRAN compilers are compatible and to use them for compiling (in this given order):

- the MPI library
- the PALM library
- the object libraries for the PALM units
- the PALM applications.

4.3. *MPI-1 installation*

For the installation of OpenPALM in the formation room, you will use the MPICH2 MPI distribution that is already installed on the stations (in the path /usr/lib64/mpich2). The option *–with-mpi1mode* allow to choose the MPI-1 mode of OpenPALM. Use the following command for the configure:

```
> ./configure --with-mpich=/usr/lib64/mpich2 --with-F90=mpif90 --with-CC=mpicc
--with-mpi1mode --enable-64bits --with-fortran\_main=main -with-copt=-DCWIPI
```

Once the configuration step is done (with no error: you can check the *config.log* file), you can compile the OpenPALM sources and *install* the library:

```
> make
> make install
```

The configure step allows the automatic determination of the installation folder depending on the chosen options. You can verify that the OpenPALM libraries are installed in *linux64r4mpich_mpi1mode/*.

4.4. *MPI-2 installation*

During the formation, you have used OpenPALM in the MPI-2 mode. This mode is the native mode of the coupler. To install the library in this mode, you have first to clean the previous compilation:

```
make clean
```

The configure command is almost the same that the one used for the MPI-1 mode (replace *with-mpi1mode* by *with-shared_lib*):

```
> /configure --with-mpich=/usr/lib64/mpich2 --with-F90=mpif90 --with-CC=mpicc
--enable-64bits --with-fortran_main=main  --with-shared_lib
```

Then compile and install:

```
> make
> make install
```

The MPI-2 libraries are installed in the folder *linux64r4mpich/*. Note that both versions of OpenPALM (MPI-1 and MPI-2) are now available. I encourage you to rerun your last session with *your version of OpenPALM* !!!!

5. Installation of the CWIPI library

The installation of the CWIPI library must be done with the same compilers and MPI distribution as the PALM library. In order to facilitate the use of both library, we recommend to install the CWIPI library in the same directory as the PALM library. Note that CWIPI is, for the moment, only accessible with the MPI-1 mode of PALM.

The installation of CWIPI is similar to the installation of PALM with the use of *configure*, *make* and *make install*. In the *configure* command, the variable *$INSTALL_PALM* correspond to the directory where PALM is installed.

```
> ./configure CC=mpicc CXX=mpicxx FC=mpif90 --disable-python-bindings
--prefix=${INSTALL_PALM}/linux64r4mpich_mpi1mode/
--exec-prefix=${INSTALL_PALM}/linux64r4mpich_mpi1mode/
--enable-shared=no --host=x86_64-unknown-linux
```

Then compile and install the library:

```
> make
> make install
```