Installing TELEMAC V6P0 on Unix, Linux and Windows

F. Decung, C. Denis, J.-M. Hervouet

Version 6.0

---

updated on 6 July 2010

This note sums up the installation procedure of Telemac on Unix, Linux and Windows operating systems.

1) Prerequisites

A Fortran 90 compiler is requested. A large number have already been tested such as gfortran, Intel, pgi, HP, Nag on Linux and Intel, gfortran, g95 on Windows.

The only mandatory extra library requested for running Telemac is Perl. You may use the version already existing on your system if any, or download Perl from the link given on our Website.

Other libraries may be useful but are optional:

For parallelism: Mpi and Metis are requested. They must be installed and compiled. The compiled library of Métis called libmetis.a must be put into the parallel library (see below).

Mumps: sequential or parallel direct solver, especially for Artemis.

Tecplot add on (to include a Telemac data loader in Tecplot if you have it). So far valid for Tecplot 360 2006.

2) Directories in the installation of TELEMAC

Once copied or untarred, your Telemac directory (called /home/user/TELEMAC in our examples) should contain the following sub-directories, some of them being optional as indicated below.

|--	bin	Perl scripts : system, tools, compiling, run…
|--	config	Definition of the environement : version, compiler, $projet There may be several of this kind, the variable $SYSTELCFG will point to them
|--	bief	Finite elements library
|--	damocles	Library for reading the keywords in the parameter files
|--	paravoid	A dummy library when there is no parallelism
|--	parallel	The interface to parallel languages. At the time being calls MPI Including libmetis.a
|--	special	System specific Fortran subroutines
|--	spartacus2d	2D Smooth Particle Hydrodynamics (not on website)
|--	spartacus3d	3D Smooth Particle Hydrodynamics (not on website)
|--	tomawac	Generation of waves by wind
|--	artemis	Agitation in harbours
|--	sisyphe	Sediment transport
|--	telemac2d	Saint-Venant equations
|--	telemac3d	Navier-Stokes equations (not on website)
|--	estel2d	Groundwater flows (not on website)
|--	estel3d	
|--	postel3d	Post-processing for cross-sections in Telemac-3D, optional
|--	stbtel	Software for converting various geometry file formats, optional
|--	mpi	MPI library for parallelism, may be installed here, optional
|--	mumpsmpi	Interface subroutines calling the parallel MUMPS library, optional
|--	mumpsvoid	Ghost subroutines in order to compile TELEMAC systel without installing MUMPS

3) Configuring the system:

A (small) number of environment variables must be set in your user profile. Depending on the shell script used this will be added to different system files, such as .bashrc for bash shell. Namely you have to modify the variable $PATH and to create and export a new variable $SYSTELCFG.

- PATH

The “bin” directory of the Telemac directory, which contains a number of executable commands, must be added, as indicated in the example below:

```
> export PATH=/home/user/TELEMAC/bin:$PATH
```

NB : most scripts are written in perl language. Check that perl is duly installed on your machine. If not there is one version in the Telemac installation, as shown above, that should be then installed. In this case the path to this perl library must be added to the $PATH variable.

- SYSTELCFG

The variable $SYSTELCFG gives the path to the directory configlinux which contains the configuration file systel.ini:

```
> export SYSTELCFG=/home/user/TELEMAC/configlinux
```

There may be several such directories, each one dedicated to a specific version of Telemac or compiler. One can also edit the systel.ini file to skip from one version to the other or to skip from one machine to the other (see next paragraph).

- PARALLELISM

If parallelism is requested, the paths to the MPI library must be given also:

```
> export PATH=/home/user/TELEMAC/mpi/intel_10/bin:$PATH
>export
LD_LIBRARY_PATH=/home/user/TELEMAC/mpi/intel_10/lib:$LD_LIBRARY_PATH
```

When all this is done, you can check the following list of commands:

```
> . .bashrc : must return the text of file .bashrc
> type cfgmak : must return the path of this command
> echo $SYSTELCFG : must return the path of the configlinux directory
> type mpirun : must return the path to this command
```

4) Configuring the file $SYSTELCFG/systel.ini :

The core of the configuration lies in the systel.ini file situated at the address given by the variable $SYSTELCFG. This file is used to build others and modifications must be followed by the perl command: cfgmak (which is in /home/user/TELEMAC/bin). It updates the cfgmak.mak which is in the same directory as the file systel.ini.

File systel.ini is formed of 4 sections:

1. [GENERAL] : for choosing version numbers and language.
2. [INSTALLATION] : for choosing the installation directory (variable PROJECT) and the type of machine (HOSTTYPE). Data on this machine are given in section 4.
3. [PERL] : paths to perl
4. [HOSTTYPE] : definition of compiler and linker options that will be used by all the Fortran software.

Section per section, here are the variables to be modified:

1. [GENERAL]
   - LNGTEL : choosing the language for telemac-2D (1 – French ; 2 – English)
   - VERSTEL : choosing the version number for telemac-2D (must be the same, it can be different only for provisional developments)
   - LNGTEL3D : language for Telemac-3D
   - VERSTEL3D : version for Telemac-3D
   - Etc. for every program

2. [INSTALLATION]
   - PROJECT : path to the installation directory

```
> PROJECT = /home/user/TELEMAC
```

   - HOSTTYPE : type of machine, or couple machine-compiler (two different hosttypes may

be defined for a single machine, to take into account two different compilers). This hosttype must then appear in section 4.

```
> HOSTTYPE = intel_10
```

3. [PERL]

```
> PERLPATH = /usr/bin
> PERL5LIB = /usr/lib
```

or any other address of the perl library.

4. [HOSTTYPE]

- We now detail the data that are necessary for every couple machine-compiler:

```
> [intel_10]
```

- DIRLIB : name of the directory where the libraries will be stored (there will be one per program and it will be put into the program directory, so it is only here a generic name, preferably equal to the HOSTTYPE for simplicity).

```
DIRLIB = intel_10
```

- Compiler data:

```
FC_NAM = “ifort” : compiler command (e.g. f90)
FC_OPT_OBJTEXT = “o” : extension for object files
FC_OPT_COMPIL = “ –c –O2 –convert big_endian ” : compiler options
FC_OPT_DEBUG = “ –c –O0 –C –g –convert big_endian” : compiler option with debugging
FC_OPT_PROFILE = “ –c –O2 –pg –convert big_endian” : compiler options with profiling
FC_OPT_INCLUDE = “-I” : compiler command for including files
FC_OPT_OTHERS =  “ ” : for extra options
LK_NAM="ifort" : name of linker
LK_OPT_NORMAL=" -lf2c -lm -lz -lstdc++ " : linker options
LK_OPT_OUTNAME=" -o " : extension for object files
LK_OPT_DEBUG=" -g90 " : option for debugging
LK_OPT_PROFILE=" -pg " : option for profiling
LK_OPT_OTHERS=" " : other options
# extra libraries, here example of libraries MED and HDF5
```

```
LK_LIB_SPECIAL=" /home/MED/libmed.a /home/MED/libhdf5.a "
#
LIB_NAM=ar
LIB_OPT_LIBEXT="a" : extension for static archives
LIB_OPT_OUTNAME="cru"
LIB_OPT_OTHERS=
LIB_RANLIB="ranlib"
#
RUN_DEBUG="dbx90 " : command for debugging
RUN_PROFILE=          : command for profiling
```

- Data related to parallelism (example of mpich in telemac distribution):

```
FC_MPI="<TELEMAC_HOME>/mpi/<DIRLIB>/bin/mpif90  "  :  mpi  wrapper  for
Fortran 90
LK_MPI="<TELEMAC_HOME>/mpi/<DIRLIB>/bin/mpif90 -o <EXE>  <OBJS>  <LIBS> "
: name of mpi linker.
LIBS_MPI="-L  <TELEMAC_HOME>/mpi/<DIRLIB>/lib  –lmpich  -lf2c  -lm  -lz  -
lstdc++ " : linker options for mpi linker.
RUN_MPI="<TELEMAC_HOME>/mpi/<DIRLIB>/bin/mpirun -machinefile mpirun.txt -
np <N> <EXE>” : command for running mpi
```

Important notes:

The TELEMAC system is often compiled with the MPICH library. The “-lmpich” or “-lmpi” flags are not working with all flavours of MPI (Ex: *`mpif90 –showme:link`* to show the flags necessary to link TELEMAC with OPENMPI library - http://www.open-mpi.org/faq/?category=mpi-apps).

If you do not use mpif90 there is also a dependency on MPI in the makefile of library “parallel”, where the include directory of MPI must be specified with the variable INCMPI.

5) Installing Mumps:

This is necessary only for running Artemis in parallel and in this case paragraph 2 is relevant. Otherwise follow paragraph 1.

ARTEMIS has to solve sparse linear systems with matrices which are not positive-definite. Iterative methods may fail to solve them. Consequently, ARTEMIS uses direct methods issued

from:

- the YSMP package for the ARTEMIS sequential version;
- the external MUMPS library for the ARTEMIS parallel version.

The user has to install the parallel version of MUMPS to be able to use the ARTEMIS parallel version. More information about the MUMPS installation can be found here: http://graal.enslyon.fr/MUMPS/. In the next TELEMAC version (6.1), it will be possible to solve sparse linear systems in sequential and parallel for all programmes in the TELEMAC system.

In order to compile successfully all programmes included the TELEMAC system, the user has to modify the *systel.ini* configuration file depending on his choice about the installation or not of the MUMPS library.

1. Configuring the systel.ini without installing the parallel MUMPS library

In this case, the user has not previously installed the parallel MUMPS library. The *mumpsvoid* library contains ghost subroutines used to compile the *bief* library without MUMPS. If the user chooses the MUMPS solver (SOLVER=9) in this configuration, the program will fail by indicating that the MUMPS library has not been installed.

Suppose that the TELEMAC root directory is called PROJECT.

The user has to fill LIBMUMPSPAR and LIBMUMPSSEQ in the systel.ini file as follows :

```
LIBMUMPSPAR=<PROJECT>mumpsvoid/mumpsvoid_v6p0/<HOSTTYPE>/mumpsvoidv6p0
.a
LIBMUMPSEQ=<PROJECT>mumpsvoid/mumpsvoid_v6p0/<HOSTTYPE>/mumpsvoidv6p0.
a
```

Example:

Suppose that the local configuration is set as follows:

```
PROJECT=/home/user/TEST_ARTEMIS/TELEMAC_V6P0/
DIRLIB=intel_10
```

The user has to fill the systel.ini with the following

```
LIBMUMPSMPI="/home/user/TEST_ARTEMIS/TELEMAC_V6P0/mumpsvoid/mumpsvoid_v6p
0/intel_10/mumpsvoidv6p0.a"

LIBMUMPSSEQ="/home/user/TEST_ARTEMIS/TELEMAC_V6P0/mumpsvoid/mumpsvoid_v6p
0/intel_10/mumpsvoidv6p0.a"
```

2. Configuring the systel.ini with the real parallel MUMPS library

In this case, the user has previously installed the parallel MUMPS library. The *mumpsmpi* library

contains MUMPS interface subroutines.

Suppose that:

- the TELEMAC root directory is called PROJECT
- the parallel MUMPS root directory is called ROOTMUMPS
- the parallel MUMPS include directory is called INCMUMPS

In the parallel MUMPS makefile, the location of the SCALAPACK and METIS libraries are put in the SCALAP and LMETIS variables. These location is also needed for the compilation of TELEMAC.

The user has to fill LIBMUMPSPAR and LIBMUMPSSEQ in the systel.ini as follows:

```
LIBMUMPSPAR=<PROJECT>/mumps/mumpsmpi/mumpsmpi_v6p0/<HOSSTYPE>/mumpsmpiv6p
0.a <SCALAP> <LMETIS> -lm
LIBMUMPSSEQ=<PROJECT>/mumps/mumpsvoid/mumpsvoid_v6p0/<HOSSTYPE>/mumpsvoid
v6p0.a
INCDIRMUMPS=INCMUMPS
```

Example:

Suppose that the local configuration is set as follows:

```
PROJECT=/home/user/TEST_ARTEMIS/TELEMAC_V6P0/
DIRLIB=intel_10
ROOTMUMPS= /home/user/MUMPS_4.9.0/
INCMUMPS=/home/user/MUMPS_4.9.0/include/
```

There is an extract of the MUMPS makefile showing the setting of METIS and SCALAP:

```
SCALAP  = -L/home/logiciels/intel/mkl/10.2.1.017/lib/em64t -
lmkl_scalapack_lp64 -lmkl_blacs_ilp64 -lmkl_intel_lp64 -lmkl_sequential
-lmkl_core -lguide -static-intel

LMETIS = /home/logiciels/metis-4.0/lib/libmetis.a
```

The user has to fill the systel.ini with the following:

```
INCDIRMUMPS = /home/user/MUMPS_4.9.0/include

LIBMUMPSPAR="/home/user/TEST_ARTEMIS/TELEMAC_V6P0/mumps/mumpsmpi/mumpsmpi
_v6p0/intel_10/mumpsmpiv6p0.a -L/home/user/MUMPS_4.9.0/lib/ -ldmumps
-lmumps_common -lpord  -L/home/logiciels/intel/mkl/10.2.1.017/lib/em64t
-lmkl_scalapack_lp64 -lmkl_blacs_ilp64 -lmkl_intel_lp64 -lmkl_sequential
-lmkl_core -lguide -static-intel /home/logiciels/metis-4.0/lib/libmetis.a
-lm"

LIBMUMPSSEQ=/home/user/TEST_ARTEMIS/TELEMAC_V6P0/mumps/mumpsvoid/mumpsvoi
```

```
d_v6p0/intel_10/mumpsvoidv6p0.a
```

The user must not forget to run the command *cfgmak* in order to take into account the modification performed on the *systel.ini* file (this is a common error).

```
> cfgmak : command after changing systel.ini
```

6) Compiling the whole TELEMAC system:

There are two methods for compiling TELEMAC in normal mode (not to speak of debugging and profiling): a global command called makeall90, and a step by step command that must be launched into every directory of sources, in the right order.

Note 1: when compiling the library PARALLEL, the METIS library « libmetis.a » must already exist in the directory « DIRLIB » (cf. systel.ini), METIS must then be compiled before, see the link to Métis on our Website.

Note 2: every programme directory contains a directory sources with the Fortran 90 sources and makefiles.

makefile is meant for Unix and Linux.

makefile.wnt is used for Intel compiler and Digital Fortran on Windows

makefile.gfo is used for g95 and gfortran compilers on Windows. **For using these compilers edit the file maktel.bat in directory bin and change the line as indicated.**

Note 3: at this level your configuration file systel.ini must be OK, and command cfgmak done (see above)

6.1 recompiling all the system

- makeall90: this command compiles all libraries, modules, scalar executables and put them in relevant directories.

```
> makeall90
```

This command is a perl file in the bin directory, and may be updated in makeall90.pl.

An extra command to be run after is makepar90, which does the same thing but with parallel executables.

```
> makepar90
```

You can also use the command makeallclean for erasing all previous compiled files in the directories containing the sources, e.g. when recompiling for a new compiler or machine.

6.2 step by step compiling

- maktel : for a step by step compiling, must be executed iin every Fortran source directory, it will actually execute the relevant makefile. The procedure is the following:

```
> maktel menage: will erase all objects, archives, modules and executables
> maktel install: compiling sources and installing modules, libraries and executables in the « DIRLIB » directory (cf. systel.ini)
> maktel parallel: same thing but for generating a default parallel executable (used in parallel when there is no extra user Fortran)
```

When compiling step by step, the following order is requested:

1. mumpsvoid
2. mumpsmpi (optional)
3. damocles
4. bief
5. paravoid
6. parallel (optional)
7. special

then all the modules in the system, but with sisyphe before telemac2d and telemac2d before telemac3d (when it is available).

Eventually if necessary:

8. stbtel

9. postel3d

7) Tecplot 360 2006 AddOns on Linux/MacIntosh platforms *(optional)*:

A TELEMAC data loader has been developed to generate plots from TELEMAC results with the TECPLOT software. This program is not distributed with the Tecplot software. General information on the installation procedure or on the Add-On Developers Kit (ADK) can be found on the Tecplot website. (http://www.tecplottalk.com/addons.php & http://www.tecplot.com/Support/Documentation.aspx)

The TELEMAC data loader can be used with TELEMAC or DELWAQ results file (in case of coupling)

In preparation, you must perform the following steps (procedure adapted from http://www.tecplot.com/Support/Documentation.aspx):

1. Install Tecplot
2. Verify that the *$TEC_360_2010* environment variable is assigned to the directory where Tecplot has been installed ($TEC360HOME or $TECHOME with older version)
3. Verify that the *PATH* environment variable includes the following:

   $TEC_360_2010/bin:$TEC_360_2010/adk/bin

4. Create the Add-On Development Root Directory
5. Set the environment variable *TECADDONDEVDIR* to the Add-On Development Root Directory
6. Set the environment variable *TECADDONDIR* to the Add-On lib Root Directory
7. Set the environment variable *TECADDONDEPLATFORM* to one of the valid platform names. A list of valid platforms can be obtained by running *tec360 –platlist* (Ex: *linux64.26* on a 64-Bits Linux 2.6 system)
8. Once the preceding steps are completed, copy the *TELEMAC data loader* (directory located at *bin/AddOns/Tecplot_Addons/seralo)* in *$TECADDONDEVDIR*
9. Compile the TELEMAC data loader with the commande Runmake (choose your platform and the release option)
10. Copy the compiled library *libseralo.so* to the lib sub-directory:

```
> cp libseralo.so $TECADDONDIR/$TECADDONPLATFORM/
```

11. To enable the add-on, the user must create a new file *tecdev.add* with the following line:

```
#!MC 1120
$!LoadAddOn "|$TECADDONDIR|/|$TECADDONPLATFORM|/libseralo"
```

Note that the 1120 value depends on the Tecplot version (11.2 for instance)

12. Set the environment variable *TECADDONFILE* to the Add-On file (*tecdev.add*)

Other add-ons may be added as well with the $*TECADDONFILE variable.*

8) Running a case:

You are ready for running a case, e.g. using the test cases accompanying every programme, either by using a user interface or directly with the command:

```
> telemac2d steering_file
```

This command (here example of the programme telemac2d with a parameter file called steering_file) must be done in the directory containing the parameter file. It will create a temporary folder and the listing showing details of the execution will be displayed on the screen.

Type telemac2d –h for more details on this command and see the user manuals of the different programmes to have details on the key-words that may be used in the parameter files.

Note: with some compilers, when specific Fortran sources are used in the file called FORTRAN FILE, it is necessary to have also in this file the main programme, which for every programme, is called homere_name_of_programme.f, e.g. homere_telemac2d.f. Otherwise the linker will ask for the main programme.
