
NEMO validation tools Documentation

Release 1.0.1

C. Levy

20120330

Contents

1	Shell scripts	1
1.1	all_functions.sh	1
1.2	fcm_job.sh	2
1.3	prepare_exe_dir.sh	3
1.4	prepare_job.sh	3
1.5	sette.sh	4
1.6	sette_beginner.sh	6

Shell scripts

1.1 all_functions.sh

1.1.1 Set of functions used by sette.sh (NEMO tests)

SYNOPSIS

```
$ ./set_namelist INPUT_NAMELIST VARIABLE VALUE
$ post_test_tidyup
```

DESCRIPTION

function superegrep input variable value

function set_namelist input namelist_name variable value output namelist

function post_test_tidyup creates nemo_validation tree, and save output & debug files this function creates tree of validation in NEMO_VALIDATION_DIR as follows :

NEMO_VALIDATION_DIR/WCONFIG_NAME/WCOMPILER_NAME/REVISION_NUMBER(or DATE)/TEST_NAME

NEMO_VALIDATION_DIR : is choosen in param.cfg

WCONFIG_NAME : set by makenemo at the moment of compilation

WCOMPILER_NAME : set by makenemo at the moment of compilation

REVISION_NUMBER(or DATE) : revision number by svn info, if problems with svn date is taken

TEST_NAME : set in sette.sh for each configuration to be tested (directory TEST_NAME is created under \${NEW_CONF} directory)

EXAMPLES

```
$ ./set_namelist namelist          nn_itend          75
$ ./set_namelist namelist_ice_lim2 cn_icerst_in  \"00101231_restart_ice\"
$ post_test_tidyup
```

TODO

option debug

EVOLUTIONS

\$Id: all_functions.sh 3294 2012-01-28 16:44:18Z rblod \$

- creation

1.2 fcm_job.sh

1.2.1 Simple job for NEMO tests

SYNOPSIS

:: launches the script \$JOB_FILE interactive or batch, one task or MPI

```
$ ./fcm_job.sh NUMBER_OF_PROCS JOB_FILE INTERACT MPI_FLAG
```

DESCRIPTION

Simple job for SET TESTS for NEMO (SETTE)

EXAMPLES

```
$ ./fcm_job.sh NUMBER_OF_PROCS JOB_FILE INTERACT MPI_FLAG
```

run a

TODO

option debug

EVOLUTIONS

\$Id: fcm_job.sh 3294 2012-01-28 16:44:18Z rblod \$

- creation

1.3 prepare_exe_dir.sh

1.3.1 Set of functions used by sette.sh (NEMO tests)

SYNOPSIS

```
$ ./prepare_exe_dir.sh
```

DESCRIPTION

prepare_exe_dir.sh creates execution directory takes name of TEST_NAME defined in every test in sette.sh

it is necessary to define in sette.sh TEST_NAME (example : export TEST_NAME="LONG") to create execution directory in where run test.

NOTE : each test has to run in its own directory (of execution), if not existing files are re-written (for example namelist)

EXAMPLES

```
$ ./prepare_exe_dir.sh
```

TODO

option debug

EVOLUTIONS

\$Id: \$

- creation

1.4 prepare_job.sh

1.4.1 create the job script for NEMO tests

Title overline too short.

```
-----
create the job script for NEMO tests
-----
```

SYNOPSIS

```
$ ./prepare_job.sh INPUT_FILE_CONFIG_NAME NUMBER_PROC TEST_NAME MPI_FLAG JOB_FILE
```

DESCRIPTION

Part of the SETTE package to run tests for NEMO
prepare the script \$JOB_FILE to run the tests

EXAMPLES

```
$ ./prepare_job.sh INPUT_FILE_CONFIG_NAME NUMBER_PROC TEST_NAME MPI_FLAG $JOB_FILE
```

prepare the \$JOB_FILE for execution

TODO

option debug

EVOLUTIONS

\$Id: prepare_job.sh 3050 2011-11-07 14:11:34Z acc \$

- creation

1.5 sette.sh

1.5.1 Set of tests for NEMO

SYNOPSIS

```
$ ./sette.sh
```

DESCRIPTION

Variables to be checked by user:

COMPILER : name of compiler as defined in NEMOGCM/ARCH directory

BATCH_COMMAND : name of the command for batch submission

MPI_INTERACT :

for MPP tests, “no” for batch execution, “yes” for interactive execution

NOTE: for run with 1 process tests are run always in MPI_INTERACT=”yes”

Principal script is sette.sh, that calls

makenemo

creates the executable in \${CONFIG_NAME}/BLD/bin/nemo.exe (and its link opa in \${CONFIG_NAME}/EXP00)

param.cfg : sets and loads following directories:

FORCING_DIR : is the directory for forcing files (tarfile)

INPUT_DIR : is the directory for input files storing

TMPDIR : is the temporary directory (if needed)

NEMO_VALIDATION_DIR : is the validation directory

(NOTE: this file is the same for all configurations to be tested with sette)

all_functions.sh : loads functions used by sette (note: new functions can be added here)

set_namelist : function declared in all_functions that set namelist parameters for tests

post_test_tidyup : creates validation storage directory and copy needed output files (solver.stat and ocean.output) in it after execution of test.

Tree of VALIDATION is:

NEMO_VALIDATION_DIR/WCONFIG_NAME/WCOMPILER_NAME/TEST_NAME/REVISION_NUMBER(or DATE)

prepare_exe_dir.sh : defines and creates directory where the test is executed

execution directory takes name of TEST_NAME defined in every test in sette.sh

(each test in executed in its own directory)

prepare_job.sh

to generate the script run_job.sh

fcm_job.sh

runs job in interactive or batch mode : all jobs using 1 process are run interactive, and all MPP jobs are

run in batch (MPI_INTERACT="no") or interactive (MPI_INTERACT="yes") see sette.sh and BATCH_TEMPLATE directory

(note this job needs to have an input_CONFIG.cfg in which can be found input tar file)

NOTE: if job is not launched for some problems you have executable ready in \${EXE_DIR} directory

NOTE: the changed namelists are leaved in \${EXE_DIR} directory whereas original namelist remains in \${NEW_CONF}/EXP00

in \${SETTE_DIR} is created output.sette with the echo of executed commands

if sette.sh is stopped in output.sette there is written the last command executed by sette.sh

if you run: ./sette.sh 2>&1 | tee out.sette

in \${SETTE_DIR} out.sette is redirected standard error & standard output

EXAMPLES

```
$ ./sette.sh
```

TODO

option debug

EVOLUTIONS

\$Id: sette.sh 3294 2012-01-28 16:44:18Z rblod \$

- creation

1.6 sette_beginner.sh

1.6.1 Set of tests for NEMO for beginners

SYNOPSIS

```
$ ./sette_beginner.sh
```

DESCRIPTION

First simple example of how to use SETTE: create GYRE_SHORT configuration, compile it with 1 proc, and test it for a SHORT test: 5days