

# Building software for ROSACE end-to-end Case Study

## Step-by-step instructions

In order to use the ROSACE Case Study you need to follow this step-by-step instructions. Alternatively you may try to use the CMake script provided here:

[https://svn.onera.fr/schedmcore/branches/ROSACE\\_Case\\_Study/instructions/ROSACE-CaseStudy-auto.cmake](https://svn.onera.fr/schedmcore/branches/ROSACE_Case_Study/instructions/ROSACE-CaseStudy-auto.cmake) which could be run as `cmake -P ROSACE-CaseStudy-auto.cmake` (The script only handles **native build**, see below). You may read the comments at the top of this CMake script file in order to have more information about its usage.

These instructions have been tested on a Linux 32 or 64 bit hosts.

1. You should have CMake (<http://www.cmake.org>) installed on your machine
2. Create some directories which will contain all necessary materials
  - a) `mkdir ROSACE-step-by-step` the base directory
  - b) `mkdir ROSACE-step-by-step/Download` directory for downloaded material
  - c) `mkdir ROSACE-step-by-step/installed` directory for local installation
3. Install Prelude compiler (you'll have to chose the appropriate version in there: <http://www.lifl.fr/~forget/prelude.html> could be 32 or 64 bits versions, you can even build it from source if needed.). I) Installation from source
  1. `cd ROSACE-step-by-step`
  2. `svn co https://svn.onera.fr/Prelude/Prelude/tags/prelude-1.5`
  3. `cd prelude-1.5; sh ./distrib.sh`
  4. `cd ../installed` (go in local installation directory)
  5. `tar zxvf ../prelude-1.5/prelude-1.5-linux-64.tgz`
1. Installation from pre-compiled archive
  - a) `cd ROSACE-step-by-step/Download` (go in Download dir)
  - b) `wget https://forge.onera.fr/attachments/download/821/prelude-1.5-linux-64.tgz`
  - c) `cd ../installed` (go in local installation directory)
  - d) `tar zxvf ../Download/prelude-1.5-linux-64.tgz`
4. Install Lustre compiler We use the lustre compiler from ONERA : <https://cavale.enseeiht.fr/redmine/projects/lustrec> I) Installation from source
  1. `cd ROSACE-step-by-step/Download`
  2. `wget https://cavale.enseeiht.fr/redmine/attachments/download/99/lustrec-1.1-Xia-Tai-Kang-src.tgz`
  3. `cd ..; tar zxvf Download/lustrec-1.1-Xia-Tai-Kang-src.tgz`
  4. `cd lustrec-1.1-Xia-Qi`
  5. `./configure --prefix=`pwd`/../installed`
  6. `make; make install`
1. Installation from pre-compiled archive  
Not Available [yet]
5. Checkout/compile and install SchedMCore v0.5 or greater (trunk will do as well).
  - a) `cd ..` (go back to base directory)

- b) `svn co https://svn.onera.fr/schedmcore/tags/SchedMCore-0.5.0 schedmcore`
- c) `mkdir build-schedmcore; cd build-schedmcore`
- d) `cmake -DCMAKE_INSTALL_PREFIX=../installed -DENABLE_PRELUDE_SUPPORT=ON ../schedmcore`
- e) `make install`

## 6. Compile the RTAS case study for native execution

1. `cd ..` (go back to base directory)
2. `svn co https://svn.onera.fr/schedmcore/branches/ROSACE_CaseStudy`
3. **Build the prelude\_c implementation**
  1. `mkdir build-ROSACE_CaseStudy-prelude_c-native`
  2. `cd build-ROSACE_CaseStudy-prelude_c-native`
  3. `cmake -DCMAKE_INSTALL_PREFIX=../installed ../ROSACE_CaseStudy/prelude_implementation/prelude_c`
  4. `make` (no need to install)
  5. `../installed/bin/lsmc_converter -l libassemblage-noencoding.so`
4. **Build the prelude\_lustre implementation**
  1. `mkdir build-ROSACE_CaseStudy-prelude_lustre-native`
  2. `cd build-ROSACE_CaseStudy-prelude_lustre-native`
  3. `cmake -DCMAKE_INSTALL_PREFIX=../installed ../ROSACE_CaseStudy/prelude_implementation/prelude_lustre`
  4. `make` (no need to install)
  5. `../installed/bin/lsmc_converter -l libassemblage-noencoding.so`

With the 'native' build one can execute the testcase with SchedMCore functional simulation mode (lsmc\_run-nort tool).

One can reproduce the (native) execution trace of the paper using the following command:

```
lsmc_run-nort -l ./libassemblage-noencoding.so -c 1
-b 1000 -m 20000 -v 0 -s fp
```

or with the functional simulator:

```
lsmc_sim -l ./libassemblage-noencoding.so -m 20000
-v0
```

The native build includes `NativeRun-assemblage-vXX` which can be launch like this

```
make NativeRun-assemblage-v1
```

**respectively:**

```
make NativeSim-assemblage-v1
```

there is two target for each assemblage options defined in the case study. The native run produces a `lsmc_run-nort-results-assemblage_vXX.txt` file which is a CSV self-documented file whereas

native simulation produces a `lsmc_sim-results-  
assemblage_vXX.txt`.

For more informations see:

`ROSACE_CaseStudy/prelude_implementation/prelude_c/README.prelude_c`

For more informations see:

`ROSACE_CaseStudy/prelude_implementation/prelude_lustre/README.prelude_lustre` For more informations about SchedMCore tools see:

<http://sites.onera.fr/schedmcore/sites/sites.onera.fr.schedmcore/files/schedmcore-user-manual.pdf> <http://sites.onera.fr/schedmcore>