LAPP
Laboratoire d'Annecy de Physique des Particules

Create a library : Asterics HPC

Pierre Aubert

Where to get the minimal examples ?

Minimal repository :
`https://lappweb.in2p3.fr/~paubert/ASTERICS_HPC/ressource/build/Correction/ExampleMinimal.tar.gz`

Correction :
`https://lappweb.in2p3.fr/~paubert/ASTERICS_HPC/ressource/build/Correction/Examples.tar.gz`

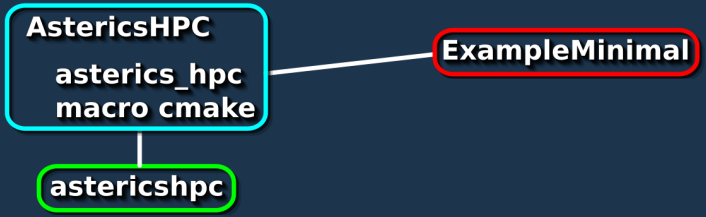
Minimal example

Minimal example

ExampleMinimal

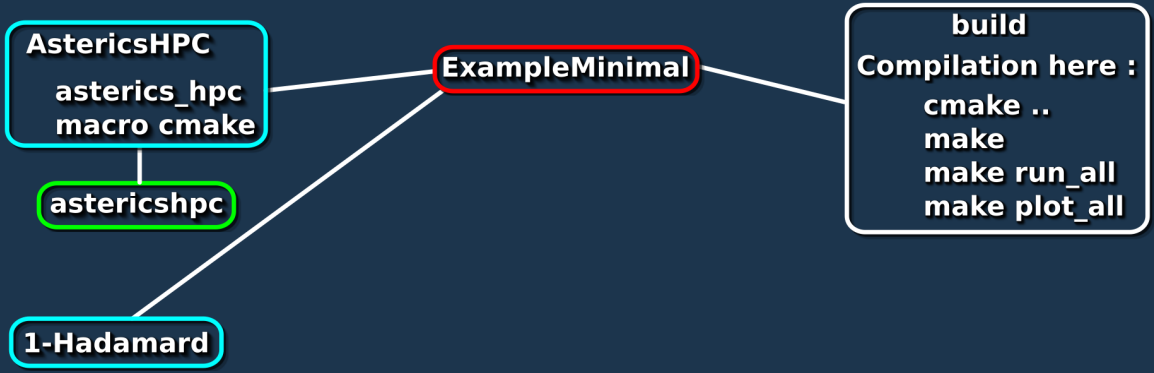
Minimal example

Minimal example

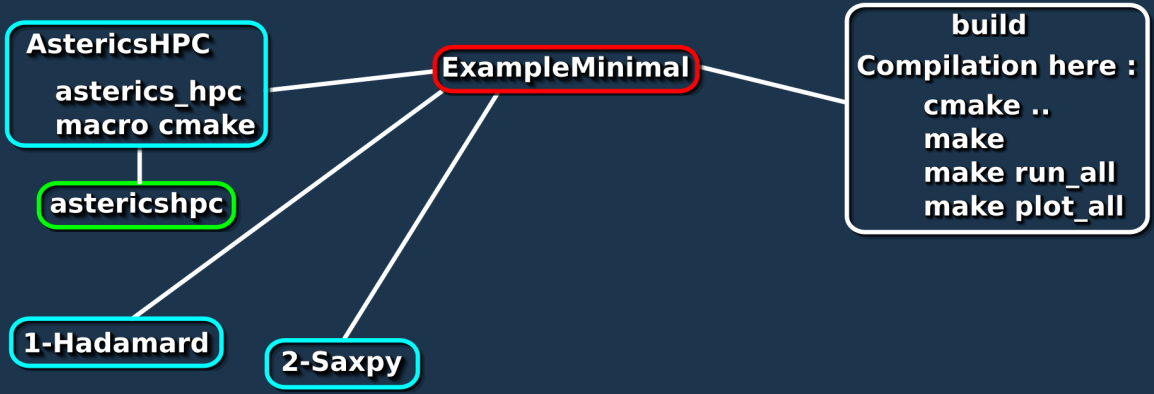


Minimal example

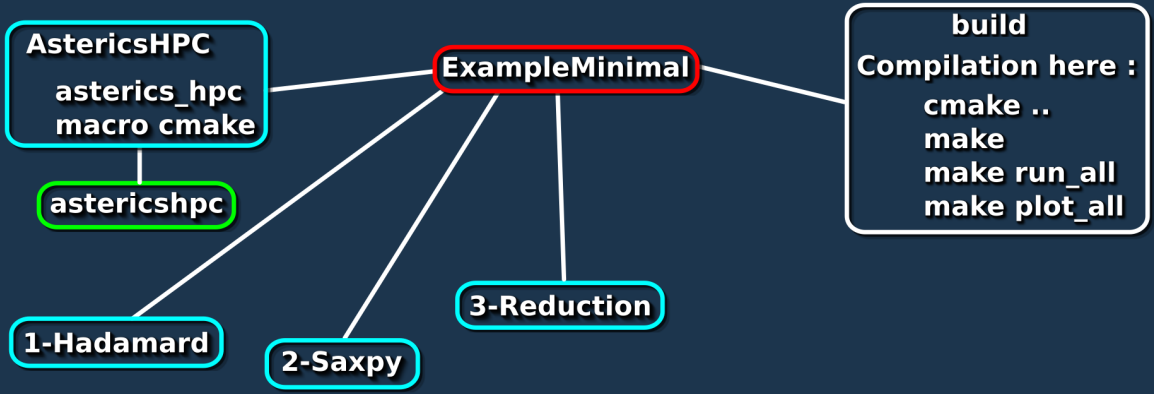
Minimal example



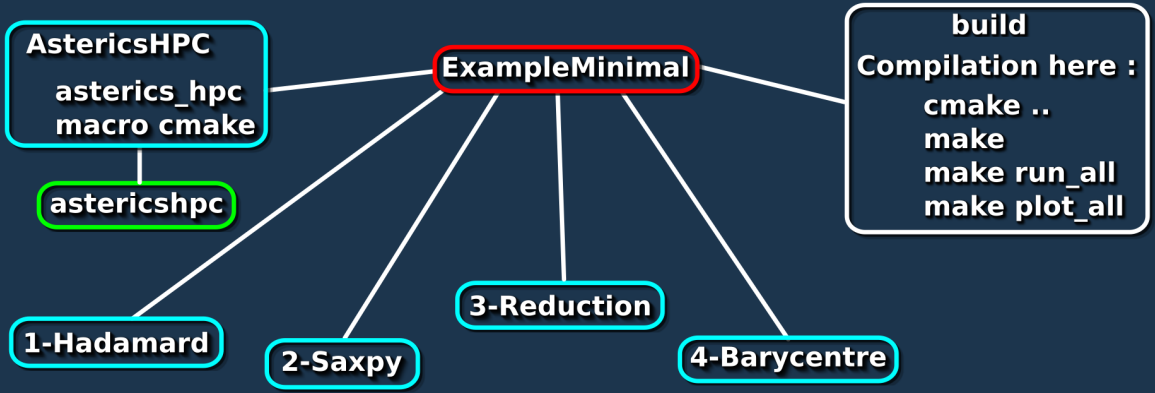
Minimal example



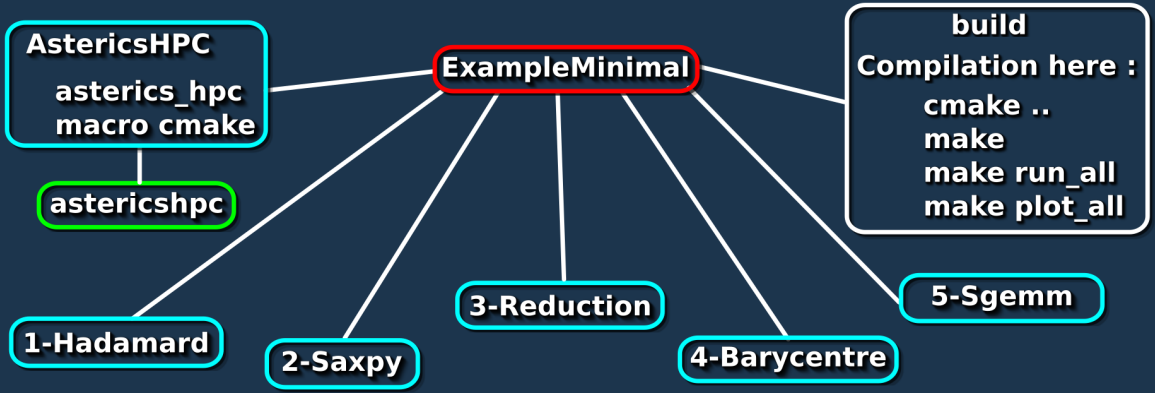
Minimal example



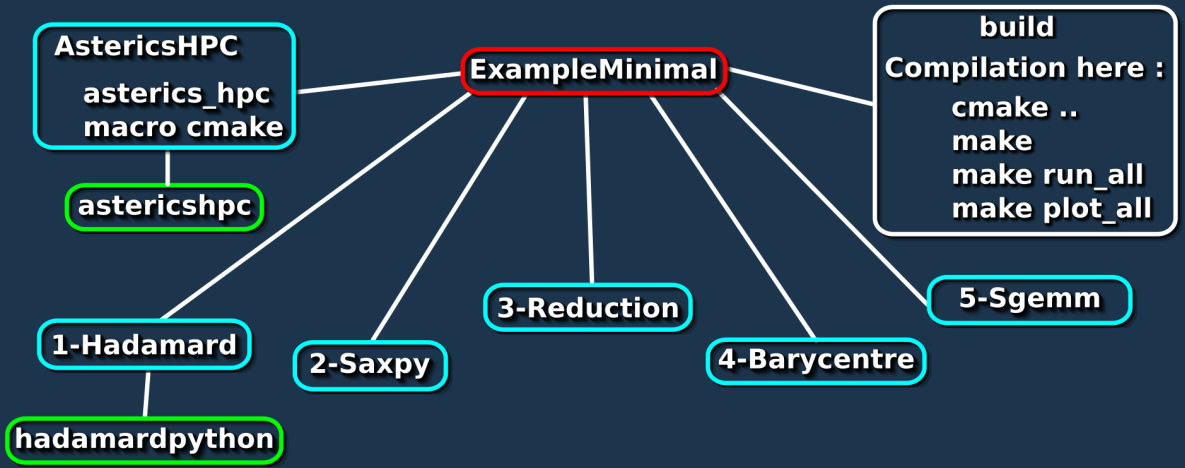
Minimal example



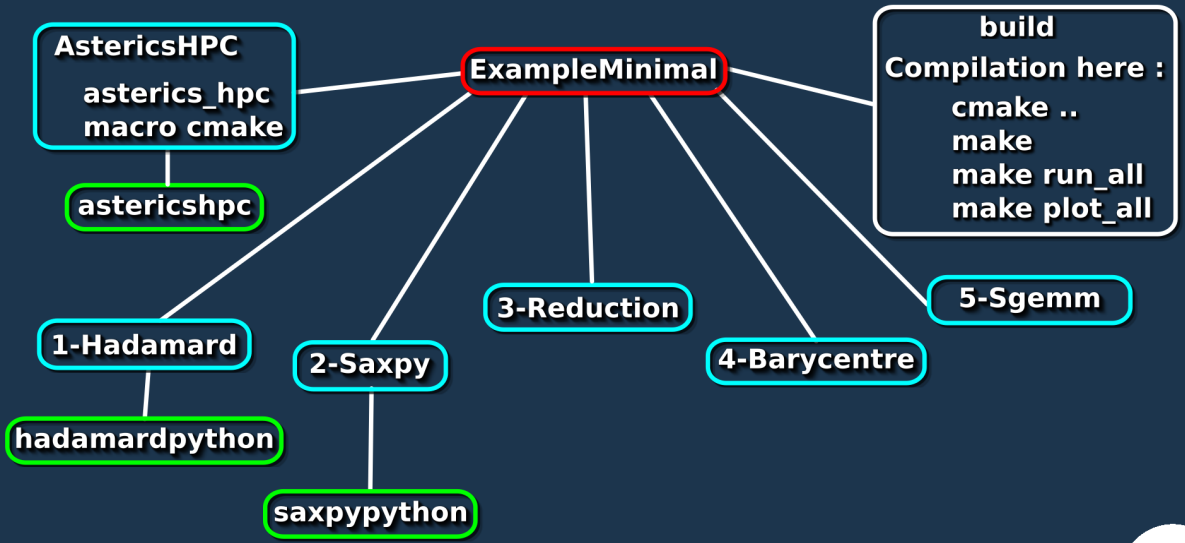
Minimal example



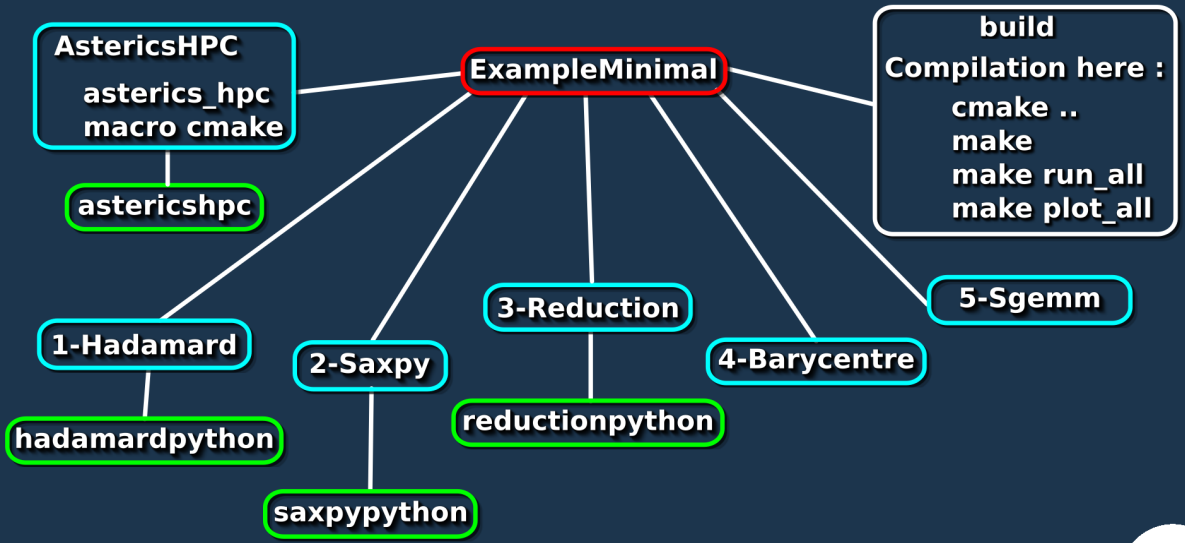
Minimal example



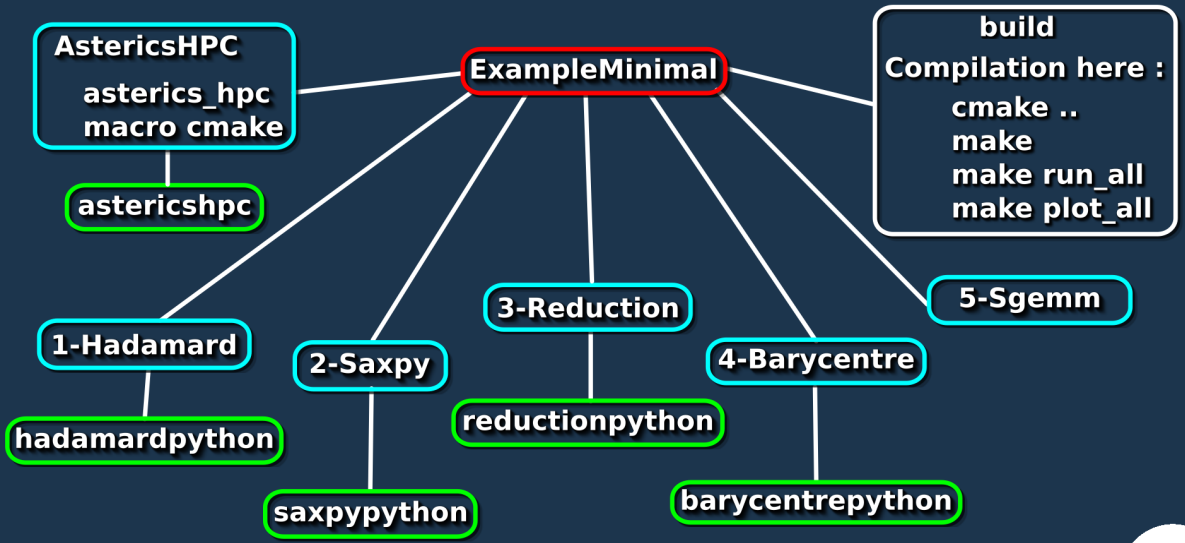
Minimal example



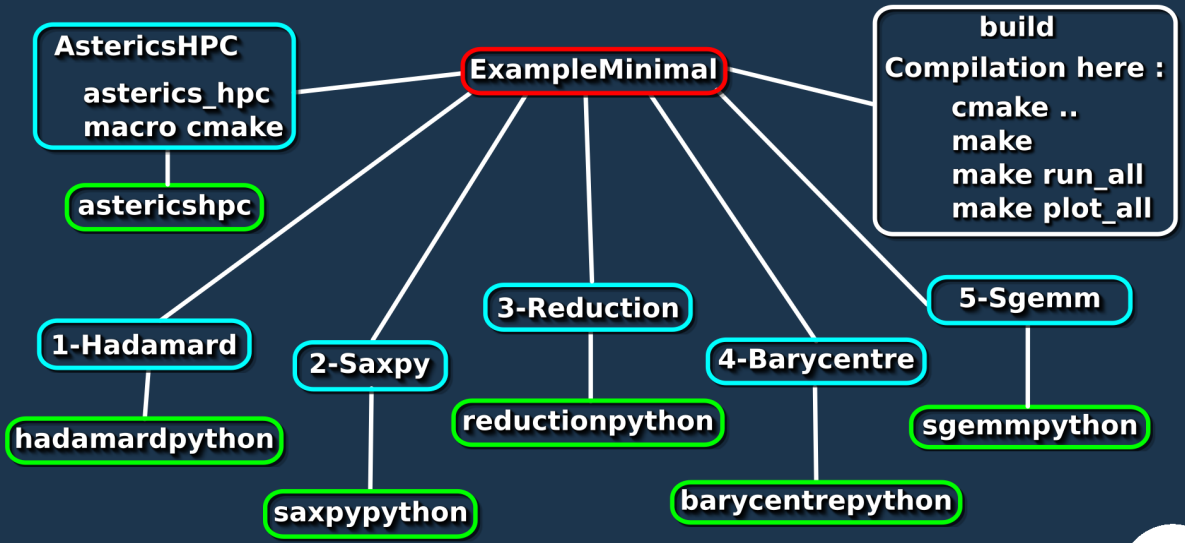
Minimal example



Minimal example



Minimal example



Aims of Asterics HPC library

- To provide :
  - The **rdtsc** function (to time functions)
  - The aligned allocation/deallocation functions (needed for optimisation)
    - Table
    - Matrix
  - Some **CMake** macros to run and plot all the results automatically
    - **runExample(target)** and **runPythonExample(target dependency)** :
      To run executables with **make run_all**
    - **plotPerf("plotName" target1 target2 ...)** :
      To plot and compare results from different targets with **make plot_all**
  - The results are created in **build/Examples/Performances**
- C++ library : **asterics_hpc**
- Python module : **astericshpc**

This will simplify all the following examples.

The ReaD Time Stamp Counter (rdtsc)

Ask the CPU the number of cycles since the program's begining

64 bits version :

```
extern long unsigned int rdtsc(void) {
»       long unsigned int a, d;
»       __asm__ volatile ("rdtsc" : "=a" (a), "=d" (d));
»       return (d<<32) | a;
}
```