



# Standalone analyses packages development in FCCAnalyses

12.08.2022  
(updated 14.08.2022)

[laurent.forthomme@cern.ch](mailto:laurent.forthomme@cern.ch)

# Case studies analysis packages



To ease interaction with various groups developing performance studies (e.g. [HEP-FCC/FCCeePhysicsPerformance](#)), development of a “standalone” analysis package skeleton to be filled by analysts.

Current WIP implementation: [HEP-FCC/FCCAnalyses#199](#). General idea:

- Provide a dedicated directory (`case-studies`) scanned at FCCAnalyses compile time through a set of CMake directives
- If a package (= directory with `src` + `include` subdirs, possibly with a `scripts` analysis configuration folder) is found, build a shared library linked against FCCAnalyses
- Base analysis content:



```
case-studies/my_analysis_package/
|-- include
|  `-- DummyAnalysis.h
|-- scripts
|  `-- analysis_cfg.py
`-- src
    |-- DummyAnalysis.cc
    |-- classes.h
    `-- classes_def.xml
```

3 directories, 5 files

Annotations:

- Red arrow from `DummyAnalysis.h` to `RDF analysis backend (C++ static functions)`
- Red arrow from `analysis_cfg.py` to `Example RDF analysis script`
- Red bracket from `classes.h` and `classes_def.xml` to `reflexion rules/selection to build ROOT dictionary`

# Case studies analysis packages



Added a new `init` mode to `fccanalysis` utility to generate basic analysis package:

```
usage: fccanalysis init [-h] [--script SCRIPT] [--author AUTHOR] [--description DESCRIPTION] [--standalone] [--output-dir OUTPUT_DIR] name

optional arguments:
  -h, --help            show this help message and exit

User options:
  name                  name of the analysis to be built
  --script SCRIPT       name of the main analysis utility script
  --author AUTHOR       author's "name <email@address>" (will use git-config if not specified)
  --description DESCRIPTION
                        analysis package description
  --standalone          also add CMake directive to build standalone package
  --output-dir OUTPUT_DIR
                        output directory where the analysis package will be written
```

With a combination of `--output-dir` and `--standalone`, allows to store the output package outside “common” area, e.g. to work on it externally.

- Generates a `CMakeLists.txt` set of directives to build the shared object, ROOT dictionary files, and set of includes required by RDF.

# Work in progress



- Properly handle CI cases for packages: either for an example package already present in directory, or by generating a new standalone package
  - With CI integration of test analysis package, a spurious, “unreproducible-outside-CI” (i.e. with local `ctest` run) error occurs, still under investigation
- Leave the possibility for the user to select which packages require compilation. Several possibilities for implementation, for instance:
  - manually (list of `make FCCAnalysis_XXX`),
  - defining a set of dependencies at runtime, and leaving the `fccanalysis` script to build the object in background