



Abstract ID : 66

## The Awkward World of Python and C++

### Content

There are undeniable benefits of binding Python and C++ to take advantage of the best features of both languages. This is especially relevant to the HEP and other scientific communities that have invested heavily in the C++ frameworks and are rapidly moving their data analyses to Python.

The version 2 of Awkward Array, a Scikit-HEP Python library, introduces a set of header-only C++ libraries that do not depend on any application binary interface. The users can directly include these libraries in their compilation, rather than linking against platform-specific libraries. This new development makes the integration of Awkward Arrays into other projects easier and more portable as the implementation is easily separable from the rest of the Awkward Array codebase.

The code is minimal, it does not include all of the code needed to use Awkward Arrays in Python, nor does it include references to Python or pybind11. The C++ users can use it to make arrays and then copy them to Python without any specialised data types - only raw buffers, strings, and integers. This C++ code also simplifies the process of JIT-compilation in ROOT. This implementation approach solves some of the drawbacks like packaging projects where native dependencies can be challenging.

In this talk, we will demonstrate the techniques of exposing C++ classes and their methods to Python and vice versa. We will also describe the implementation of a new `LayoutBuilder` and a `GrowableBuffer` that are more performant in building the Awkward Arrays as compared to the previous approach. Furthermore, examples of wrapping the C++ data into Awkward Arrays and exposing Awkward Arrays to C++ without copying them will be discussed.

### Significance

This submission represents our evolving view of best practices for creating Awkward Arrays in C++. Previously, the main codebase was written in C++ with the idea that downstream code would link to `libawkward.so`, but as Jim Pivarski described in his talk at the last ACAT, that route is full of hidden gotchas. This method of a small, header-only library that only fills array buffers for downstream code to pass from C++ to Python using C types (integers and raw pointers) has considerably more promise. Already, two applications are ready to use it: Awkward  $\longleftrightarrow$  RDataFrame (which needs the header-only library to JIT-compile) and `ctapipe` in gamma ray astronomy (which has array types that are known at compile-time).

### References

### Experiment context, if any

IRIS-HEP

**Primary authors:** GOYAL, Manasvi (Delhi Technological University); OSBORNE, Ianna (Princeton University); PIVARSKI, Jim (Princeton University)

**Presenter:** GOYAL, Manasvi (Delhi Technological University)

**Track Classification:** Track 1: Computing Technology for Physics Research

**Contribution Type:** Oral

Submitted by **GOYAL, Manasvi** on **Friday, July 8, 2022**