

Extending ROOT through Modules

Monday 9 July 2018 15:15 (15 minutes)

The ROOT software framework is foundational for the HEP ecosystem, providing capabilities such as IO, a C++ interpreter, GUI, and math libraries. It uses object-oriented concepts and build-time modules to layer between components. We believe additional layering formalisms will benefit ROOT and its users.

We present the modularization strategy for ROOT which aims to formalize the description of existing source modules, making available the dependencies and other metadata externally from the build system, and allow post-install additions of functionality in the runtime environment. Modules can then be grouped into packages, installable from external repositories to deliver post-install step of missing packages. This provides a mechanism for the wider software ecosystem to interact with a minimalistic install. Reducing intra-module dependencies improves maintainability and code hygiene. We believe helping maintain the smallest “base install” possible will help embedding use cases.

The modularization effort draws inspiration from the Java, Python, and Swift ecosystems. Keeping aligned with the modern C++, this strategy relies on forthcoming features such as C++ modules. We hope formalizing the module layer will provide simpler ROOT installs, improve extensibility, and decrease the complexity of embedding in other ecosystems.

Authors: BOCKELMAN, Brian Paul (University of Nebraska Lincoln (US)); SHADURA, Oksana (University of Nebraska Lincoln (US)); VASSILEV, Vassil (Princeton University)

Presenter: SHADURA, Oksana (University of Nebraska Lincoln (US))

Session Classification: T5 - Software development

Track Classification: Track 5 –Software development