

# IRIS-HEP Fellows Program Project Proposal

## **Integrate File System Spec (*fsspec*) into Uproot**

Fellow: Luis Antonio Obis Aparicio

Mentors: Jim Pivarski and Nicholas Smith

### **Project Summary**

The [ROOT framework](#) is arguably the most important software tool for high energy physics (HEP). Much of the HEP data is stored in the ROOT format in the form of TTrees and many HEP analysis make use of ROOT in order to transform data and produce derived objects such as histograms, all usually stored inside ROOT files.

While the ROOT framework offers robust capabilities for storing and manipulating HEP data, it is accompanied by a notable complexity. Utilizing ROOT involves a steep learning curve, and due to its extensive capabilities, it becomes a substantial dependency.

[Uproot](#) [1], a lightweight Python library, allows direct reading of ROOT files without compiled code dependencies. This native compatibility empowers researchers to effortlessly work with ROOT data using familiar Python libraries like [NumPy](#) and [Matplotlib](#), broadening accessibility and fostering collaborative engagement.

Currently Uproot is capable of reading ROOT files from different sources such as the local filesystem, an HTTP endpoint or [XRootD](#). Other sources such as [S3](#) are to be implemented.

The goal of this project is to increase the compatibility of Uproot with other file protocols while offloading the specifics of the different protocols into a dedicated third-party library, *fsspec*.

The [File System Spec](#) [2] (*fsspec*) is a Python library designed to provide a unified and abstracted interface for working with various file systems and storage backends using a consistent API.

As part of this project, we will integrate *fsspec* as a dependency into Uproot and adapt the code to leverage the *fsspec* API for file interactions. Through this integration, Uproot will gain the ability to seamlessly access various storage backends such as HTTP or S3 without requiring dedicated code for each. Consequently, the existing implementation specific to different sources can be eliminated, leading to a more streamlined and simplified codebase. In particular, hand-written redirect logic can be removed from the HTTP backend, and wildcard file-matching can be consistently added to all backends.

Upon successful completion of this project, custom HTTP and XRootD backends will be removed from Uproot, in favor of *fsspec*, and issues related to backends can be directed to the *fsspec* libraries, such as [fsspec-xrootd](#) [3], rather than Uproot itself. This project will also result in a significant refactoring of the Uproot module which provides asynchronous capabilities.

This project focuses on integrating fsspec with Uproot for reading files, but fsspec also provides remote file writing capabilities. Implementing this feature would fall out of scope of this project, however, if circumstances permit, this possibility will be explored.

Besides the code contribution, existing documentation and examples will be updated and tests for different backends will be added to validate the implementation. At the end of the project there will be a presentation of results for fellow IRIS-HEP collaborators.

## Proposed Timeline

Weeks	Goals
1-3	Study the Uproot code responsible for reading from different storage backends, as well as the code responsible for asynchronous tasks. Become familiar with fsspec understanding how it works via the documentation and examples. Start to plan the integration strategy. Create a GitHub PR to centralize design discussions.
4-6	Begin writing the code for the integration of fsspec in Uproot. Implement new tests to increase coverage for different file sources. Define a strategy to make files available in different backends for testing. Begin writing documentation. Address concerns of users and developers to reach a final design.
7-8	Finish most of the core development work. Identify and address potential issues. Finish most of the documentation. If there is available time, begin the implementation of remote file writing using fsspec.
9-10	Refine and optimize code. Finish documentation.
11-12	Prepare the final presentation.

## References

- [1] <https://github.com/scikit-hep/uproot5>
- [2] <https://filesystem-spec.readthedocs.io/>
- [3] <https://github.com/CoffeaTeam/fsspec-xrootd>