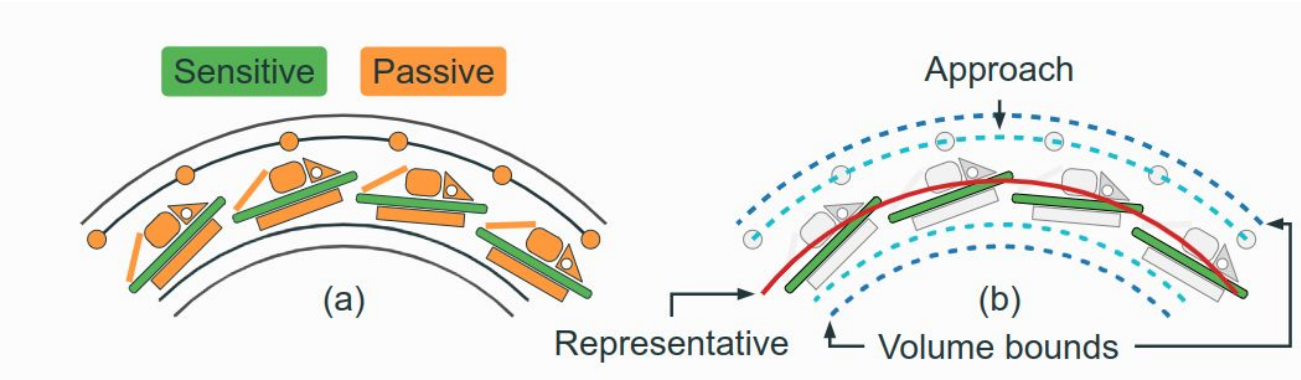
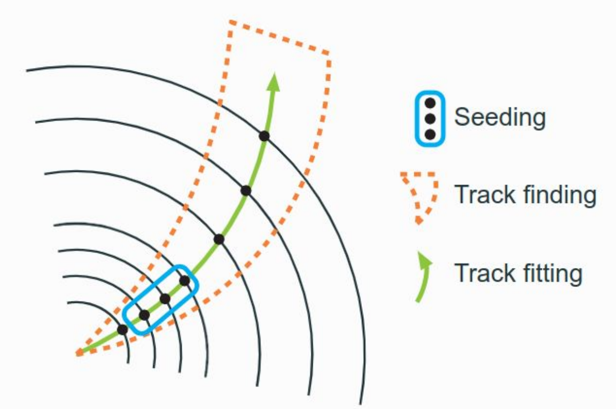
Strategy for Future ACTS Integration and Development

Wouter Deconinck



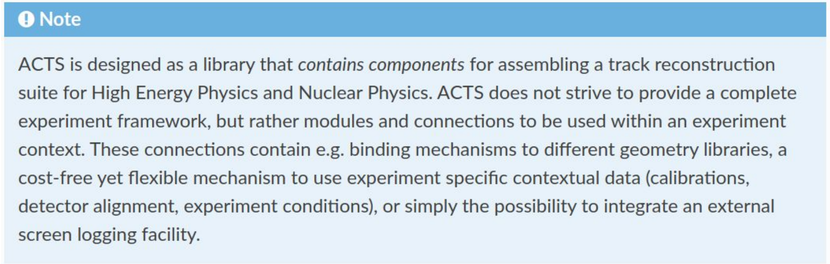


What Is ACTS?

A Common Tracking Software, `ACTS`, or `Acts`, is a **software library** that provides **common track reconstruction algorithms** for HEP and NP experiments, including ePIC.

`ACTS` is **not** a full event reconstruction suite: it provides the building blocks for experiments to combine in their event reconstruction framework (e.g. `JANA2`).

`ACTS` is best thought of as a **set of interfaces** to use when developing (or using) modern and computationally efficient track reconstruction algorithms.



Note

ACTS is designed as a library that *contains components* for assembling a track reconstruction suite for High Energy Physics and Nuclear Physics. ACTS does not strive to provide a complete experiment framework, but rather modules and connections to be used within an experiment context. These connections contain e.g. binding mechanisms to different geometry libraries, a cost-free yet flexible mechanism to use experiment specific contextual data (calibrations, detector alignment, experiment conditions), or simply the possibility to integrate an external screen logging facility.

What Key Components Are In ACTS

`ACTS` consists of three major components

- `Core`: the data structures, interfaces, and algorithms used throughout ACTS,
- `Plugins`: optional functionality, e.g. `DD4hep`, `EDM4hep`, and `Podio` of interest to ePIC, but also `Cuda` acceleration and `ONNX` AI/ML plugins,
- `Examples`: demonstrations of how to use the `Core` and `Plugins` functionality with minimal distracting features.
  - `Examples` are the basis for many algorithms in `EICrecon`, and stay close to their structure.

In addition, but less directly relevant to ePIC, `ACTS` contains extensive validation and benchmarking tools to ensure changes do not introduce regressions, including the triggering of experiment validation workflows (for ATLAS).

`ACTS` allows us to have performant and validated track reconstruction.

What Key Components Are In ACTS Core?

- `EventData`: definitions of data structures that hold measurements, space points, track states, tracks, trajectories, etc. In some cases, these are C++ *concepts* that define the properties that data structures in our data model have to satisfy to be used directly by ACTS.
    - TASK: Within ePIC we would like to migrate away from using `Acts::Examples` data structures and towards having our own data types satisfy the required interfaces directly.
- `Geometry`: ACTS primarily uses surface-based modeling of the detector geometry (as opposed to volume-based modeling in e.g. Geant4). A finite and sufficient set of surfaces is supported (cone, cylinder, disc, plane, straw,…) with associated bounds on the surface coordinates. Geometry defined in `DD4hep` is converted by an ACTS plugin.

`ACTS`'s focused on performance: zero-copy storage, optimized surface functions

What Key Components Are In ACTS Core?

- `Material`: passive material in the geometry is projected onto binned surfaces for rapid evaluation of the impact on tracks.
  - TASK: There is some freedom in how material maps are binned, and as the ePIC geometry description becomes more realistic (tiling, supports), the material maps need validation.
- `Navigation`: includes propagators, `Propagator<Navigator, Stepper>`, with various navigators from surface to surface, and various steppers to integrate the EOM (e.g. RKDP in `EigenStepper`)
  - TASK: Navigators track particles to world boundaries in fringe fields. Aborters are available in ACTS to avoid this, but are not currently used in ePIC.
- `MagneticField`: includes efficient interpolated field map functionality
  - TASK: Ensure we take full advantage of cell caching even when using the `DD4hep` magnetic field backend.

`ACTS`'s focus on performance: materials maps, compile-time EOMs in steppers

What Key Algorithms Are In ACTS Core?

- Seeding: Orthogonal Seed Finder
- Ambiguity Resolution: Greedy, Score-based
- Track Finding: Combinatorial Kalman Filter (CKF)
- Track Fitting: CKF, Gaussian Sum Filter, Kalman Filter (wip), Global Chi-Square Fitter
- Vertexing (linearizer, finder, fitter): Adaptive Multi-Vertex Finder, Iterative Vertex Finder, Full Billoir Vertex Finder
- …

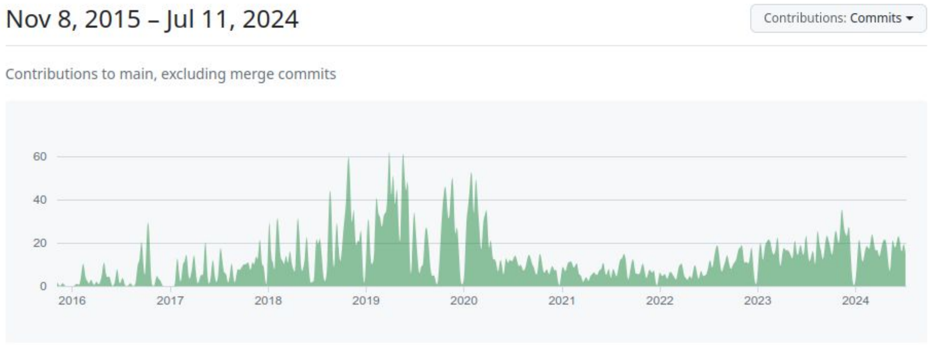
This is more than we are currently using in ePIC (most of the focus has been on CKF), and excludes activity that aims to apply ACTS in tracking for far-forward and far-backward detectors (*telescope* detectors).

Like ePIC, ACTS Is An Actively Developed Project

- ACTS is a library of interfaces but also a testbed for new algorithms (ExaTrkX, various machine learning approaches, Cuda and SYCL acceleration, new geometry model)
- Software projects that are not actively being developed and maintained are more often than not a sustainability risk.

We benefit from the development of new algorithms in `ACTS` (now v35):

- `OrthogonalSeedFinder` as of v20
- `AmbiguityResolution` as of v27
- `FullBilloirVertexFitter` v27



Nov 8, 2015 – Jul 11, 2024

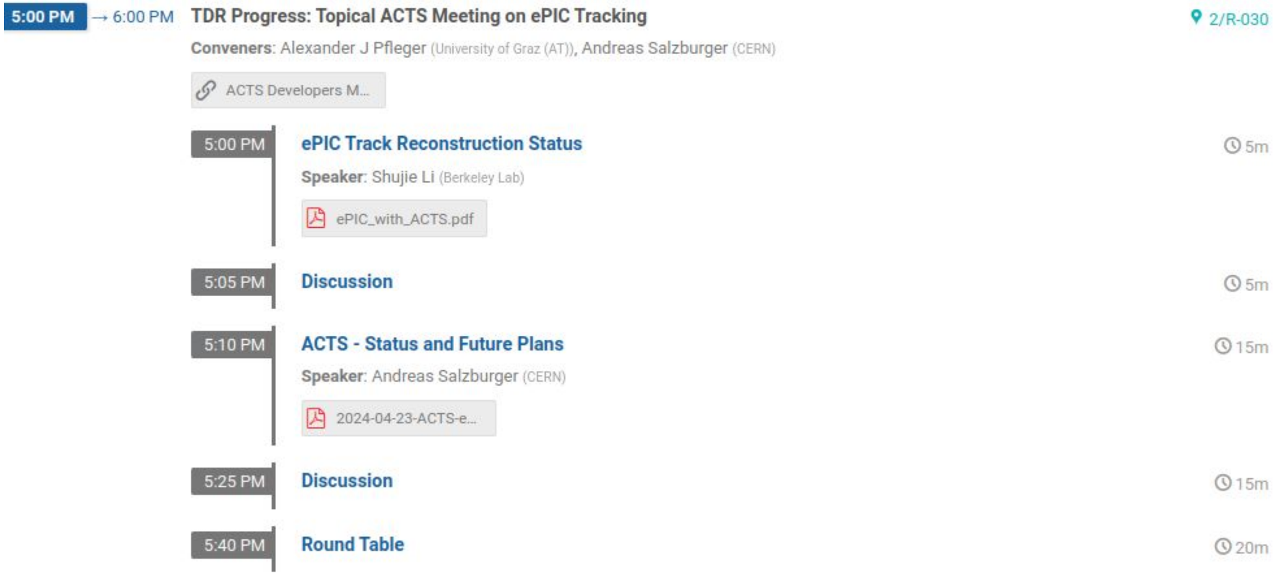
Contributions: Commits ▾

Contributions to main, excluding merge commits

Note: ePIC 23.11 had v26, 24.07 has v31

Our Collaboration With The ACTS Project

- Attendance at weekly ACTS developer meetings (Tu, 11am)
- Integration in ~ePIC-ACTS channel on ACTS Mattermost
- Joint ePIC S&C meeting / ACTS developer meeting (CERN, April 23, 2024):



5:00 PM → 6:00 PM **TDR Progress: Topical ACTS Meeting on ePIC Tracking** 2/R-030

Conveners: Alexander J Pfleger (University of Graz (AT)), Andreas Salzburger (CERN)

ACTS Developers M...

5:00 PM **ePIC Track Reconstruction Status** 5m
Speaker: Shujie Li (Berkeley Lab)
ePIC_with_ACTS.pdf

5:05 PM **Discussion** 5m

5:10 PM **ACTS - Status and Future Plans** 15m
Speaker: Andreas Salzburger (CERN)
2024-04-23-ACTS-e...

5:25 PM **Discussion** 15m

5:40 PM **Round Table** 20m

Risks Of Running Behind Upstream Project Versions

As a collaboration, we have decided to use **community-supported software** libraries, work with developers to include our features, and to contribute back to them by developing those features ourselves.

We have **multiple such dependencies**: `ACTS`, `Geant4`, `DD4hep`, `podio`, `EDM4hep`,... Since we are often on the requesting end of new functionality, our incentive is to upgrade within a few months of each release.

Piecewise upgrades (instead of skipping multiple versions) allows us to identify new issues sooner, narrow the source of issues to a shorter range in time, and spreads the effort involved in upgrading across time.

Risks Of Running Behind Upstream Project Versions

Moreover, components in the software ecosystem are **interconnected**.

- `ACTS` depends on `DD4hep`, `EDM4hep`, `podio`. Newer versions of these dependencies are not guaranteed to work with older `ACTS` versions.
- The current version of `podio` in `eic-shell` requires adding custom patches to the older version of `ACTS` that we are using (issues that are fixed in the latest `ACTS` version).

Integrating functionality now for older version of `ACTS` is **causing double work**:

- Secondary vertexing data model in `ACTS` v33 must first be reverted from the `Examples` in `ACTS`, and will have to be reintroduced again when we upgrade.

Decoupling Upgrade-ability and Upgrades

Thanks to recent and ongoing work in developing **validation benchmarks** for the track and vertex reconstruction code, we can now more easily assess what the impact of upgrades will be (spoilers: usually none at all).

We can **support newer `ACTS` versions** in `EICrecon` **without rolling them out** in the `eic-shell` containers and for production campaigns.

This allows creation of dedicated pre-release environments to allow validation benchmarks to run on newer versions, by the track reconstruction working group.

□ New `ACTS` versions will be supported by `EICrecon` soon after release, allowing new pre-release environments to be provided (e.g. `24.09-alpha`).

□ Validation will be allowed to take place for one month from `alpha` release.

Decoupling Upgrade-ability and Upgrades

Projected timeline:

ePIC	Availability	ACTS Version
24.08	August 0x, 2024	31.2.0
24.09-alpha	July 1x, 2024	33.1.0
24.09	September 0x, 2024	33.1.0
24.10-alpha	August 1x, 2024	35.2.0
24.10	October 0x, 2024	35.2.0
24.11-alpha	September 1x, 2024	36.x.0
24.11	November 0x, 2024	36.x.0

Updated ACTS Versions Allow New Functionality

Weekly `ACTS` developer meetings, https://indico.cern.ch/category/7968/ (Tu 11am)

- v32: Cleaner separation of finding/fitting, smoothing, extrapolation in the CKF: allows changing settings of each step independently in ePIC
- v33: Remove templating in vertex data structures: directly affects secondary vertex finding in ePIC
- …
- More in slides by Andreas Stefl in ePIC joint track reconstruction meeting, June 27, 2024: https://indico.bnl.gov/event/23932

These changes are **added functionality** with **minor changes** in the `Core` components (support already as PR in `EICrecon`)

Specific Tasks In The Medium Term

- 24.09: Remove need to copy `EDM4eic` measurements into `ACTS` structures and use `IndexSourceLink` for direct access instead (zero-copy storage).
- 24.09: Use `EDM4eic` ability to store `ACTS` reference surfaces in output file (remove `ACTS` geometry conversion dependence in output files)
- 24.10: Modify the `ACTS` plugin for `Podio` data models to remove the need to copy from `EDM4eic` into the `ACTS` data model and back again after track fitting (zero-copy storage, this enables front-to-back association of tracking objects to truth information)
- 24.10: Use `SurfaceReachedAborter` to prevent unnecessary propagation
- 24.12: Avoid combinatorial inefficiency in propagation of multiple tracks to multiple surfaces (dRICH, calorimeters)

Milestones defined, in large part, by the upgrade schedule to new `ACTS` versions.

Summary

ACTS is an active project with significant ePIC developer involvement, similar to some other components of our software stack (DD4hep, podio, EDM4hep).

We can stay up to do date with ACTS releases more closely than we have been doing, and provide support for new ACTS functionality to the ePIC collaboration.

- □ Support new ACTS versions in EICrecon soon after release
- □ Publish pre-release environments at least one month in advance
- □ Track reconstruction validation benchmarks on pre-release environments

New features under development in the ACTS stack inside EICrecon:

- Zero-copy integration of EDM4eic data structures with ACTS algorithms
- New propagation aborters to improve performance of track/cluster matching