

Update from JANA2

Nathan Brei

nbrei@jlab.org

"What does JANA do?"

- Organizes event processing code
 - Plugin architecture simplifies build, allows extreme configuration at runtime
 - Decouples inputs, outputs, and algorithms from each other
 - Results are calculated exactly once and only when needed
- Provides multithreading
 - Minimize the amount of parallel programming knowledge needed
 - Encourage patterns for achieving safe and correct program behavior

"How do I create a plugin for ejana?"

- EJANA is a lightweight layer on top of JANA. Compare:
 - `$ jana -Pplugins=Tutorial,MyCrazyAnalysis path/to/inputfile.root -Pnthreads=4`
 - `$ ejana -Pplugins=g4e_reader,trk_fit,trk_eff path/to/g4e_output.root -Pnthreads=4`
- Idea: Start by understanding JANA first!
- BRAND NEW JANA TUTORIAL:
 - <https://jeffersonlab.github.io/JANA2/Tutorial.html>
- Script for generating *commented, buildable* code:
 - Entire project skeletons: ``$ jana-generate.py plugin MyCrazyAnalysis``
 - Individual components: ``$ jana-generate.py JFactory ActsTrackingFactory``

Takeaways

- If you want to create an ejana plugin, work through the tutorial on creating a standalone plugin first. (Should take 30 mins – 1 hr)
- The tutorial only covers the basics. For more advanced topics, see the 'Howtos' tab. (Many of these are works in progress)
- If you are confused, ask me questions! If something is broken, file an issue! This will help me improve both the documentation and the underlying code.
 - nbrei@jlab.org
 - <https://github.com/JeffersonLab/JANA2/issues>
- Dmitry and I are working on streamlining the process of creating plugins within Docker and Jupyterlab. Meanwhile, you can build everything locally using ejana.

Thank you!