

JENA WP2 Survey working document

Useful links:

[List of relevant documents](#)

[Participants of the working group](#)

[Indico](#)

Please add questions as you see fit within one of the three sections of the survey. Use comments to suggest edit to existing questions or ask for clarification.

1. About You

- What experiment/collaboration are you working in?
 - Is this in the ECFA/APPEC/NuPEC/Other domain?
- What is your role in software development?
 - Formal experiment responsible (e.g., software coordinator)
 - Researcher who uses code significantly
 - Research software engineer
- What percentage, roughly, of your time do you dedicate to software matters (not just writing code, but all software-centered activities)

2. Current Practice

- Which programming languages do you use in your research today?
- Which computing platforms do you make use of today?
 - CPU: x86_64 (Intel/AMD), aarch64 (ARM), PowerPC
 - GPU: Nvidia, AMD, Intel
- On modern processors with multiple cores, how do you run?
 - Only one serial core is used
 - Run multiple jobs to use all cores
 - Run multi-threaded job(s) to use all cores
- On modern processes with vector (SIMD) registers, how do you run?
 - No significant use of vector instructions
 - Some code is vectorised automatically
 - Some code is vectorised explicitly
- Do you make use of GPUs / FPGAs today?
 - If yes, why (required for performance / hardware constraints / other)?
- What major software dependencies does your research software rely on?

- E.g., CORSIKA, Geant4, ROOT, PyTorch
- Do people enter your field with the right software skills?
 - If not, how do you train them?
- Is a lack of software expertise and/or losing software experts a problem for your research?
- Is software sustainability a problem in your research area?
- Do you adopt FAIR for software¹ principles?
- Do you work on cross-domain (or cross-collaboration?) software?
- Do you feel the need for software engineers within your research team?
- How do you acquire new skills?
 - In person Schools? MOOC? Website tutorial? Video?Other?
- Which kind of training content you would like in the future
 - In person Schools? MOOC? Website tutorial? Video?Other?

3. Foreseen evolution

- Which programming languages do you anticipate using in the future, and why? (Including ones you think you will have to use and ones you would like to use)
- Do you anticipate using more hardware architectures than today? Which ones?
- Will you take more advantage of multi-core processors than today?
 - Less/same/more
- Will you take more advantage of vector/SIMD CPU registers than today?
 - Less/same/more
- Do you foresee the use of GPUs / FPGAs?
 - If yes, why (required for performance / hardware constraints / other)?
- Do you have the resources needed to evolve your software as you will need to?
 - If not, what resources are you missing?
- Do you think that software performance will be an issue for you in the next decade?
 - If yes, is this because of... increased data volumes, increased data complexity, pushing the science reach?

¹ <https://doi.org/10.15497/RDA00068>

4. Institutional changes

- Do you feel software effort is fairly recognized by your institution?
- Does your institution provide short, medium and/or long-term career paths for software specialists?
 - Do you think they need to?
- Do you think the growing demand for software specialists could pose problems for your research?
 - If yes, how and why?
- Does your institution provide software development guidelines?
 - If yes: do you feel this is useful?
 - If not: do you feel the need for one?
- Does your institution provide software licensing guidelines?
 - If yes: do you feel this is useful?
 - If not: do you feel the need for one?
- What do you see as the biggest institutional challenges related to software in your field?