



***art* news and a proposal for *art* 3.01**

Kyle J. Knoepfel

28 June 2018

User developments with *gallery*

- *gallery* is a library that provides quick access to data products:
 - It is not a framework
 - It provides minimal provenance
- *gallery* users in LArSoft/MicroBooNE and Muon g-2 have developed utilities to enable easier:
 - Wes Ketchum: explorations using *gallery* with Jupyter
 - <https://indico.fnal.gov/event/17395/contribution/3/material/slides/0.pdf>
 - James Stapleton: heist—a Python wrapper for using *gallery*
 - <https://github.com/drstapleton/heist>
- Test them out.

Repeating module labels

- It is allowed to repeat module labels in paths:

```
tp1: [a, b]
tp2: [a, c]
```

- *art* guarantees that a module corresponding to a given label is executed only once per event.
- It is safe for both modules *b* and *c* to read products produced by module *a*.

Repeating module labels

- It is allowed to repeat module labels in paths:

```
tp1: [a, b]
tp2: [a, c]
```

- *art* guarantees that a module corresponding to a given label is executed only once per event.
- It is safe for both modules *b* and *c* to read products produced by module *a*.
- Occasionally, a module label may be repeated in one path:

```
tp: [a, b, ..., j, k, b, l]
```

- Although this is not an error, it is inefficient and it makes it more difficult understanding the graph of data dependencies between modules.

Proposal for *art* 3.01.00

1. For repeated module labels that do not correspond to filters, any repeated module label instances will be removed:

[a, b, b, c] → [a, b, c]
[a, b, c, b] → [a, b, c]

Proposal for *art* 3.01.00

1. For repeated module labels that do not correspond to filters, any repeated module label instances will be removed:

```
[a, b, b, c] → [a, b, c]
[a, b, c, b] → [a, b, c]
```

2. For repeated module labels corresponding to filters, any repeated module label instances with the same filtering action will be removed:

```
[a, f, b, f, c] → [a, f, b, c]           // normal filtering
[a, "!f", b, "!f", c] → [a, "!f", b, c] // negated filtering
[a, "-f", b, "-f", c] → [a, "-f", b, c] // ignored filtering
```

Proposal for *art* 3.01.00

3. If negated filtering is specified for one module label instance, and normal filtering is specified for the other, then the second module label instance is removed **as well as all trailing module labels**:

```
[a, "!f", b, f, c] → [a, "!f", b]  
[a, f, b, "!f", c] → [a, f, b]
```

Proposal for *art* 3.01.00

3. If negated filtering is specified for one module label instance, and normal filtering is specified for the other, then the second module label instance is removed **as well as all trailing module labels**:

```
[a, "!f", b, f, c] → [a, "!f", b]  
[a, f, b, "!f", c] → [a, f, b]
```

4. If ignored filtering is specified for the first module label instance, but not the second, then no module label specifications are removed:

```
[a, "-f", b, f, c] → [a, "-f", b, f, c]  
[a, "-f", b, "!f", c] → [a, "-f", b, "!f", c]
```

Proposal for *art* 3.01.00

5. If ignored filtering is specified for the second module label instance, but not the first, then the second module label instance is removed:

```
[a, f, b, "-f", c] → [a, f, b, c]  
[a, "!f", b, "-f", c] → [a, "!f", b, c]
```

Proposal for *art* 3.01.00

5. If ignored filtering is specified for the second module label instance, but not the first, then the second module label instance is removed:

```
[a, f, b, "-f", c] → [a, f, b, c]
[a, "!f", b, "-f", c] → [a, "!f", b, c]
```

6. For complicated path specifications, module label removal is determined by moving from left to right:

```
[a, f, b, "-f", c, b, "!f", c, a]
→ [a, f, b, c, b, !f, c, a] // remove '-f' (rule 5)
→ [a, f, b, c, !f, c, a] // remove second 'b' (rule 1)
→ [a, f, b, c] // remove '!f' and all trailing labels (rule 3)
```

Proposal for *art* 3.01.00

- Changes apply to repeated module labels within a path, **not** repeated labels across paths.
- Breaking changes to users:
 - The post-processed configuration will be smaller/different.
 - The number of times a module has been “visited” (as reported in the job summary) may be reduced; the number of times a module has been executed will remain the same.

Talk to your experiments.