

Proposal for common ROOT tree output for NLO programs in C++

Frank-Peter Schilling (Uni. Karlsruhe),
Joanna Weng (ETH Zurich),
Peter Uwer (HU Berlin)

- **Aim**
 - ❑ Common output format for NLO programs
 - ❑ Useful for theory/experiment interaction
 - ❑ Format should not be key issue, but well defined & agreed content
- **Format**
 - ❑ Overlap with LHE file format (but need only part of information)
 - ❑ ROOT tree (Joey's idea)
- **Advantages of using ROOT**
 - ❑ data compression
 - ❑ indexed access to specified events
 - ❑ immediate plotting of the variables
 - ❑ wide acceptance in experimental community
- **NLO authors can provide tree with well tested content (if program itself is not ready for release)**
- **Users can work on trees, make cuts etc.**

- **tt+jet @ NLO calculation (in C++)**
 - ❑ Dittmaier, Uwer, Weinzierl, Phys.Rev.Lett. 98 (2007) 262002
 - ❑ Dittmaier, Uwer, Weinzierl, arXiv:0810.0452 [hep-ph]
- **Had several discussions with Peter Uwer @ CERN about ROOT tree content**
 - ❑ (unfortunately he could not come to LesHouches)
- **Content proposal:**
 - ❑ Number and 4-vectors of particles (initial & final state partons)
 - ❑ x_1, x_2, μ_f, μ_r
 - ❑ Event weights (total, individual subprocesses)
 - ❑ Alternatively: just one weight plus process Id (coding initial state qq, qg, gg, etc.)
 - ❑ Optionally PDF reweights
 - ❑ Per file: Cuts, comments, cross section etc.
 - ❑ Could use sub-folders
 - ❑ Idea: coefficients needed for scale reweight (?)

- **Basic idea: provide wrapper for writing a ROOT tree/ntuple**
 - ❑ **NLO author does not have to know anything about ROOT**
 - ❑ **Only requirement: link to ROOT lib's**
- **Already existing: FROOT (P. Nadolsky)**
 - ❑ **Basic C wrappers which can be easily called from FORTRAN**
 - o **Open, close, book, fill tree**
 - ❑ **Does not define the contents of the tree**
 - ❑ **Already used in MCFM (since v5.3)**
- **Our approach:**
 - ❑ **Define (standardize) content, to be filled by various NLO codes**
 - ❑ **Provide C++ helper classes**
 - ❑ **Can also provide FORTRAN wrapper**

- **Class LhaNLOEvent**

- ❑ Container for the event information which is going to be stored
- ❑ Data members correspond to Tree/ntuple contents per event
- ❑ “Setter” / “getter” functions to be called in NLO code
- ❑ Example:

```
LhaNLOEvent* evt = new LhaNLOEvent(n-ptn,n-wgt);
```

```
evt->setParton(1,px,py,pz,E);
```

```
evt->setParton(2,px,py,pz,E);
```

```
...
```

```
evt->setKinematics(x1,x2);
```

```
...
```

- ❑ Once all information is filled, it can be passed to the ...

- **Class LheNLOTreeWriter**
 - ❑ **Writes LheNLOEvent to ROOT tree**
 - ❑ **Can also write per-file info (comments, cross section etc.)**

```
LheNLOTreeWriter* writer = new LheNLOTreeWriter("data.root");  
...  
int error = writer->writeEvent(evt); // do this many times  
...  
delete writer; // close file
```

- **Note: the NLO author does not have to deal with any details about ROOT etc.**

Outlook: Want to use this to compare tt+jet NLO with other codes (MC@NLO, POWHEG, ALPGEN, MADGRAPH)