A standard format for
Les Houches Event Files

J. Alwall [a] A. Ballestrero [f] P. Bartalini [v] S. Belov [h] E. Boos [i]
A. Buckley [p] J.M. Butterworth [q] L. Dudko [i] S. Frixione [m,e]
L. Garren [s] S. Gieseke [c] A. Gusev [j] I. Hinchliffe [t] J. Huston [u]
B. Kersevan [k] F. Krauss [b] N. Lavesson [ℓ] L. Lönnblad [ℓ]
E. Maina [f] F. Maltoni [a] M.L. Mangano [m] F. Moortgat [n]
S. Mrenna [s] C.G. Papadopoulos [d] R. Pittau [f] P. Richardson [p]
M.H. Seymour [m,r] A. Sherstnev [o] T. Sjöstrand [m,ℓ,*] P. Skands [s]
S.R. Slabospitsky [j] Z. Wąs [g] B.R. Webber [o] M. Worek [g]
D. Zeppenfeld [c]

[a] *Centre for Particle Physics and Phenomenology (CP3), Université Catholique de Louvain, Chemin du Cyclotron 2, B–1348 Louvain-la-Neuve, Belgium*

[b] *Institute for Theoretical Physics, TU Dresden, D–01062 Dresden, Germany*

[c] *Institut für Theoretische Physik, Universität Karlsruhe, D–76128 Karlsruhe, Germany*

[d] *Institute of Nuclear Physics, NCSR "Demokritos" 15310 Athens, Greece*

[e] *INFN, Sezione di Genova, Via Dodecaneso 33, I–16146 Genova, Italy*

[f] *INFN, Sezione di Torino and Dip. Fisica Teorica, Universitá di Torino, via Giuria 1, I–10125 Torino, Italy*

[g] *Institute of Nuclear Physics, PAN, ul. Radzikowskiego 152, 31-3420 Kraków, Poland*

[h] *Joint Institute for Nuclear Research, Joliot-Curie 6, Dubna, 141980 Moscow region, Russia*

[i] *Skobeltsyn Institute of Nuclear Physics, MSU, 119992 Moscow, Russia*

[j] *Institute for High Energy Physics, Protvino, Moscow Region, Russia*

[k] *Faculty of Mathematics and Physics, University of Ljubljana, Jadranska 19, SI-1000 Ljubljana, Slovenia*

[ℓ] *Department of Theoretical Physics, Lund University, Sölvegatan 14A, SE–223 62 Lund, Sweden*

[m]*CERN/PH, CH–1211 Geneva 23, Switzerland*

[n]*Institute for Particle Physics, ETH Zurich, CH–8093 Zurich, Switzerland*

[o]*Cavendish Laboratory,University of Cambridge, J J Thomson Avenue, Cambridge CB3 0HE, United Kingdom*

[p]*Institute for Particle Physics Phenomenology, University of Durham, South Rd, Durham DH1 3LE, United Kingdom*

[q]*Physics and Astronomy Department, University College London, Gower St, London WC1E 6BT, United Kingdom*

[r]*School of Physics and Astronomy, The University of Manchester, Manchester, M13 9PL, United Kingdom*

[s]*FNAL, P.O. Box 500, Batavia, IL 60510, USA*

[t]*Lawrence Berkeley National Lab, Berkeley, CA 94720, USA*

[u]*Michigan State University, East Lansing, MI 48840, USA*

[v]*Department of Physics, University of Florida, Gainesville, FL 32611, USA*

Abstract

A standard file format is proposed to store process and event information, primarily output from parton-level event generators for further use by general-purpose ones. The information content is identical with what was already defined by the Les Houches Accord five years ago, but then in terms of Fortran commonblocks. This information is embedded in a minimal XML-style structure, for clarity and to simplify parsing.

Key words: event generators, Les Houches Accord, file format
PACS: 07.05.Kf, 13.85.-t, 13.87.-a

1   Introduction

The (original) Les Houches Accord (LHA) for user-defined processes [1] has been immensely successful. It is routinely used to pass information from matrix-element-based generators to general-purpose ones, in order to generate complete events for a multitude of processes. The original standard was in terms of two *Fortran commonblocks* where information could be stored, while the actual usage has tended to be mainly in terms of *files* with parton-level events, and increasingly will be used by *C++* generators. Since the format of such event files is not specified by the standard, several different formats are in

---

* Corresponding author. *Email address*: `torbjorn@thep.lu.se`.

current usage. This leads to a duplication of effort when such files are to be parsed, a problem that may increase when more programs are developed for different LHC physics aspects.

Following the recent discussions at the MC4LHC-06 workshop at CERN, and subsequent e-mail exchanges, an agreement on a Les Houches Event File (LHEF) format has been reached among the signing authors, representing several of the most commonly-used parton-level and general-purpose generators, as well as interested end-users. This standard should allow all information specified by the LHA to be read in from any file that is LHEF-compliant, and read using one common parser.

The current agreement only standardizes the storage of information already defined by the LHA, and its wrapping in a lightweight XML-style structure. In order to allow further information to be stored, however, comment lines can be appended to the compulsory information.

2 Existing commonblocks

In the LHA two commonblocks are used to store data. A brief summary follows, as a reminder, but anyone not familiar with the details is referred to the original publication [1].

Initialization information is stored in HEPRUP:

```
INTEGER MAXPUP
PARAMETER (MAXPUP=100)
INTEGER IDBMUP,PDFGUP,PDFSUP,IDWTUP,NPRUP,LPRUP
DOUBLE PRECISION EBMUP,XSECUP,XERRUP,XMAXUP
COMMON/HEPRUP/IDBMUP(2),EBMUP(2),PDFGUP(2),PDFSUP(2),
&IDWTUP,NPRUP,XSECUP(MAXPUP),XERRUP(MAXPUP),
&XMAXUP(MAXPUP),LPRUP(MAXPUP)
```

The first few variables refer to the two incoming beams: identities (IDBMUP), energies (EBMUP), and PDF sets used (PDFGUP, PDFSUP). IDWTUP defines the weighting strategy to be used; e.g., 3 corresponds to accepting all events as they come. The rest defines a set of NPRUP separately identified processes, with cross-section information (XSECUP, XERRUP, XMAXUP) and an integer label (LPRUP) for each.

Information on each separate event is stored in HEPEUP:

```
INTEGER MAXNUP
PARAMETER (MAXNUP=500)
```

```
      INTEGER NUP,IDPRUP,IDUP,ISTUP,MOTHUP,ICOLUP
      DOUBLE PRECISION XWGTUP,SCALUP,AQEDUP,AQCDUP,PUP,VTIMUP,
     &SPINUP
      COMMON/HEPEUP/NUP,IDPRUP,XWGTUP,SCALUP,AQEDUP,AQCDUP,
     &IDUP(MAXNUP),ISTUP(MAXNUP),MOTHUP(2,MAXNUP),
     &ICOLUP(2,MAXNUP),PUP(5,MAXNUP),VTIMUP(MAXNUP),
     &SPINUP(MAXNUP)
```

Here `NUP` is the number of particles in the event, with each particle characterized by its identity (`IDUP`, using the standard PDG numbering [2]), status (`ISTUP`), mother(s) (`MOTHUP`), colours(s) (`ICOLUP`), four-momentum and mass (`PUP`), proper lifetime (`VTIMUP`) and spin (`SPINUP`). In addition the event as a whole is characterized by the an event weight (`XWGTUP`), a scale (`SCALUP`), and the α_{em} (`AQEDUP`) and α_{s} (`AQCDUP`) values used.

Only in one respect is there any need to update the meaning of the variables: the original PDF set numbering in `PDFGUP` and `PDFSUP` was based on the then-prevalent PDFLIB library [3]. Now usage has shifted towards LHAPDF [4], where a set is specified by a file name and a number, and its LHAGLUE [5] interface, where these two are mapped onto a single number. Users of LHAGLUE are recommended to set `PDFGUP` equal to 0, a value that does not clash with PDFLIB, and set `PDFSUP` equal to the LHAGLUE code. We recognize that other schemes may be required in the future, with other `PDFGUP` values.

3   Basic write and read

The basic principle for an LHE file is that, in the representation of both commonblocks, an initial line should contain all information fields that only appear once, while as many subsequent lines as required follow, i.e. `NPRUP` or `NUP` ones, each with information on one process or particle, respectively. On each line, all variables to appear there should be written out in the same order as they appear in the commonblocks, and *with no omissions*. Each value is separated by at least one blank from the preceding one, so that free-format reading is possible.

Wrapper and comment lines will be allowed in the file, as outlined in the next section. Omitting such lines, the rules above lead to a unique structure for a file:

1) Initialization information, given once
a) one line with process-number-independent information:
```
  IDBMUP(1) IDBMUP(2) EBMUP(1) EBMUP(2) PDFGUP(1) PDFGUP(2) ↲
 ↳ PDFSUP(1) PDFSUP(2) IDWTUP NPRUP
```

b) `NPRUP` lines, one for each process `IPR` in the range 1 through `NPRUP`:
   `XSECUP(IPR) XERRUP(IPR) XMAXUP(IPR) LPRUP(IPR)`

2) Event information, repeated as many times as there are events
a) one line with common event information:
   `NUP IDPRUP XWGTUP SCALUP AQEDUP AQCDUP`
b) `NUP` lines, one for each particle `I` in the range 1 through `NUP`
   `IDUP(I) ISTUP(I) MOTHUP(1,I) MOTHUP(2,I) ICOLUP(1,I)` ↶
   ↳ `ICOLUP(2,I) PUP(1,I) PUP(2,I) PUP(3,I) PUP(4,I) PUP(5,I)` ↶
   ↳ `VTIMUP(I) SPINUP(I)`

The continuation arrows ↶ and ↳ are there to emphasize that what is (has to be!) a single line in a file here has been split for reasons of visibility.

We stress that all fields must be written out, to make reading predictable. When space is at premium, there is no need to have more than one blank between fields, to align output in columns, or to use more precision than required. For instance, for particles that do not produce a secondary vertex (or have no spin information) it is enough to write "0." (or "9.", the default value for unknown spin) in the `VTIMUP(I)` (or `SPINUP(I)`) field.

4   Complete file format

All the above data has to be stored in one single file, together with further program-specific information on how the events were generated, i.e. all input needed to reproduce the event sample in the file: generator name and version, processes associated with the respective `LPRUP(IPR)` label, masses, couplings, cuts, etc., as required. Also further information specific to each event can be added, such as the actual parton density values or multiple scales needed for matrix-element-to-shower matching. Currently, such information has to be generator-specific, since it would not be a trivial task to define a common scheme to cover every possible case. This further information must be clearly distinguishable from the compulsory one already described in the previous section, however.

To this end, an XML-like structure has been introduced, where XML-style tags provide blocks where the compulsory and optional information is to be found. The file then looks like

```
<LesHouchesEvents version="1.0">
  <!--
    # optional information in completely free format,
    # except for the reserved endtag (see next line)
  -->
```

```
  <header>
    <!-- individually designed XML tags, in fancy XML style -->
  </header>
  <init>
    compulsory initialization information
    # optional initialization information
  </init>
  <event>
    compulsory event information
    # optional event information
  </event>
  (further <event> ... </event> blocks, one for each event)
</LesHouchesEvents>
```

Indentation of tags and information is allowed (but not required), but the tags defined above must be alone on their respective line. This is crucial for the `<init>` and `<event>` tags, which define the points after which the compulsory information is to be found.

The all-enclosing `LesHouchesEvents` block defines the root element required by XML, and makes it obvious which standard and version the file is based on.

Information on how events were generated — whatever the author deems relevant — should appear in the beginning of the file, inside either or both of the `<!-- ... -->` and `<header> ... </header>` blocks. Both blocks are optional, and the order between them can be reversed. The difference is that the former block can be written in any format desired, except for the reserved endtag, while the `header` block has to be based on pure XML syntax, so that standard XML parsers could be used to extract any specific piece of information. Any comments in this block must also be of the standard `<!-- .... -->` kind.

The `init` block contains the `HEPRUP` initialization information, and each `event` block the `HEPEUP` event information. To ensure simple parsing, the `<init>` line is *immediately* followed by the initialization information as strictly defined by point 1 in the previous section. Correspondingly, each `<event>` line is *immediately* followed by the event information defined by point 2 above.

The advantage of this scheme is that parsing is made simple, since the format to use next is always known, at least for the compulsory information. That is, a character string is read from each line, repeatedly, until either `<init>` or `<event>` is encountered. Then the next line has to be of the form 1a) or 2a), respectively, and the subsequent `NPRUP` or `NUP` ones of 1b) or 2b).

To allow for the future introduction of attributes in `<init>` and `<event>`, so

long as they are all on one line, `<init` or `<event` strings followed by a blank (and then further text on the same line) should be considered as equivalent to the normal forms, in terms of how a file is parsed for Les Houches standard information. Also `<header>` might acquire attributes, but that would of course not affect the LHA information parsing. Private attributes are allowed, but with the understanding that any future standard would take precedence in case of conflict.

Any number of comment lines can follow *after* the compulsory initialization or event information. These lines can contain whatever information desired in whatever format desired, except for the already reserved tags. (Anybody inserting an `<event>` tag as part of a comment should expect parsing to derail.) However, to avoid conflicts with XML parsers, it is recommendable to avoid the reserved symbols `<,>` and `&`, except as part of some proper XML syntax.

Furthermore, it is recommended (but not required) to begin comment lines with an `#` symbol, to make them easily distinguishable from the compulsory information. As an example, there is a demand from the experimental community that the parton densities used event by event should be stored somewhere, for use when reweighting to another parton-density set. This could be accommodated with an optional line of the form
`#pdf id1 id2 x1 x2 scalePDF xpdf1 xpdf2`
where `#pdf` is an identifying label and the rest are the respective values (identity [2], x, Q, $xf_i(x, Q)$).

To read a large file, with many events, is slower with a conventional XML parser than with simple Fortran/C++ code, potentially by orders of magnitude. This is a main reason to keep the event data format simple. The initial information is only to be read once, however, and could therefore use a more sophisticated XML structure. It is then assumed that normal XML parsing could be stopped after `</init>`, as is possible with some of the existing parsers, while ones that load the whole document would be unsuitable.

To summarize, the objective of the current standard is to promote a structure that is completely consistent with XML, so that XML parsers could be used if desired, but also simple enough that all key information should be easily obtainable with fast Fortran/C++ code. It would then not be wise to classify a file as useless simply because it happens to contain some minor XML-syntactical error, like an `&` symbol in a comment. A sensible strategy for writing a minimal event-generator parser is only to rely on the `<init` and `<event` tags opening their respective line, being followed on consecutive lines by the respective compulsory information, and consider everything else as potentially program-dependent.

5 Outlook

Several program authors are committed to implementing the standard in the near future, so that it should rapidly come to ease the intercommunication burden. The webpage

> http://www.thep.lu.se/∼torbjorn/lhef

contains some early examples of Fortran and C++ implementations of parsers.

To allow easy identification of files that follow the standard we propose that these be given a `.lhe` file name extension.

The current proposal must not be viewed as the end of the road. There is much further information exchange that ought to be standardized. It is allowed to use/promote a "private standard" of tags in the `header` block or of additional event information, and experience with such could point the way towards an extended 2.0 standard at a later date. Such complementary work has been performed within the MCDB project [6] and will continue within the LCG/CEDAR HEPML project [7].

References

[1] E. Boos et al., in proceedings of the Workshop on Physics at TeV Colliders, Les Houches, France, 21 May – 1 June 2001 [hep-ph/0109068]

[2] Particle Data Group, W.-M. Yao et al., J. Phys. **G33** (2006) 1

[3] H. Plothow-Besch, Computer Physics Commun. **75** (1993) 396

[4] W. Giele et al., in proceedings of the Workshop on Physics at TeV Colliders, Les Houches, France, 21 May – 1 June 2001 [hep-ph/0204316]

[5] M.R. Whalley, D. Bourilkov and R.C. Group, in proceedings of the HERA and the LHC workshop, eds. A. De Roeck and H. Jung, CERN–2005–014, p. 575 [hep-ph/0508110]

[6] C. Buttar et al., in summary report of the Standard Model and Higgs working group, p. 200, Workshop on Physics at TeV Colliders, Les Houches, France, 2 – 20 May 2005 [hep-ph/0604120]

[7] see webpages `https://twiki.cern.ch/twiki/bin/view/Main/HepML` and `http://hepforge.cedar.ac.uk/hepml/`