



LUND UNIVERSITY



MCnet open meeting  
CERN  
10 January 2007

# PYTHIA 8 Status Report

**Torbjörn Sjöstrand**

CERN/PH and

Department of Theoretical Physics, Lund University

Apologies to participants of GENSER meeting on 22 November 2006:  
only minor updates!

# PYTHIA 6 status

PYTHIA 6 still being actively developed and maintained:

- multiple interactions and underlying event, with
- transverse-momentum-ordered showers
- SUSY interfaces (SLHA) and simulation
- regular bug fixes and minor improvements
- moving to CEDAR HepForge (code management, bugtracking)

Currently PYTHIA 6.409 ( $\approx$  6.408):

- 74,800 lines of code (including comments/blanks)
- 580 page PYTHIA 6.4 Physics and Manual  
T. Sjöstrand, S. Mrenna and P. Skands,  
JHEP05 (2006) 026 [hep-ph/0603175]
- available on <http://www.thep.lu.se/~torbjorn/Pythia.html>
- together with sample main programs, old code, etc.

... but

- only add, never subtract
- ⇒ has become bloated and unmanageable
- is in Fortran 77, so not understood by young people

# PYTHIA 8: A fresh start

Problem: PYTHIA 7 stalled, no other manpower

Solution?: take a sabbatical and work “full-time”!

( $\Rightarrow$  baseline model, S. Mrenna & P. Skands join later ?)

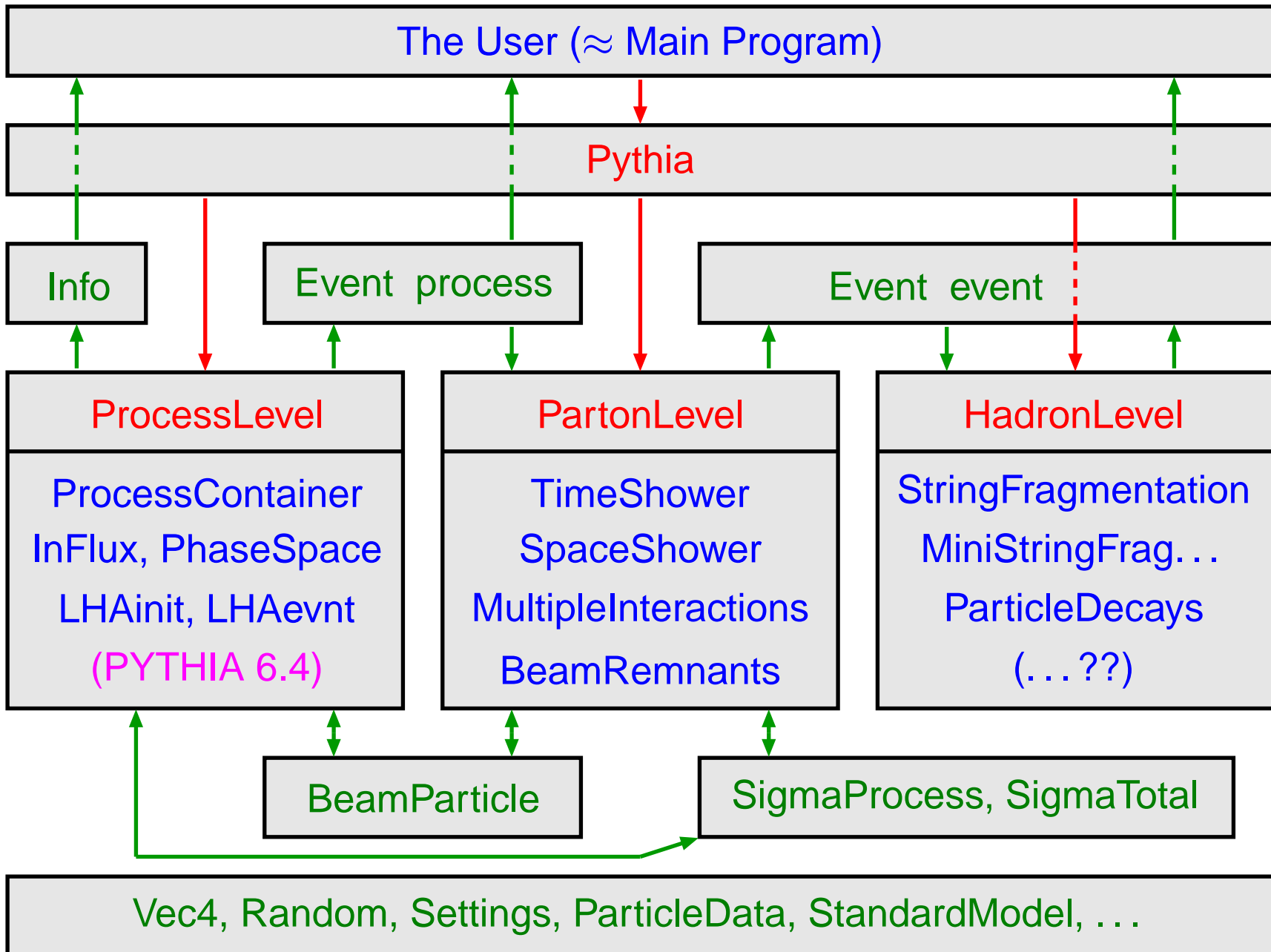
Tentative schedule (spring 2003):

time	date	processes	final states
0 =	1 Sept. 2004	—	—
1 =	1 Sept. 2005	LHA-style input	incomplete draft
2 =	1 Sept. 2006	a few processes	complete, buggy(?)
3 =	1 Sept. 2007	more processes	stable, debugged

Objectives:

- clean up, keep the most recent models
- core program completely standalone, but
- Les Houches Accord style input central
  - interfaces to other libraries foreseen

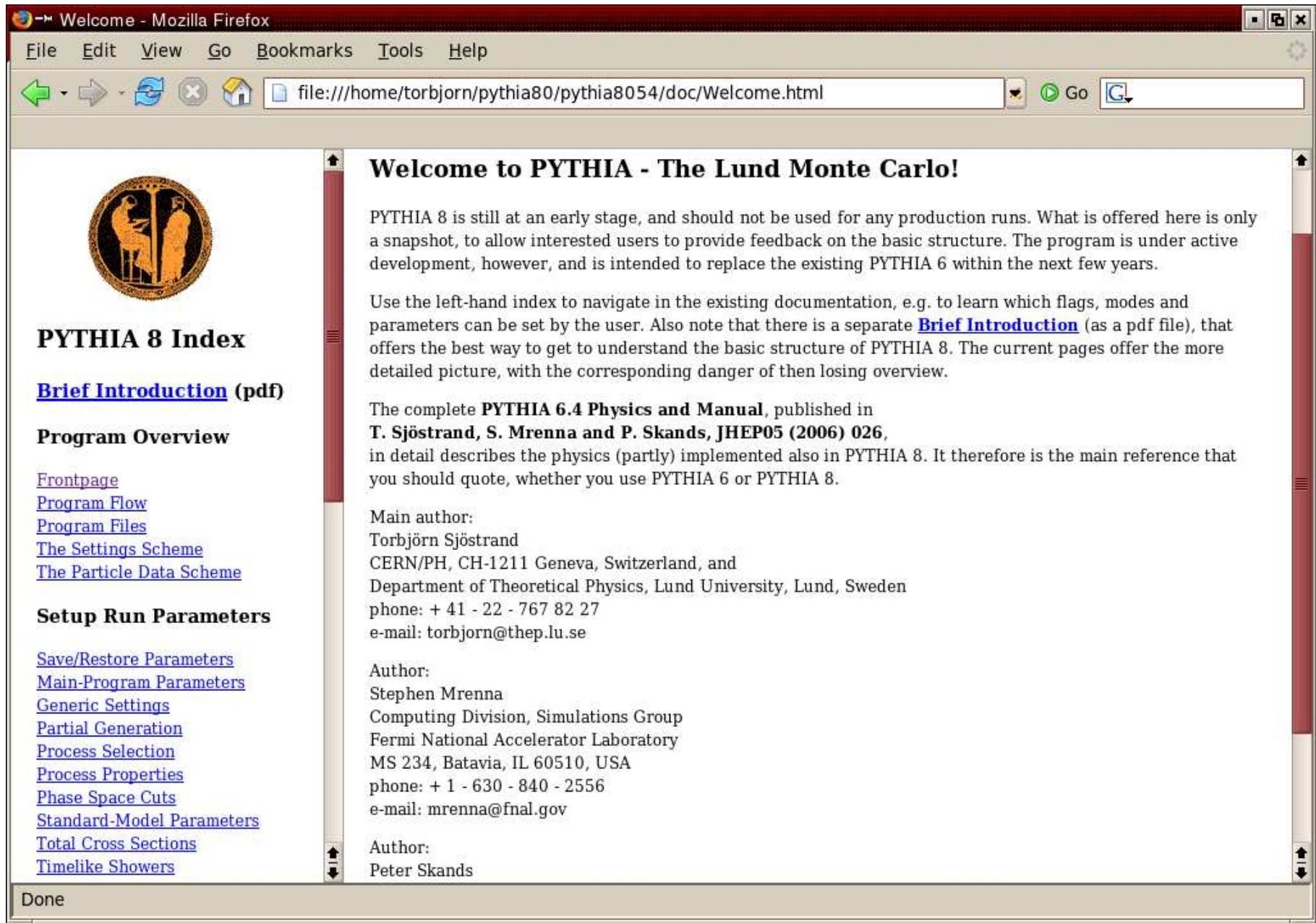
# Current PYTHIA 8 structure



# Example of a main program

```
// File: main01.cc. The charged multiplicity distribution at the LHC.
#include "Pythia.h"
using namespace Pythia8;
int main() {
    // Generator. Process selection. LHC initialization. Histogram.
    Pythia pythia;
    pythia.readString("HardQCD:all = on");
    pythia.readString("PhaseSpace:pTHatMin = 20.");
    pythia.init( 2212, 2212, 14000.);
    Hist mult("charged multiplicity", 100, -0.5, 799.5);
    // Begin event loop. Generate event. Skip if error. List first one.
    for (int iEvent = 0; iEvent < 100; ++iEvent) {
        if (!pythia.next()) continue;
        if (iEvent < 1) {pythia.info.list(); pythia.event.list();}
        // Find number of all final charged particles and fill histogram.
        int nCharged = 0;
        for (int i = 0; i < pythia.event.size(); ++i)
            if (pythia.event[i].isFinal() && pythia.event[i].isCharged())
                ++nCharged;
        mult.fill( nCharged );
    }
    // End of event loop. Statistics. Histogram. Done.
    pythia.statistics();
    cout << mult;
    return 0;
}
```

# Online manual $\implies$ GUI??



Welcome - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

file:///home/torbjorn/pythia80/pythia8054/doc/Welcome.html

## Welcome to PYTHIA - The Lund Monte Carlo!

PYTHIA 8 is still at an early stage, and should not be used for any production runs. What is offered here is only a snapshot, to allow interested users to provide feedback on the basic structure. The program is under active development, however, and is intended to replace the existing PYTHIA 6 within the next few years.

Use the left-hand index to navigate in the existing documentation, e.g. to learn which flags, modes and parameters can be set by the user. Also note that there is a separate [Brief Introduction](#) (as a pdf file), that offers the best way to get to understand the basic structure of PYTHIA 8. The current pages offer the more detailed picture, with the corresponding danger of then losing overview.

The complete **PYTHIA 6.4 Physics and Manual**, published in **T. Sjöstrand, S. Mrenna and P. Skands, JHEP05 (2006) 026**, in detail describes the physics (partly) implemented also in PYTHIA 8. It therefore is the main reference that you should quote, whether you use PYTHIA 6 or PYTHIA 8.

Main author:  
Torbjörn Sjöstrand  
CERN/PH, CH-1211 Geneva, Switzerland, and  
Department of Theoretical Physics, Lund University, Lund, Sweden  
phone: + 41 - 22 - 767 82 27  
e-mail: torbjorn@thep.lu.se

Author:  
Stephen Mrenna  
Computing Division, Simulations Group  
Fermi National Accelerator Laboratory  
MS 234, Batavia, IL 60510, USA  
phone: + 1 - 630 - 840 - 2556  
e-mail: mrenna@fnal.gov

Author:  
Peter Skands

Done



# Example: timelike parton showers

Welcome - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

file:///home/torbjorn/pythia80/pythia8054/doc/Welcome.html

[Process Selection](#)  
[Process Properties](#)  
[Phase Space Cuts](#)  
[Standard-Model Parameters](#)  
[Total Cross Sections](#)  
[Timelike Showers](#)  
[Spacelike Showers](#)  
[Multiple Interactions](#)  
[Beam Remnants](#)  
[Fragmentation](#)  
[Flavour Selection](#)  
[Particle Decays](#)  
[Particle Data](#)

**Study Output**

[Four-Vectors](#)  
[Particle Properties](#)  
[The Event Record](#)  
[Event Information](#)  
[Event Statistics](#)  
[Histograms](#)  
[Event Analysis](#)  
[HepMC Interface](#)

**Link to Other Programs**

[The Les Houches Accord](#)  
[Access PYTHIA 6 Processes](#)  
[Parton Distributions](#)  
[External Decays](#)  
[Random Numbers](#)

**Main variables**

The amount of QCD radiation in the shower is determined by

**parameter name="TimeShower:alphaSvalue" default="0.1265" min="0.06" max="0.25"**  
The  $\alpha_{strong}$  value at scale  $M_Z^2$ . The default value corresponds to the one tuned to LEP data (using a first-order running), so should be taken rather seriously [Rud04].

The actual value is then regulated by the running to the scale  $p_T^2$ , at which the shower evaluates  $\alpha_{strong}$

**mode name="TimeShower:alphaSorder" default="1" min="0" max="2"**  
Order at which  $\alpha_{strong}$  runs,  
**option value="0"**: zeroth order, i.e.  $\alpha_{strong}$  is kept fixed.  
**option value="1"**: first order, which is the normal value.  
**option value="2"**: second order. Since other parts of the code do not go to second order there is no strong reason to use this option, but there is also nothing wrong with it.

QED radiation is currently regulated by StandardModel:alphaEMfix, since no QED running is implemented in the shower.

The rate of radiation is divergent in the  $p_T \rightarrow 0$  limit. Here, however, perturbation theory is expected to break down. Therefore an effective  $p_T$  min cutoff parameter is introduced, below which no emissions are allowed. The cutoff may be different for QCD and QED radiation off quarks, and is mainly a technical parameter for QED radiation off leptons.

**parameter name="TimeShower:pTmin" default="0.5" min="0.1" max="2.0"**  
Parton shower cut-off  $p_T$  for QCD emissions.

**parameter name="TimeShower:pTminChgQ" default="0.5" min="0.1" max="2.0"**  
Parton shower cut-off  $p_T$  for photon coupling to coloured particle.

**parameter name="TimeShower:pTminChgL" default="0.0005" min="0.0001" max="2.0"**  
Parton shower cut-off  $p_T$  for pure QED branchings. Assumed smaller than (or equal to)  $p_{TminChgQ}$ .

Done

# Hard-process generation

Currently limited selection implemented internally:

ProcessGroup	ProcessName
SoftQCD	minBias, elastic, singleDiffractive, doubleDiffractive
HardQCD	gg2gg, gg2qqbar, qg2qg, qq2qq, qqbar2qqbarNew, qqbar2gg, gg2ccbar, qqbar2ccbar, gg2bbbar, qqbar2bbbar
PromptPhoton	qg2qgamma, qqbar2ggamma, gg2ggamma, qqbar2gammagamma, gg2gammagamma
WeakBosonExchange	ff2ff(t:gmZ), ff2ff(t:W)
WeakSingleBoson	ffbar2gmZ, ffbar2W, ffbar2ffbar(s:gm)
WeakDoubleBoson	ffbar2ZW, ffbar2WW
WeakBosonAndParton	qqbar2Wg, qg2Wq, ffbar2Wgm
Charmonium	gg2QQbar[3S1(1)]g + 18 more
Bottomonium	gg2QQbar[3S1(1)]g + 18 more
Top	gg2ttbar, qqbar2ttbar, qq2tq(t:W)
SUSY	qqbar2chi0chi0

(with resonance decays still primitive)

but can use Fortran PYTHIA 6 library transparently via LHA interface.

Can also use Les Houches Accord for any other hard process,  
input via runtime C++ or Fortran interfaces or via LHE files.



# Progress report and future plans

- ★ **August 2005, 12 month mark with 8.040**: no hard processes, simplified machinery for rest, as planned (!)
- ★ Then unforeseen interruption for  $\sim 1/2$  year
- ★ Back at CERN since April 2006, progressing again (summer slowly)
- ★ **September 2006, 18 month mark with 8.060**: basic structure, utilities, process machinery and some hard processes, LHEF, SLHA, ...
- Oct 2006: Decays: update tables (PDG2006, c&b from DELPHI and EVTGEN/LHC-B), new matrix elements, selection, ...
- Nov 2006: Hadronization:  $L = 1$  mesons, popcorn baryons, **ministrings**, ...
- Dec 2006: Multiple Interactions: more processes, ...
- Jan 2007: Showers: photons, **LHAPDF**, ...
- Feb 2007: Showers & MI: interleave FSR with ISR and MI, ...
- Mar 2007: Ditto & beam remnants: colour flow, reconnections, ...
- ★ **March 2007, 24 month mark with 8.080 (?)**: useful version
- ★ 3rd year: more processes, resonance decays, GUI?, **official release**
- ★ Debugged and tuned by LHC startup 2008 (??)
- ★ Overtaking Fortran version usage by 2009 (???)

# Trying It Out

- Download `pythia8070.tgz` from <http://www.thep.lu.se/~torbjorn/Pythia.html>, link “Future”
- Unzip and expand with `tar xvfz pythia8070.tgz`
- Move to the thus created `pythia8070` directory
- Follow the [README](#) instructions (edit links to PYTHIA 6, HepMC)
- `make` will compile in  $\sim 6$  minutes  
(1/3 for archive library, 1/3 for shared, 1/3 for PYTHIA 6)
- The `pythia8070.pdf` file contains an introduction to the program
- Open `doc/Welcome.html` in a web browser for the full manual  
(in the future: GUI with xml + Javascript ?)
- The `examples` subdirectory contains 17 sample main programs  
(`make mainNN` and then `mainNN.exe > outfile`)

Makefiles, install procedure & HepMC interface by Mikhail Kirsanov.