Quick User Guide for HELAC-Onia (v0.1)

March 1, 2021

Disclaimer: this text is updating.
If you face any issue or you need help, please contact us at
nloaccess@ijclab.in2p3.fr.

With this guide we review how to create an input file for HELAC-Onia [1, 2] (version $\geq$ 2.5.0). First of all, we should specify the process (or processes) we want to calculate via:

```
generate {process}
```

where `process` could be built using the syntax for Standard Model particles and quarkonia listed in Tables 1, 2, 3. Every input parameter should then be set according to the following syntax:

```
set {parameter} = {value}
```

Using this syntax, we can build our input file to be sent to the server for the calculation. Note: for setting bool variables, use `set {parameter} = T(F)` for `True`(`False`).

1 Collision parameters

The main parameters are:

- *colpar* (integer): represents the type of colliding particles. The available choices are:
  - `set colpar = 1`: pp
  - `set colpar = 2`: $p\bar{p}$
  - `set colpar = 3`: e^+e^-
  - `set colpar = 4`: e^+p
  - `set colpar = 5`: e^-p
  - `set colpar = 6`: $e^+\bar{p}$
  - `set colpar = 7`: $e^-\bar{p}$
  - `set colpar = 8`: $\gamma_{e^\pm}p$
  - `set colpar = 9`: $\gamma_{e^\pm}\bar{p}$
  - `set colpar = 10`: $\gamma_p p$
  - `set colpar = 11`: $\gamma_p \bar{p}$
  - `set colpar = 12`: $\gamma_{e^\pm}\gamma_{e^\pm}$
  - `set colpar = 13`: $\gamma_{e^\pm}\gamma_p$
  - `set colpar = 14`: $\gamma_p\gamma_p$

  where $\gamma_{e^\pm}$ and γ_p are the photon from $e^\pm$ and elastic p (proton) via the improved Weizsäcker-Williams formula.
- *energy_beam1* and *energy_beam2* (real): these are the energies (in GeV) of beam 1 and beam 2.
- *fixtarget* (bool): a flag to specify whether we want to calculate a collision in the fixed-target mode (`set fixtarget = T`) or not (`set fixtarget = F`).

2 Monte Carlo integration parameters

- *gener* (integer): specifies the Monte Carlo integration generator, i.e.:
  - `set gener = 0` for PHEGAS (recommended for event generation)
  - `set gener = 1` for RAMBO (not recommended)
  - `set gener = 2` for DURHAM (not recommended)
  - `set gener = 3` for VEGAS

- *nmc* (integer): the number of the Monte Carlo iterations, i. e. the total number of phase space points. It can be modified via e.g. `set nmc = 100000`
- *nopt, nopt_step, noptlim* (integers): parameters for Monte Carlo optimization when `gener = 0`. The recommended values are: `nopt = nmc/10`, `nopt_step = nmc/10`, `noptlim = nmc`.
- *ranhel* (integer): is a parameter to determine whether the program uses the Monte Carlo sampling over the helicity configurations. Specifically, if `ranhel = 0`, it does the helicity summation, while if `ranhel`> 0, it does the Monte Carlo sampling. For the latter case, we have:
  - `set ranhel = 1`: the program uses Monte Carlo sampling over the helicities of the elementary particles in the Standard Model and summing over helicities of quarkonia
  - `set ranhel = 2`: it also performs Monte Carlo sampling over L polarization vector (orbital angular momentum) for the P-wave states
  - `set ranhel = 3`: it does Monte Carlo sampling over all polarization vectors of heavy quarkonia (of course also over helicites of the elementary particles in the Standard Model)

3 Theory parameters

- *qcd* (integer): its value determines the theory in which the amplitudes should be calculated in:
  - `set qcd = 0` for only electroweak vertices
  - `set qcd = 1` for electroweak and QCD vertices (i.e. the Standard Model)
  - `set qcd = 2` for only QCD vertices
  - `set qcd = 3` for only QED vertices
  - `set qcd = 4` for QCD and QED vertices
- *alphasrun* (integer): determines whether the strong coupling constant α_S should be running (`set alphasrun = 1`) or not (`set alphasrun = 0`).
- *gauge*, *ihiggs* and *widsch*: they respectively determine the gauge (0 for Feynman gauge, 1 for unitary gauge), whether Higgs should be included (`set ihiggs = 1`) or not (`set ihiggs = 0`) and if we want to use the fixed (`set widsch = 0`) or complex (`set widsch = 1`) mass scheme for the widths of $W^{\pm}$ and Z bosons.
- *Scale* (integer): specifies which renormalization (and PDF factorization) central scale μ_0 should be used. Up to now, the possible choices are:
  - `set Scale = 0`: fixed scale. In this case, we should also supply the value of the scale, via `set FScaleValue = {value}`, where `value` is a real number
  - `set Scale = 1`:
  $$\mu_0 = \sqrt{p_{T1}^2 + m_1^2}$$
  - `set Scale = 2`:
  $$\mu_0 = \sqrt{p_{T1}^2 + \left(\sum_{i=1}^{n} m_i\right)^2}$$
  - `set Scale = 3`: is used only when we calculate e^+e^- showers in the final state (via QEDPS)
  - `set Scale = 4`:
  $$\mu_0 = \frac{H_T}{2} = \frac{\sum_i \sqrt{p_{Ti}^2 + m_i^2}}{2}$$
  - `set Scale = 5`:
  $$\mu_0 = \frac{m_{T1} + m_{T2}}{2} = \frac{\sqrt{p_{T1}^2 + m_1^2} + \sqrt{p_{T2}^2 + m_2^2}}{2}$$

  where we have used the notations p_{Ti} and m_i to stand for the transverse momentum and the mass of the i-th final state.

  Note also that the real scale is given by `Scale*ScaleFactor`, where the latter is a real parameter and can be set via the command `set ScaleFactor = {value}`. It is also possible to set a different scale factor for renormalization and factorization scales. This can be achieved setting specific values for the

real parameters *muR_over_ref* and *muF_over_ref* respectively.
The scale uncertainty could be automatically calculated setting *reweight_Scale* (bool) to be true (`set reweight_Scale = T`). The scale variation is then calculated according to:

$$\texttt{rw\_Rscale\_down} < \frac{\mu_R}{\mu_0} < \texttt{rw\_Rscale\_up}, \quad \texttt{rw\_Fscale\_down} < \frac{\mu_F}{\mu_0} < \texttt{rw\_Fscale\_up} \tag{1}$$

and one would then have to specify the (real) value of *rw_Rscale_down*, *rw_Rscale_up*, *rw_Fscale_down*, *rw_Fscale_up*. Their default values are `rw_Rscale_down` = `rw_Fscale_down` = 0.5 and `rw_Rscale_up` = `rw_Fscale_up` = 2.0.

4 PDF parameters

HELAC-Onia is interfaced to the Les Houches accord PDF, LHAPDF [3]. Currently, LHAPDF v6.2.1 is installed on NLOAccess. To use it, the following parameters should be set:

- *lhapdf* (bool): flag to ask for using the LHAPDF library. Default value is false.
- *pdf* (integer): is the PDF set number proposed in `pdfsets.index` of LHAPDF. `set pdf = 0` means no PDF is convoluted. If one wants to use LHAPDF, before setting the value of *pdf* we have to set *lhapdf* flag to be true (`set lhapdf = T`)
- *pdf_min, pdf_max* (integer): the minimum and maximum LHAPDF id for PDF set members. Note that an even number of members is needed.
- *reweight_pdf* (bool): a flag to ask for PDF reweighting. `set reweight_pdf = T` would activate it and calculate the PDF uncertainty.

At the moment, the following LHAPDF sets are installed:

- CT10 (SetIndex: 10800, Members: 53)
- CT14lo (SetIndex: 13200, Members: 1)
- CT14nlo (SetIndex: 13100, Members: 57)
- CT14nnlo (SetIndex: 13000, Members: 57)
- CT14nnloIC (SetIndex: 13081, Members: 6)
- CT14qed_proton (SetIndex: 13300, Members: 31)
- CT18NLO (SetIndex: 14400, Members: 59)
- cteq6 (SetIndex: 10550, Members: 45)
- cteq66 (SetIndex: 10000, Members: 41)
- cteq6l1 (SetIndex: 10042, Members: 1)
- MMHT2014lo68cl (SetIndex: 25000, Members: 51)
- MMHT2014nlo68cl (SetIndex: 25100, Members: 51)
- NNPDF30_nlo_as_0118 (SetIndex: 260000, Members: 101)

More PDF sets can be installed upon request. We plan to have all the available LHAPDF sets in future.

5 Kinematical cuts

HELAC-Onia is able to set cuts on quantities like absolute rapidity, absolute pseudorapidity, transverse momentum, separation, invariant mass etc.

5.1 Transverse momentum cuts

In order to set a cut on minimum and maximum transverse momentum, it is sufficient to use the commands:

```
set minpt{tag} = {value},     set maxpt{tag} = {value}
```

where tag could be a lepton, a light quark (or anti-quark or gluon, `q` in the following), a heavy quark, a photon and a quarkonium:

tag = `[l, q, c, b, t, p, conia, bonia, Bconia]`

and `value` is a real number. Note that if *maxpt* is set as negative, no such cut is used.

For instance, for setting the minimum charmonia P_T to e.g. 5 GeV, it is sufficient to write:

```
set minptconia = 5.0
```

5.2 (Pseudo)rapidity cuts

HELAC-Onia is also able to decide whether to put cuts on the absolute value of (pseudo)rapidity or on the value itself. This is achieved by setting the value of *absoluterap* (bool) to:

- `set absoluterap = T`: in this case, the cut will then be imposed on $|y|$
- `set absoluterap = F`: in the case the flag is set to false, the cut is imposed on y.

In order to set these cuts, we have for maximum rapidity:

```
set maxrap{tag} = {value}
```

and for y rapidity:

```
set minyrap{tag} = {value},     set maxyrap{tag} = {value}
```

where again tag could be a lepton, alight quark (or anti-quark or gluon), a heavy quark, a photon and a quarkonium:

tag = `[l, q, c, b, t, p, conia, bonia, Bconia]`

and `value` is a real number.

For example, for setting the charmonia y rapidity between 2 and 5, one can write:

```
set minyrapconia = 2.0
set maxyrapconia = 5.0
```

5.3 Separation cuts

It is possible to set separation cuts on the ΔR between some pair of particles via

```
set mindr{tag} = {value}
```

where the tag now can refer to lepton-lepton, lepton-quark, quark-quark, quark-b, b-b and photon-fermion:

tag = `[ll, lq, qq, qb, bb, pf]`

and `value` (real) is the value of ΔR to be used.

5.4 Invariant-mass cuts

Some cuts on the invariant mass can be set in some situations, via

```
set minm{tag} = {value}
```

where the tag now can refer to lepton-lepton, lepton-quark, photon-fermion, quark-quark in the $p-p(\bar{p})$ case, quark-b in the $p-p(\bar{p})$ case and b-b in the $p-p(\bar{p})$ case:

tag = `[ll, lq, pf, qqp, qb, bb]`

and `value` (real) is the value of the minimum invariant-mass chosen.

5.5 e^+e^- cuts

In the case of e^+e^- collisions (`set colpar = 3`), we can set cuts on energy, angle, and decay related cuts. The former ones are:

- *cutoff* (real): cutoff in the e^+e^- case
- *minenl*, *minenq*, *minenp* (real): minimum energy of lepton, quark and photon respectively
- the cut on the minimum angle between different pair combination can be set as:

  `set minang{tag} = {value}`

  where tag refers to lepton and beam, quark and beam, photon and beam, two leptons, lepton and quark, two quarks, photon and fermion:

  `tag` = `[lb, qb, pb, ll, lq, qq, pf]`

  and the value is expressed in degrees.
- *minmqqe* (real): represents the minimum mass of quark/gluon with quark/gluon

The decay cuts that can be imposed are:

- *decay_minptl* (real): the minimum P_T of the lepton from the decay
- *decay_minycl*, *decay_maxycl* (real): minimum and maximum y rapidity of the lepton from the decay
- *decay_maxrapl* (real): the maximum pseudorapidity of the lepton from the decay
- *decay_minel* (real): the minimum energy of the lepton from the decay
- *decay_maxcl* (real): the maximum $\cos\theta$ between the beam and the lepton from the decay

6 Quarkonium specific parameters

- *exp3pjQ* (bool): is a flag that determines whether summing over (F) $3PJ$ ($J = 0, 1, 2$) or not (T). If `set exp3pjQ = T` returns $3PJ$ states seperately.
- *modes* (integer): determines whether the calculated result is the polarized one (1) or not (0). In the polarized case (`set modes = 1`), the user should also supply the values of *SDME1* and *SDME2*, first and second index for the spin density matrix element of the first quarkonium, in order to let the program know which SDME to calculate. Meanwhile, the value of *LSJ* represents which "spin" (S, L or J) in quarkonium should be specified. The user should also specify the polarization frame (*PolarFrame*, integer). The possible choices for the frames are:
  - `set PolarFrame = 1`: helicity frame
  - `set PolarFrame = 2`: Collins-Soper frame
  - `set PolarFrame = 3`: Gottfried-Jackson frame
  - `set PolarFrame = 4`: target frame
- *muNRQCD* (real): the renormalization scale for Non-Relativistic QCD

6.1 LDMEs

Since HELAC-Onia is based on NRQCD, it uses long distance matrix elements (LDMEs) to calculate cross section. Differently from usual definition of LDMEs (e.g. the ones in Ref. [4]), they are rescaled. The standard LDMEs for Color Singlet are

$$\begin{aligned}\langle\mathcal{O}\left({}^{(2S+1)}S_J^{[1]}\right)\rangle &= (2J+1)2N_C\frac{|R(0)|^2}{4\pi},\\ \langle\mathcal{O}\left({}^{(2S+1)}P_J^{[1]}\right)\rangle &= (2J+1)2N_C\frac{3\,|R'(0)|^2}{4\pi},\end{aligned} \tag{2}$$

while in HELAC-Onia the following are used:

$$\begin{aligned}\langle\mathcal{O}\left({}^{(2S+1)}S_J^{[1]}\right)\rangle &= \frac{|R(0)|^2}{4\pi},\\ \langle\mathcal{O}\left({}^{(3)}P_0^{[1]}\right)\rangle = \langle\mathcal{O}\left({}^{(3)}P_1^{[1]}\right)\rangle = \langle\mathcal{O}\left({}^{(3)}P_2^{[1]}\right)\rangle &= \frac{3\,|R'(0)|^2}{4\pi}.\end{aligned} \tag{3}$$

Moreover, the Color Octet LDMEs in HELAC-Onia are scaled in the following way:

$$\text{CO LDME}|_{\text{HO}} = \frac{\text{CO LDME}|_{\text{Literature}}}{(N_C^2-1)\,(2J+1)}. \tag{4}$$

We can set a different value for all the LDMEs via

```
set LDME{system}{Fock state} = {value}
```

where `system` = `[cc, bb, bc]` refers to charmonia, bottomonia and B_c systems, while the possible Fock states are:

`Fock state` = `[1S01, 3S01, 1S11, 3P01, 3P11, 3P21, 1S08, 3S08, 1S18, 3P08, 3P18, 3P28]`

and `value` is the value associated to the LDME.

7 User output

- *topdrawer_output*, *gnuplot_output*, *root_output*, *hwu_output* (bool): plot flags to let HELAC-Onia plot histograms and output into Topdrawer, Gnuplot, ROOT and HwU files.
- *unwgt* (bool): flag to get Les Houches Events (i.e. `.lhe`) samples for any single partonic process.
- *preunw, unwevt* (integers): when `gener = 0` and `unwgt = T`, these parameters controls the number of pre-unweighted events and the number of unweighted events in `.lhe` files.

Note that, at the moment, as results we provide the histograms for the total cross section, and the y and P_T spectrum for each particle in the final state of the desired reaction.

8 Physical constants in HELAC-Onia

It is possible to modify some of the physical constants in HELAC-Onia. As usual, it should be used the syntax `set {parameter} = {value}`. In the following, we list the constants with the default values (all the masses are indicated in GeV):

- Electroweak sector:
  - Fermi coupling constant G_F: `gfermi = 1.16639d-5`
  - M_Z: `zmass = 91.188d0`
  - Γ_Z: `zwidth = 2.446d0`
  - M_W: `wmass = 80.419d0`
  - Γ_W: `wwidth = 2.048d0`
  - $\sin^2\theta_W$: if `set sin2thetaw -1`, we are setting $\sin^2\theta_W = 1-(\frac{M_W}{M_Z})^2$
  - α_{em}: if `set alphaem -1`, we are setting $\alpha_{\text{em}} = \sqrt{2}G_F M_W^2 \frac{\sin^2\theta_W}{\pi}$
  - M_H: `higmass = 126.d0`
  - Γ_H: `higwidth = 4.291d-3`
  - m_e: `emass 0.0d0`; m_μ: `mumass 0.0d0`; m_τ: `taumass 0.0d0`;
  - m_{ν_e}: `nemass = 0.0d0`; m_{ν_μ}: `nmumass = 0.0d0`; m_{ν_τ}: `ntaumass = 0.0d0`;
- Quarks and strong interaction:
  - α_S not running value (used if `alphasrun = 0`): `alphas2 = 0.118d0`
  - m_u: `umass = 0.0d0`; m_d: `dmass = 0.0d0`; m_s: `smass = 0.0d0`;
  - m_c: `cmass = 0.0d0`; m_b: `bmass = 0.0d0`;
  - m_t: `tmass = 174.3d0`; Γ_t: `twidth = 1.6d0`;

9 Standard Model particles and quarkonia in HELAC-Onia

In the following tables the symbols used for Standard Model particles and quarkonia in HELAC-Onia are listed.

Particles	Particles symbols
ν_e, e^-, u, d, ν_μ, μ^-, c, s, ν_τ, τ^-, t, b	ve, e-, u, d, vm, m-, c, s, vt, tt-, t, b
$\bar{\nu}_e$, e^+, $\bar{u}$, $\bar{d}$, $\bar{\nu}_\mu$, μ^+, $\bar{c}$, $\bar{s}$, $\bar{\nu}_\tau$, τ^+, $\bar{t}$, $\bar{b}$	ve~, e+, u~, d~, vm~, m+, c~, s~, vt~, tt+, t~, b~
γ, Z, W^+, W^-, g	a, z, w+, w-, g
H, χ, Φ^+, Φ^-	h, g0, g+, g-

Table 1: Syntax for Standard Model particles in HELAC-Onia.

Particle	Particle symbol	Particle	Particle symbol
$c\bar{c}\left[{}^1S_0^{[1]}\right]$	cc~(1S01)	$b\bar{b}\left[{}^1S_0^{[1]}\right]$	bb~(1S01)
$c\bar{c}\left[{}^1S_0^{[8]}\right]$	cc~(1S08)	$b\bar{b}\left[{}^1S_0^{[8]}\right]$	bb~(1S08)
$c\bar{c}\left[{}^3S_0^{[1]}\right]$	cc~(3S11)	$b\bar{b}\left[{}^3S_0^{[1]}\right]$	bb~(3S11)
$c\bar{c}\left[{}^3S_1^{[8]}\right]$	cc~(3S18)	$b\bar{b}\left[{}^3S_1^{[8]}\right]$	bb~(3S18)
$c\bar{c}\left[{}^1P_1^{[1]}\right]$	cc~(3P11)	$b\bar{b}\left[{}^1P_1^{[1]}\right]$	bb~(3P11)
$c\bar{c}\left[{}^1P_1^{[8]}\right]$	cc~(3P18)	$b\bar{b}\left[{}^1P_1^{[8]}\right]$	bb~(3P18)
$c\bar{c}\left[{}^1P_{J=0,1,2}^{[1]}\right]$	cc~(3PJ1)	$b\bar{b}\left[{}^1P_{J=0,1,2}^{[1]}\right]$	bb~(3PJ1)
$c\bar{c}\left[{}^1P_{J=0,1,2}^{[8]}\right]$	cc~(3PJ8)	$b\bar{b}\left[{}^1P_{J=0,1,2}^{[8]}\right]$	bb~(3PJ8)

Table 2: Syntax for charmonia and bottomonia in various Fock states in HELAC-Onia.

Particle	Particle symbol	Particle	Particle symbol
$c\bar{b}[{}^1S_0^{[1]}]$	cb~(1S01)	$b\bar{c}[{}^1S_0^{[1]}]$	bc~(1S01)
$c\bar{b}[{}^1S_0^{[8]}]$	cb~(1S08)	$b\bar{c}[{}^1S_0^{[8]}]$	bc~(1S08)
$c\bar{b}[{}^3S_0^{[1]}]$	cb~(3S11)	$b\bar{c}[{}^3S_0^{[1]}]$	bc~(3S11)
$c\bar{b}[{}^3S_1^{[8]}]$	cb~(3S18)	$b\bar{c}[{}^3S_1^{[8]}]$	bc~(3S18)
$c\bar{b}[{}^1P_1^{[1]}]$	cb~(3P11)	$b\bar{c}[{}^1P_1^{[1]}]$	bc~(3P11)
$c\bar{b}[{}^1P_1^{[8]}]$	cb~(3P18)	$b\bar{c}[{}^1P_1^{[8]}]$	bc~(3P18)
$c\bar{b}[{}^1P_{J=0,1,2}^{[1]}]$	cb~(3PJ1)	$b\bar{c}[{}^1P_{J=0,1,2}^{[1]}]$	bc~(3PJ1)
$c\bar{b}[{}^1P_{J=0,1,2}^{[8]}]$	cb~(3PJ8)	$b\bar{c}[{}^1P_{J=0,1,2}^{[8]}]$	bc~(3PJ8)

Table 3: Syntax for mixed flavour quarkonium $B_c^\pm$ in various Fock states in HELAC-Onia.

References

[1] H. S. Shao, Comput. Phys. Commun. **184** (2013), 2562-2570 doi:10.1016/j.cpc.2013.05.023 [arXiv:1212.5293 [hep-ph]].

[2] H. S. Shao, Comput. Phys. Commun. **198** (2016), 238-259 doi:10.1016/j.cpc.2015.09.011 [arXiv:1507.03435 [hep-ph]].

[3] A. Buckley, J. Ferrando, S. Lloyd, K. Nordström, B. Page, M. Rüfenacht, M. Schönherr and G. Watt, Eur. Phys. J. C **75** (2015), 132 doi:10.1140/epjc/s10052-015-3318-8 [arXiv:1412.7420 [hep-ph]].

[4] P. Artoisenet, F. Maltoni and T. Stelzer, JHEP **02** (2008), 102 doi:10.1088/1126-6708/2008/02/102 [arXiv:0712.2770 [hep-ph]].