Keywords (I)

- `InterfaceVersion`: can take the values `BLHA1` or `BLHA2`.
- `Model`: `SM, SMdiag, MSSM`.
  For BSM standards, the `UFO` format has been proposed.
- `MassiveParticles`: defines a list of massive particles at the level of the order file/code generation, e.g.
  `MassiveParticles 5 6`
- `LightMassiveParticles`: useful if mass regularisation instead of dim.reg. is used (EW), defines set of particles where only $\log(m)$ terms are kept, but not $\mathcal{O}(m)$ terms
- `ExcludedParticles`: exclude particles which are contained by default in `Model`.
- `AlphasPower`: integer which specifies the α_s power of the cross section (can also be different for individual sub-processes)
- `AlphaPower`: specifies the α power of the Born cross section, default is zero.

Keywords (II)

- `CorrectionType`: standard keywords are `QCD`, `EW`, `QED`.
- `IRregularisation`: possible choices for QCD are `CDR`, `DRED`.
- `TreatUnstable`: defines how to treat unstable phase space points. Possible values are `discard, sloppy, debug`
- `Precision`: precision threshold dividing points into "stable" or "unstable". Should be a relative accuracy, resulting from OLP internal stability checks
- `AmplitudeType:` could take the values `LoopInduced, Tree, ColorCorrelated, SpinCorrelated`
- `Extra`: can be used to write special requirements relevant to the OLP into the order file. The MC will either ignore this option altogether, or ignore it but write it into the order file, or need it and have it included in the MC setup

Keywords (III)

- `SubdivideSubprocess`: useful if a given process is composed of several components

  may be useful to return colour/spin correlated matrix elements, to be discussed
- `EWScheme`: can be flagged by the keywords `alpha0`, `alphaMZ`, `alphaGF`, `alphaRUN`, `alphaMSbar`, `UserDefined` (default).
- `MassiveParticleScheme`: standard choice is `OnShell`.
- `WidthScheme`: defines the treatment of unstable particles. Standard values are `ComplexMass`, `FixedWidth`, `RunningWidth`, `PoleApprox`.
- `PolvecsAll`: specify polarisation vectors for all particles.
- `PolvecsMVB`: specify polarisation vectors for (polarized) massive vector bosons only, while for fermions specify reference vectors for light-cone decomposition.

keywords which probably will be retired

- `MatrixElementSquareType`: was intended to distinguish colour (C) and helicity (H) treatment. Possible values were defined as `CHsummed`, `Csummed`, `Hsummed`, `NOTsummed`. Could declare `CHsummed` as default. The flag `Extra` could be used to accommodate for the keywords `HelAvgInitial`, `ColAvgInitial`, `MCSymmetrizeFinal`.
- `ModelFile`: Keyword replaced by `Model`.
- `OperationMode`: optional flag, was intended to specify OLP defined approximations to the one-loop contribution, e.g. `LeadingColour`, `HighEnergyLimit`, etc.

  used in the mode `CouplingsStrippedOff` so far. `CouplingsStrippedOff` can be ambiguous in the presence of EW couplings
- `ResonanceTreatment`: has been replaced by `WidthScheme`, as e.g. the complex mass scheme also concerns non-resonant propagators.

Functions: initialisation phase

- `OLP_Start(char*,&int)`: `char:` name of the contract file, `int:` is set to 1 if contract file is accepted
- `OLP_Info(char[15] olp_name, char[15] olp_version, char[255] message)`: name of the OLP, version, `message:` info about the relevant publications
- `OLP_SetParameter(const char*,const &double,const &double,&int)`: (in initialisation phase: for static parameters)
  `char:` string serving as a keyword for the parameter to be set two double precision numbers for the parameter value (can be complex); suggestion: pointer to an array instead of two doubles, would allow e.g. passing of CKM matrix elements
  `int:` is set by the OLP to tell the MC whether the stetting of the parameter was successful

Functions: initialisation phase

- `int:`
  `int=1` parameter has been set successfully,
  `int=0` failure: issue an error message and stop,
  `int=2` the parameter is unknown or the setting is ignored, but the program should proceed
- `OLP_PrintParameter(const char*,const &double,const &double)`:
  info on parameter settings for the user

Functions: runtime

- `OLP_SetParameter(const char*,const &double,const &double,&int)`
  called at runtime for dynamical parameters
- `OLP_EvalSubProcess(const int* i, const double* pp, const double* mu, double* rval, int* status)`
  - const int* i: one element array with the label of the subprocess
  - const double* pp: array of momenta, conventions $(E_j, k_j^x, k_j^y, k_j^z, M_j)$ unchanged
  - const double* mu: one element array with renormalisation scale
  - double* rval: array of return values
  - int* status: one element array denoting the status of the accuracy check.

  previously: contained both μ_r and $\alpha_s(\mu_r)$
  However, $\alpha_s(\mu_r)$ can now be set using `OLP_SetParameter`