# WHiZard/O'Mega Tutorial

Jürgen Reuter

Carleton University, Ottawa $\longrightarrow$ University of Freiburg

W. Kilian, T. Ohl, JR, 1998-20xx, always in progress

MSU, East Lansing/Carleton, Ottawa, March 16/19, 2007

# The WHiZard/O'Mega Generator Generator

SM/BSM at "future" colliders

- high-multiplicity multi-particle final states
- interferences signal/signal, signal/background
- complicated collider/detector environments
- $2 \to 2$ wrong, even at LHC
  (therefore: Pythia, Herwig, Isajet strongly non-recommended)

Level of Complexity:

- $e^+e^- \to ZHH \to ZWWWW \to bbjjjjjjjj$ (12,000,000 diagrams)
- $pp \to \tilde{\chi}_1^0\tilde{\chi}_1^0 bbbb$ (32,000 diagrams, 22 color flows, $\sim 10,000$ PS channels)
- $pp \to VVjj \to jj\ell\ell\nu\nu$ incl. anomalous TGC/QGC

Current versions: WHiZard 1.51 / O'Mega 000.011beta

`http://theorie.physik.uni-wuerzburg.de/~ohl/omega/`

`http://www-ttp.physik.uni-karlsruhe.de/whizard/`

Actual are (will be) available from my homepage:

`http://www.physics.carleton.ca/~reuter`

Major upgrade this spring/summer:

WHiZard 2.0 / O'Mega 1.0

# Prerequisites for WHiZard/O'Mega

Standard GNU tools (like `make`, `sed`, `grep` etc.)

`O'Mega`

- `O'Caml` (Objective Caml) programming language interpreter/compiler (Version $\geq$ 3.04)
- `Fortran 90/95` compiler
  Intel 8.x + 9.x, NAG, Lahey, still problematic: Portland pgf, g95

`Perl5` for glueing scripts (might become obsolete in future)

`Fortran77` for interoperation with old libraries (CERNLIB)

**Optional:**

- STDHEP library
- CERNLIB for PDFs and Pythia hadronization
- LHAPDF library for PDFs
- $\LaTeX$ and MetaPost for on-line generation of histograms and plots
- Non-hostile to friendly contact to the authors

# O'Mega installation

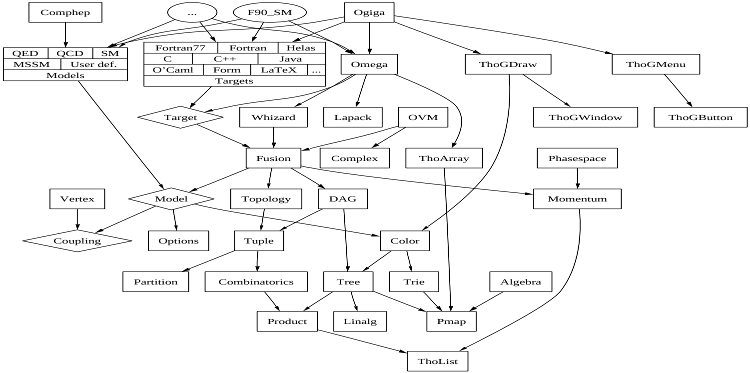
## O'Caml installation

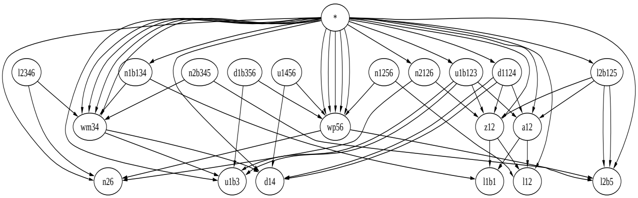
- Contained in most Linux distribution (ask your SysAdmin)
- available from `http://pauillac.inria.fr` (3.09.3)
- precompiled binaries for most OS (even It-which-must-not-be-named)
- otherwise do: `./configure --prefix make world.opt`, `umask 022; make install`

## O'Mega installation

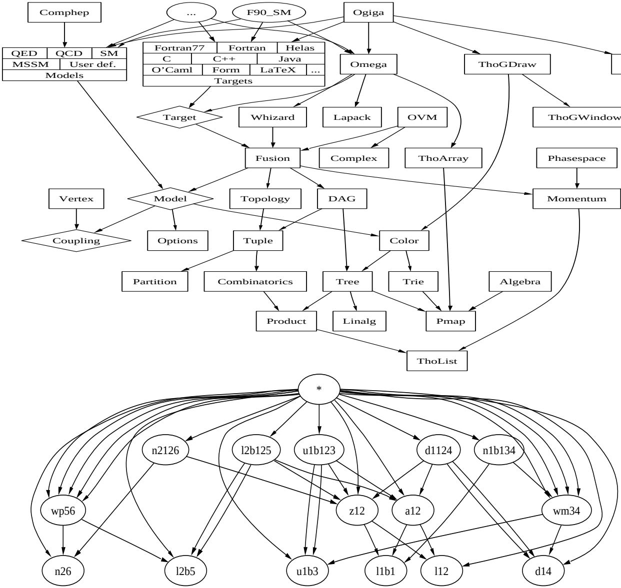
- unpack the O'Mega tarball with `tar --xzf`
- From the main directory do `./configure`
- Create the binaries by `make bin opt`
- Compile the libraries by `make f95`
- Binaries (e.g. `f90_SM.opt`) and testbeds (e.g. `test_omega95`) are located in e.g. `arch/i686-suse-linux/bin`

# How O'Mega works – The Dark Side of the Black Box



# How O'Mega works – The Dark Side of the Black Box



# O'Mega Usage

- In `web/` subdir.: Documented source code (`omega.ps`, 1100 pages)
- Short online manual:
  `http://theorie.physik.uni-wuerzburg.de/~ohl/omega/`
- Just do: `f90_MyModel.opt -scatter "e+ e- -> mu+ mu-"`

```
usage: ./f90_SM.opt [options]
[e-|nue|u|d|e+|nuebar|ubar|dbar|mu-|numu|c|s|mu+|numubar|cbar|sbar|tau-|nutau|t|b|tau+|nutaubar|
 tbar|bbar|A|Z|W+|W-|g|H|phi+|phi-|phi0|gx]

-target:function function name
-target:90 don't use Fortran95 features that are not in Fortran90
-target:....................
-model:constant_width use constant width (also in t-channel)
-model:fudged_width use fudge factor for charge particle width
-model:.....................
-warning: check arguments and print warning on error
-error: check arguments and terminate on error
-...............................
-scatter in1 in2 -> out1 out2 ...
-scatter_list proc1 ;; proc2 ;; proc3 ;;  ...
-decay in -> out1 out2 ...
-decay_list in -> out1 ;; in -> out2 ;; in -> out3
-cascade select certain cascade channels
-forest symbolic output of all Feynman diagrams
-feynmf LaTeX output of all Feynman diagrams (for feynmp.sty)
-revision print revision control information
-summary print only a summary
-poles print the Monte Carlo poles
-dag print minimal DAG
-full_dag print complete DAG
-help  Display this list of options
--help  Display this list of options
```

# WHiZard installation

- Create a directory `[whizard_dir]` and unpack the WHiZard tarball therein
- Edit the file `config.site` and insert the locations of

  `OMEGA_DIR=$ HOME/[omega_dir]`

  `CERNLIB_DIR=[cernlib_dir]` (optional)

  LHAPDF will be available in release 2.0

  `STDHEP_DIR=[stdhep_dir]` (optional)
- from the top directory do:

  `./configure`

  `—disable-mad —disable-chep`, since (old) Madgraph and Comphep mainly for internal sanity checks

  will be standard in new version anyhow

  `make man`: creates the manual `manual.ps` in `doc/`

  `make doc`: creates docu. source code `whizard.ps` (if `noweave` present)
- Now, you can already specify a process in file `conf/whizard.prc`
- from the top directory do:

  `make prg install`

# `configure` status

```
...................
config.status: creating Makefile
config.status: executing default-1 commands
config.status: executing default-2 commands
config.status: executing default-3 commands

--- Configure summary: ---
--- Enabled features: ---
CompHEP    (Matrix elements)     chep-src/setup/
Madgraph   (Matrix elements)     mad-src/
O'Mega     (Matrix elements)     /home/jr/Physik/progs/omega/omega-2006
Autoconf   (Restricted bundle)   autoconf
CIRCE      (Beamstrahlung)       circe-src/
CIRCE2     (Beamstrahlung)       circe2-src/
PDFLIB     (Structure functions) /usr/local/cernlib/pro/lib/libpdflib80
PYTHIA     (Fragmentation)       /usr/local/cernlib/pro/lib/libpythia62
LaTeX/Metapost  (Histograms)     /usr/bin/mpost

--- Disabled or absent features: ---
STDHEP     (Binary event files)

--- Configuration complete. --
```

# Models currently supported by `WHiZard/O'Mega`

| Model type | with CKM matrix | trivial CKM |
|---|---|---|
| QED with $e, \mu, \tau, \gamma$ | – | `QED` |
| QCD with $d, u, s, c, b, t, g$ | – | `QCD` |
| Standard Model | `SM_CKM` | `SM` |
| SM with anomalous couplings | `SM_ac_CKM` | `SM_ac` |
| MSSM | `MSSM_CKM` | `MSSM` |
| Littlest Higgs | — | `Littlest` |
| Littlest Higgs with ungauged $U(1)$ | — | `Littlest_Eta` |
| Simplest Little Higgs (anomaly-free) | — | `Simplest` |
| Simplest Little Higgs (universal) | — | `Simplest_univ` |
| SM with spin-2 graviton | — | `Xdim` |
| SM with gravitino and photino | — | `GravTest` |
| Augmentable SM template | — | `Template` |

New version: LH with $T$ parity, NMSSM, SUSY exotics like ESSM/PSSSM, NCSM, UED, maybe NLO SUSY stuff

# Process file: `conf/whizard.prc`

**Model selection:** we already had that

**Particles names:** look in `conf/whizard.prc.XXX` or `conf/models/Model.mdl`

`test`: Constant matrix elements for testing, e.g. structure function

```
# WHIZARD configuration file

# The selected model
model   SM

alias q u:d:s:c
alias Q U:D:S:C

# Processes
#  Methods: ....., omega=O'Mega, test=trivial)
#  Options: r      restricted intermediate state (O'Mega)
#           c      apply exact color algebra (O'Mega)
#           w:XXX  width scheme (O'Mega)
#           u      unit matrix element (test)
#
# Tag            In       Out              Method  Option
#======================================================
em               e1,E1    e2,E2            omega
em_test          e1,E1    e2,E2            test
em_test          e1,E1    e2,E2            test    u
eeqq             e1,E1    u:d:s,U:D:S      omega
uudd_nc          u,U      d,D              omega
uudd             u,U      d,D              omega   c
qqwz             q,Q      W,Z              omega
ee_z_only        e1,E1    e1,E1            omega   r:3+4~Z
```

Options available:

- `u (test)`: unit matrix element
- `c (omega)`: exact color amplitudes and full color flow information
- `r: (omega)`: restricted diagrams (cascading)
- `w: (omega)`: different width schemes (fudge, fixed, . . .

After changes in `whizard.prc` do `make prg install`

# Input file: `results/whizard.in` (`NAMELIST` fmt)

Let's just do a simple example (better than audience buffer overflow): $e^+e^- \to \mu^-\bar{\nu}_\mu u\bar{d}$

- `conf/whizard.prc`:

```
# The selected model
model    SM
# Tag    In        Out           Method   Option
#==========================================
cc10     e1,E1     e2,N2,u,D     omega
```

- `make prg install`
- edit `results/whizard.in`

```
&process_input
process_id = "cc10"
sqrts = 500
/
&integration_input
/
&simulation_input
/
&diagnostics_input
/
&parameter_input
Mmu = 0
/
&beam_input
/
&beam_input
/
```

- `whizard` executable in `results` is MC generator for process cc10, to run do
  - a) Execute `./whizard` in `results` with command line options (cf. `./whizard --help`)
  - b) `make run`, either in `results` or the top directory

```
! WHIZARD 1.51 (Jun 15 2005)
! Reading process data from file whizard.in
! Wrote whizard.out
!
! Process cc10:
!    e a-e ->  mu a-nu_mu   u a-d
!   32  16 ->   1        2   4   8
! Process energy set to     500.00     GeV
! Reading vertices from file whizard.mdl ...
! Model file:        54 trilinear vertices found.
! Model file:        54 vertices usable for phase space setup.
! Generating phase space channels for process cc10...
! Phase space:        8 phase space channels generated.
! Scanning phase space channels for equivalences ...
! Phase space:        8 equivalence relations found.
! Note: This cross section may be infinite without cuts.
! Wrote default cut configuration file whizard.cc10.cut0
! Wrote phase space configurations to file whizard.phx
!
! Created grids:      8 channels,  8 dimensions with 20 bins
```

```
! WHIZARD run for process cc10:
!=============================================================================
! It      Calls  Integral[fb]  Error[fb]   Err[%]    Acc  Eff[%]   Chi2 N[It]
!-----------------------------------------------------------------------------
! Reading cut configuration data from file whizard.cut1
! No cut data found for process cc10
! Using default cuts.
cut M of   12      within  1.00000E+01  1.00000E+99
! Preparing (fixed weights):   1 sample of      20000 calls ...
   1      20000  2.6806323E+02  1.01E+01    3.76    5.31*  1.66    0.00   1
!-----------------------------------------------------------------------------
! Adapting (variable wgts.):  10 samples of      20000 calls ...
   2      20000  2.7592027E+02  1.05E+01    3.81    5.38   1.50
   3      20000  2.7127725E+02  1.96E+00    0.72    1.02* 10.69
   4      20000  2.7123539E+02  1.51E+00    0.56    0.79* 11.73
   5      20000  2.7016999E+02  1.36E+00    0.50    0.71* 15.36
   6      20000  2.7204042E+02  1.32E+00    0.49    0.69* 16.44
   7      20000  2.7265921E+02  1.30E+00    0.47    0.67* 16.40
   8      20000  2.7105262E+02  1.28E+00    0.47    0.67* 13.70
   9      20000  2.7154268E+02  1.27E+00    0.47    0.66* 15.15
  10      20000  2.7265788E+02  1.33E+00    0.49    0.69  11.91
  11      20000  2.7105441E+02  1.32E+00    0.49    0.69  12.45
!-----------------------------------------------------------------------------
! Integrating (fixed wgts.):   3 samples of      20000 calls ...
  12      60000  2.7196199E+02  7.48E-01    0.27    0.67  10.72    1.39   3
!-----------------------------------------------------------------------------
!
! Time estimate for generating 10000 unweighted events:    0h 00m 03s
!=============================================================================
! Summary (all processes):
!-----------------------------------------------------------------------------
! Process ID     Integral[fb]  Error[fb]   Err[%]        Frac[%]
!-----------------------------------------------------------------------------
  cc10           2.7196199E+02  7.48E-01    0.27         100.00
!-----------------------------------------------------------------------------
  sum            2.7196199E+02  7.48E-01    0.27         100.00
!=============================================================================
! Wrote whizard.out
! Integration complete.
! No event generation requested
! WHIZARD run finished.
```

# Further steps: Event Generation

whizard.in:

```
&process_input
process_id = "cc10"
sqrts = 500
luminosity = 10
/
&integration_input
read_grids = T
/
&simulation_input /
&diagnostics_input /
&parameter_input
Mmu = 0
/
&beam_input /
&beam_input /
```

Screen output:

```
! Using grids and results from file:
! Reading analysis configuration data from file whizard.cut5
! No analysis data found for process cc10
! Event sample corresponds to luminosity [fb-1] =   9.999
! Event sample corresponds to       22665  weighted events
! Generating       2717 unweighted events ...
!=============================================================================
! Analysis results for process cc10:
! It      Events Integral[fb]  Error[fb]   Err[%]    Acc  Eff[%]   Chi2 N[It]
!-----------------------------------------------------------------------------
  13       2717  2.7173259E+02  5.21E+00    1.92    1.00 100.00
!-----------------------------------------------------------------------------
! Warning: Excess events:    1.2        (    0.04% )  | Maximal weight:  1.04
! There were no errors and  2 warning(s).
! WHIZARD run finished.
```

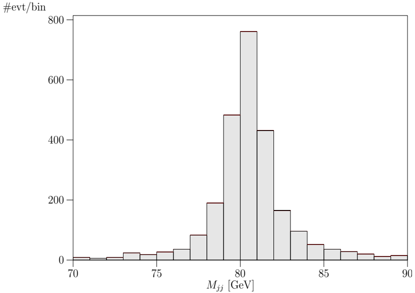
# Further steps: Event Generation

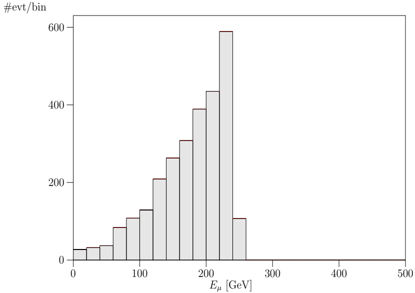
whizard.in:

```
&process_input
process_id = "cc10"
sqrts = 500
luminosity = 10
/
&integration_input
read_grids = T
/
&simulation_input /
&diagnostics_input /
&parameter_input
Mmu = 0
/
&beam_input /
&beam_input /
```

Screen output:

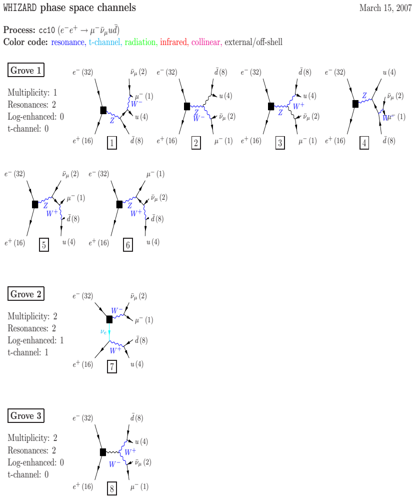
```
! Using grids and results from file:
! Reading analysis configuration data from file whizard.cut5
! No analysis data found for process cc10
! Event sample corresponds to luminosity [fb-1] =   9.999
! Event sample corresponds to       22665  weighted events
! Generating       2717 unweighted events ...
!=============================================================================
! Analysis results for process cc10:
! It       Events Integral[fb]  Error[fb]   Err[%]    Acc  Eff[%]   Chi2 N[It]
!-----------------------------------------------------------------------------
  13        2717  2.7173259E+02  5.21E+00    1.92    1.00 100.00
!-----------------------------------------------------------------------------
! Warning: Excess events:    1.2        (    0.04% )  | Maximal weight:  1.04
! There were no errors and  2 warning(s).
! WHIZARD run finished.
```



# Useful: `results/make graphs`

make graphs in `results/` produces most important Feynman diagrams with phase space information:



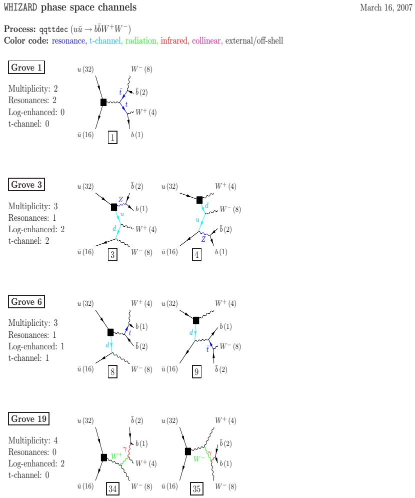
WHIZARD phase space channels — March 15, 2007

**Process:** cc10 ($e^-e^+ \to \mu^-\bar{\nu}_\mu u\bar{d}$)
**Color code:** resonance, t-channel, radiation, infrared, collinear, external/off-shell

**Grove 1**
Multiplicity: 1
Resonances: 2
Log-enhanced: 0
t-channel: 0

**Grove 2**
Multiplicity: 2
Resonances: 2
Log-enhanced: 1
t-channel: 1

**Grove 3**
Multiplicity: 2
Resonances: 2
Log-enhanced: 0
t-channel: 0



WHIZARD phase space channels — March 16, 2007

**Process:** qqttdec ($u\bar{u} \to b\bar{b}W^+W^-$)
**Color code:** resonance, t-channel, radiation, infrared, collinear, external/off-shell

**Grove 1**
Multiplicity: 2
Resonances: 2
Log-enhanced: 0
t-channel: 0

**Grove 3**
Multiplicity: 3
Resonances: 1
Log-enhanced: 2
t-channel: 2

**Grove 6**
Multiplicity: 3
Resonances: 1
Log-enhanced: 1
t-channel: 1

**Grove 19**
Multiplicity: 4
Resonances: 0
Log-enhanced: 2
t-channel: 0

# Input blocks in `results/whizard.in`

**&process_input**

```
process_id = ""            ! Process Name

sqrts = 0                  ! CM energy
luminosity = 0             ! [fb^-1]

structured_beams = F       ! structur f.

input_file = ""            ! appends .in
input_slha_format = F       ! SLHA format
```

**&integration_input**

```
calls = 1 10000 3 10000 ! process dep.

read_grids = F             ! avoids adap.
```

**&simulation input**

```
n_events = 0               ! N_unweighted
n_calls = 0                ! ME calls (weighted)

unweighted = T             ! unweighted samples

write_events = F           ! whizard.evt (ext. use)
write_events_format = 1    ! Event format
several_file_opt. = ""     ! user-def. files
write_events_raw = T       ! whizard.evx
```

**&diagnostics_input**

**&parameters_input**

```
depends on used model,      e.g. gg = 1.218 ! g_s
```

**&beam_input**

```
polarization = 0 0 0       ! fraction of pols.
```

# Input blocks in results/whizard.in

**&process_input**

```
process_id = ""              ! Process Name
cm_frame = T                 ! CM frame
sqrts = 0                    ! CM energy
luminosity = 0               ! [fb^-1]
polarized_beams = F          ! Polarization
structured_beams = F         ! structur f.
beam_recoil = F              ! Beam recoil
recoil_cons._mom. = F        ! p, not E cons.
filename = ""                ! instead of W
directory = ""               ! dir. for I/O
input_file = ""              ! appends .in
input_slha_format = F         ! SLHA format
```

**&integration_input**

```
calls = 1 10000 3 10000 ! process dep.
seed = undef.                ! random seed
reset_seed_each_process = F
accuracy_goal = 0            ! stops grid adap.
efficiency_goal = 100        ! stops grid adap.
time_limit_adaption = 0 ! time limit
stratified = T               ! srat. vs. import.
use_efficiency = F           ! eff. vs. acc.
weights_power = 0.25         ! Channel adap.
min_bins = 3                 ! bins per dim.
max_bins = 20                ! bins per dim.
min_calls_per_bin = 10       ! calls per bin
min_calls_per_channel = 0
write_grids = T              ! Grid output
write_grids_raw = F          ! grid raw fmt
write_all_grids = F          ! all grid adap.
read_grids = F               ! avoids adap.
read_grids_raw = F           !
read_grids_force = F         ! forces reading
generate_phase_space = T ! whizard.phx
read_phase_space = T         ! read whizard.phs
several_file_opt. = ""       ! user-def. files
phase_space_only = F         ! stop after PS gen.
use_equivalences = T         ! use permut. symm.
azimuthal_dependece = F ! no azimuth info.
phase_space_setup_opt.       ! detailed PS setup
```

**&simulation input**

```
n_events = 0                 ! N_unweighted
n_calls = 0                  ! ME calls (weighted)
N_events_warmup = 0          ! xtra warmup events
unweighted = T               ! unweighted samples
normalize_weight = T         ! norm. to 1 vs. sigma
write_weights = F            ! write weights
write_events = F             ! whizard.evt (ext. use)
write_events_format = 1 ! Event format
several_file_opt. = ""       ! user-def. files
write_events_raw = T         ! whizard.evx
read_events[_force] = F ! read whizard.evx
keep_xxx = F                 ! record remnants/beam
guess_color_flow = F         ! for non-ex. flow info
recalculate = F              ! reeval. ME
fragment = F                 ! fragmentation on/off
fragmentation_method = 0 ! PYTHIA/Jetset
user_fragmentation_method = 0
pythia_parameters = ""       ! PYTHIA frag. params.
```

**&diagnostics_input**

```
chattiness = 4               ! Message level
catch_signals = T            ! catch ext. sign.
time_limit = F               ! see manual
warn_empty_channel = F       !    dto.
screen_xxx = F               ! show on screen
show_pythia_xxx = T          ! Pythia output
write_logfile = T            ! whizard.xxx.out
show_input = T               ! see manual
show_results = T             ! integr. results
show_phase_space = F         ! PS config.
show_cuts = T                ! cuts in log file
show_histories = F           ! detailed VAMP history
show_history = T             ! VAMP summary
```

**&parameters_input**

```
depends on used model,      e.g. gg = 1.218 ! g_s
```

**&beam_input**

```
energy = 0                   ! E of beam
angle = 0                    ! angle of beams
direction = 0 0 0            ! beam direction in LAB
vector_polarization = F ! long./transv. vs. hel.
polarization = 0 0 0         ! fraction of pols.
```

# Output files, Cuts and Histograms

When you run a process, then the following files are written in `results/`:

- `whizard.out`: generic output summary
- `whizard.XXX.out`: process specific output
- `whizard.XXX.grc`, `whizard.XXX.grb[grc]`: best [current] grid for process XXX
- `whizard.phx`: phase space for current process

```
! e-   e+   ->  e-   e+   gamma
! 16    8        1    2     4
process eeg
  cut Q of 10 within -99999 -1
  cut Q of 17 within -99999 -1
  cut M of  3 within 10 99999
  cut E of  4 within  5 99999
  cut PT of 4 within 19 99999
  cut THETA(DEG) of  4 1 within 5 180
  cut THETA(DEG) of  4 2 within 5 180
```

## Cuts and Histograms:

- File `results/whizard.cut1` Real kinematic cuts, taken into account for phase space int.
- File `results/whizard.cut5` Cuts for histogramming, declaration of desired histograms
- **Events needed for plots!**
- `make plots` produces `whizard-plots.ps`

```
! e-   e+   ->  e-   e+   gamma
! 16    8        1    2     4
process eeg
  cut M of 3 within  80 100
and
  cut M  of 3 within 180 200
  cut PT of 4 within 100 99999
and
  cut E of 4 within 0 100
  histogram PT of 1 within 0 500
histogram PT of 1 within 0 500
histogram PT of 1 within 0 500
```

# Overview over allowed cuts and Histogram syntax

| Code | Alternative code(s) | # Args | Description |
|---|---|---|---|
| - | | 0 — 2 | No cut |
| M | Q | 1 | (Signed) invariant mass $M = \text{sgn}(p^2)\sqrt{\|p^2\|}$ |
| LM | LQ | 1 | $\log_{10} \|M\|$ |
| MSQ | QSQ S T U | 1 | Squared invariant mass $M^2 = p^2$ |
| E | | 1 | Energy in the lab frame |
| LE | | 1 | $\log_{10} E$ |
| PT | | 1 | Transverse momentum $p_\perp$ |
| LPT | | 1 | $\log_{10} p_\perp$ |
| PL | | 1 | Longitudinal momentum $p_L$ |
| P | | 1 | Absolute value of momentum $\|\vec{p}\|$ |
| Y | RAP RAPIDITY | 1 | Rapidity $y$ |
| ETA | | 1 | Pseudorapidity $\eta$ |
| DETA | DELTA-ETA | 2 | Pseudorapidity distance $\Delta\eta$ |
| PH | PHI | 1 | Azimuthal angle $\phi$ (lab frame) in radians |
| PHD | PHID PHI(DEG) | 1 | Azimuthal angle $\phi$ (lab frame) in degrees |
| DPH | DPHI DELTA-PHI | 2 | Azimuthal distance $\Delta\phi$ (lab frame) in radians |
| DPHD | DPHID DELTA-PHI(DEG) | 2 | Azimuthal distance $\Delta\phi$ (lab frame) in degrees |
| AA | ANGLE-ABS TH-ABS THETA-ABS | 1 | Absolute polar angle $\theta_{\text{abs}}$ (lab frame) in radians. Reference axis is the $z$-axis. |
| AAD | ANGLE(DEG) | | |
| | TH-ABS(DEG) THETA-ABS(DEG) | 1 | Absolute polar angle $\theta_{\text{abs}}$ (lab frame) in degrees |
| CTA | COS(TH-ABS) COS(THETA-ABS) | 1 | $\cos\theta_{\text{abs}}$ |
| A | ANGLE TH THETA | 2 | Relative polar angle $\theta$ (lab frame) in radians |
| AD | ANGLE(DEG) | | |
| | TH(DEG) THETA(DEG) | 2 | Relative polar angle $\theta$ (lab frame) in degrees |
| CT | COS(TH) COS(THETA) | 2 | $\cos\theta$ |
| A* | ANGLE* TH* THETA* | 2 | Relative polar angle $\theta^*$ (rest frame of part.#2) in radians |
| AD* | ANGLE*(DEG) | | |
| | TH*(DEG) THETA*(DEG) | 2 | Relative polar angle $\theta^*$ (rest frame of part.#2) in degrees |
| CT* | COS(TH*) COS(THETA*) | 2 | $\cos\theta^*$ |
| DR | DELTA-R CONE | 2 | Distance in $\eta$-$\phi$ space, i.e. $\sqrt{\Delta\eta^2 + \Delta\phi^2}$ |
| LDR | LOG-DELTA-R LOG-CONE | 2 | $\log_{10}\sqrt{\Delta\eta^2 + \Delta\phi^2}$ |

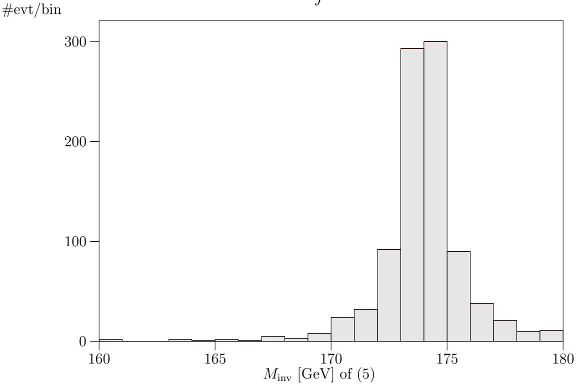
# WHiZard histograms

## New completely general cut syntax in W 2.0/O 1.0

`WHIZARD` **data analysis** March 16, 2007

**Process:** `qqttdec` ($u\bar{u} \to b\bar{b}W^+W^-$)

$\sqrt{s} = 500.0$ GeV $\int \mathcal{L} = 0.2754 \times 10^{-01}$ fb$^{-1}$




$\sigma_{\rm tot} = 36305. \pm 310.$ fb [$\pm 0.85$ %] $n_{\rm evt,\ tot} = 1000$

$\sigma_{\rm cut} = 36305. \pm 0.115 \times 10^{+04}$ fb [$\pm 3.16$ %] $n_{\rm evt,\ cut} = 1000$ [100.00 %]

# Examples for structured beams

```
&beam_input
 particle_name = "p"
 PDF_on = T
 PDF_ngroup = 4
 PDF_nset = 46
 PDF_scale = 1000
! PDF_running_scale = T
 /

 &beam_input
 particle_name = "p"
 PDF_on = T
 PDF_ngroup = 4
 PDF_nset = 46
 PDF_scale = 1000
! PDF_running_scale = T
```

```
 &beam_input
 particle_name = "A"
 CIRCE2_on = T
 CIRCE2_file = "teslagg_500_rr.circe"
 CIRCE2_polarized = T
 /

 &beam_input
 particle_name = "A"
 CIRCE2_on = T
 CIRCE2_file = "teslagg_500_rr.circe"
 CIRCE2_polarized = T
/
```

```
 &beam_input
 USER_strfun_on = T /
 &beam_input
 USER_strfun_on = T /
```

```
&beam_input
 particle_name = "e-"
 polarization = 0.80 0
 CIRCE_on = T
 CIRCE_acc = 2
 ISR_on = T
 ISR_alpha = 0.0072993
 ISR_m_in = 0.000511
 /

 &beam_input
 particle_name = "e+"
 polarization = 0 0.40
 CIRCE_on = T
 CIRCE_acc = 2
 ISR_on = T
 ISR_alpha = 0.0072993
 ISR_m_in = 0.000511
```

```
! WHIZARD 1.51 (Jun 15 2005)
! Reading process data from file whizard.in
! Reading process data from file sps1a.in
! Reading SUSY Les Houches Accord (SLHA) data
! SLHA: Spectrum calculator name:    SOFTSUSY
! SLHA: Spectrum calculator version: 1.9
! Wrote whizard.out
! Reading phase space configurations from file whizard.phx
!
! Process qqttdec:
!    u a-u ->   b a-b  W+ a-W+
!   32  16 ->   1   2   4    8
! Process energy set to     14000.     GeV

 *****  CERN Computer Program Library     -     Reference:  W5051  *****
 *****  PDFLIB Version:  8.04  Released on  2000-04-17  at  12.24  *****
 PDFLIB  : TMAS value
 Warning : NON standard settings, TMAS value =   174.  set by user !!
```

# The Phantom Menace – the MSSM

- 5318 couplings (with Goldstone/4-point)
- negative neutralino matrices: explicit factor of $i$
- Fully implemented, fully tested and fully functional
- Model `MSSM`
- Recommended usage: SUSY Les Houches Accord (SLHA)

```
&process_input
 process_id ``your_susy_proc''
 .....................
 input_file = "sps1a"
 input_slha_format = T
```

**What about tests? Have we checked?**


- Unitarity Checks $2 \to 2, 2 \to 3$
- Ward-/Slavnov-Taylor identities for gauge symmetries and SUSY

# Comparison of Automated Tools for Perturbative Interactions in SuperSymmetry

cf. e.g. `http://www.physics.carleton.ca/~reuter/susy_comparison.html`

$\tau^+\tau^- \to X$

| Process | status | Madgraph/Helas 0.5 TeV | Madgraph/Helas 2 TeV | Whizard/O'Mega 0.5 TeV | Whizard/O'Mega 2 TeV | Sherpa/A'Megic 0.5 TeV | Sherpa/A'Megic 2 TeV |
|---|---|---|---|---|---|---|---|
| $\tilde\tau_1\tilde\tau_1^*$ | ● | 257.57(7) | 79.63(4) | 257.32(1) | 79.636(4) | 257.30(1) | 79.638(4) |
| $\tilde\tau_2\tilde\tau_2^*$ | ● | 46.55(1) | 66.86(2) | 46.368(2) | 66.862(3) | 46.372(2) | 66.862(3) |
| $\tilde\tau_1\tilde\tau_2^*$ | ● | 95.50(3) | 19.00(1) | 94.637(3) | 19.0015(8) | 94.645(5) | 19.000(1) |
| $\tilde\nu_\tau\tilde\nu_\tau^*$ | ● | 502.26(7) | 272.01(8) | 502.27(2) | 272.01(1) | 502.30(3) | 272.01(1) |
| $\tilde\chi_1^0\tilde\chi_1^0$ | ● | 249.94(2) | 26.431(1) | 249.954(9) | 26.431(1) | 249.96(1) | 26.431(1) |
| $\tilde\chi_1^0\tilde\chi_2^0$ | ● | 69.967(3) | 9.8940(3) | 69.969(2) | 9.8940(4) | 69.968(3) | 9.8937(5) |
| $\tilde\chi_1^0\tilde\chi_3^0$ | ● | 17.0387(3) | 0.7913(1) | 17.0394(1) | 0.79136(2) | 17.040(1) | 0.79137(5) |
| $\tilde\chi_1^0\tilde\chi_4^0$ | ● | 7.01378(4) | 1.50743(3) | 7.01414(6) | 1.5075(5) | 7.0141(4) | 1.50740(8) |
| $\tilde\chi_2^0\tilde\chi_2^0$ | ● | 82.351(7) | 18.887(1) | 82.353(3) | 18.8879(9) | 82.357(4) | 18.8896(1) |
| $\tilde\chi_2^0\tilde\chi_3^0$ | ● | — | 1.7588(1) | — | 1.75884(5) | — | 1.7588(1) |
| $\tilde\chi_2^0\tilde\chi_4^0$ | ● | — | 2.96384(7) | — | 2.9640(1) | — | 2.9639(1) |
| $\tilde\chi_3^0\tilde\chi_3^0$ | ● | — | 0.046995(4) | — | 0.0469966(9) | — | 0.046999(2) |
| $\tilde\chi_3^0\tilde\chi_4^0$ | ● | — | 8.5852(4) | — | 8.55857(3) | — | 8.5856(4) |
| $\tilde\chi_4^0\tilde\chi_4^0$ | ● | — | 0.26438(2) | — | 0.264389(5) | — | 0.26437(1) |
| $\tilde\chi_1^+\tilde\chi_1^-$ | ● | 185.09(3) | 45.15(1) | 185.093(6) | 45.147(2) | 185.10(1) | 45.151(2) |
| $\tilde\chi_2^+\tilde\chi_2^-$ | ● | — | 26.515(1) | — | 26.5162(6) | — | 26.515(1) |
| $\tilde\chi_1^+\tilde\chi_2^-$ | ● | — | 4.2127(4) | — | 4.21267(9) | — | 4.2125(2) |
| $h^0h^0$ | ● | 0.3533827(3) | 0.0001242(2) | 0.35339(2) | 0.00012422(3) | 0.35340(2) | 0.000124218(6) |
| $h^0H^0$ | ● | — | 0.005167(4) | — | 0.0051669(3) | — | 0.0051671(3) |
| $H^0H^0$ | ● | — | 0.07931(3) | — | 0.079301(6) | — | 0.079311(4) |
| $A^0A^0$ | ● | — | 0.07975(3) | — | 0.079758(6) | — | 0.079744(4) |
| $Zh^0$ | ● | 59.591(3) | 3.1803(8) | 59.589(3) | 3.1802(1) | 59.602(3) | 3.1829(2) |
| $ZH^0$ | ● | 2.8316(3) | 4.671(5) | 2.83169(9) | 4.6706(3) | 2.8318(1) | 4.6706(2) |
| $ZA^0$ | ● | 2.9915(4) | 4.682(5) | 2.99162(9) | 4.6821(3) | 2.9917(2) | 4.6817(2) |
| $A^0h^0$ | ● | — | 0.005143(4) | — | 0.0051434(3) | — | 0.0051440(3) |
| $A^0H^0$ | ● | — | 1.4880(2) | — | 1.48793(9) | — | 1.48802(8) |
| $H^+H^-$ | ● | — | 5.2344(6) | — | 5.2344(2) | — | 5.2345(3) |

# How to add a new model?

Easiest and safest: invite the authors for a (couple of) beer

Version WHiZard 2.0/O'Mega 1.0 brings major improvement:
CompHep-style model declarations possible within O'Mega (this sacrifices fully functionality and object-orientedness of O'Caml)

Until then (if you dare to go the high road):
Works only with the version on my homepage

- Add your new particles and couplings in `omega/src/models3.ml` in the `template[_Col]`
- Declare the couplings in `omega/src/omega_parameters_template.nw`
- Add the new particles and (trilinear) vertices to `whizard/conf/models/Template.mdl`
- Add the couplings constants (with their names declared in `omega/src/omega_parameters_template.nw`) to `whizard/conf/models/parameters.Template.omega.f90`
- Do the debugging by yourself, no responsibility from the authors!

# Final Remarks

**Left out:**

- Details about phase space generation and integration

  We don't use black boxes, we write them!
- Fragmentation, hadronization, showers, underlying event (stay tuned!)
- No Advanced WHiZard spells: grid adaptation, dirty tricks, failures
- Implementation of NLO calculations
  (come to my talk at LoopFest/Fermilab)

**Thanks to all contributors** (list is not exhaustive!)

T. Barklow, M. Beyer, R. Chierici, K. Desch, M. Mertens, N. Meyer, K. Mönig, M. Moretti, H. Reuter, T. Robens, S. Rosati, A. Rosca, J. Schumacher, M. Schumacher, C. Schwinn

# Upgrade '07

# Upgrade '07