

# WHIZARD -Tutorial



Beijing, China

August 2014

## 1 How to use this tutorial

WHIZARD is a program system designed for the efficient calculation of multi-particle scattering cross sections and simulated event samples. The project web page can be reached via the URL

<http://whizard.hepforge.org/>

This tutorial will introduce you to the handling of NLO (next-to-leading order) QCD corrections in WHIZARD. Please note that this is a new capability of the program that is not officially supported yet, and has been successfully tested only for simple applications. The tutorial will provide a preview of this feature, but the specific commands that control it are still subject to change and will probably look differently in the final program..

Whenever you encounter a line starting with a dollar sign \$, the remainder is a command which you should run in the shell. Boxed frames filled with code are SINDARIN scripts (WHIZARDs flavor of input files) which you should copy into your favorite editor and run through WHIZARD. Typewrite-style paragraphs are sample WHIZARD output.

Considering standard usage of WHIZARD, please consider the manual that is available from the WHIZARD homepage, or within the distribution. The manual does not yet cover NLO calculations

## 2 Installation Procedure

For this specific tutorial, you will have to install both the GoSam package and a current development version of WHIZARD. We assume that you are running a computer with a recent Linux distribution.

1. Please check whether a suitable Fortran compiler (`gfortran 4.8` or newer) is installed as the default Fortran compiler on your machine.
2. You also need the `OCaml` program language package. This should be available from the software repository of your Linux system, if it is not installed already.
3. You should now download and install `GoSam`. The website is

`http://gosam.hepforge.org/`

Please use the installation script as described on that page. The installation path that you choose here will also be required for installing `WHIZARD`.

4. (If not already done:) Choose an empty directory and unpack the development version of `WHIZARD` that you got as a tar file

```
$ tar xzf whizard-2.2.3.tar.gz
```

This creates a subdirectory `whizard-2.2.3`, which you should leave untouched.

5. For convenience, create a separate build directory:

```
$ mkdir build-nlo
$ cd build-nlo
```

6. Configure `WHIZARD` in the current directory with the following flags

```
$ ../whizard-2.2.3/configure --prefix=$PWD/../install-nlo --disable-static
--enable-gosam --with-gosam=/your/gosam/installation/path
```

(The flags should be all on the same line with the command.)

You will have to do this even if you installed `WHIZARD` previously, because you probably didn't enable `GoSam` then.

7. Check if this was successful. For instance, within the build directory:

```
$ grep -i gosam.py config.log
```

The check was actually displayed on screen, but likely too fast to read.

8. You are now ready to compile and install the private version. Since you have used the `prefix` option, the installation will reside in the `install-nlo` subdirectory of the initial directory.

```
$ make install
```

### 3 NLO Example

For the project, you may start with an empty work directory,

```
$ mkdir work-nlo
$ cd work-nlo
```

It is probably a good idea to add the current NLO WHIZARD installation to your `$PATH`. You achieve this by

```
$ source ../install-nlo/bin/whizard-setup.sh
```

where you may replace the installation path accordingly, if the work and install directory are not on the same directory level. Check your setup with

```
$ whizard --version
```

You are now ready to run the example. Open your favorite editor and create a script named `nlo.sin` with the contents

```
mW = 80.376
mZ = 91.1876
GF = 1.16637E-005
wZ = 2.4952
wW = 2.124
```

```
process eeuu_nlo = E1, e1 => u, U {nlo}
sqrts = 250 GeV
```

```
integrate (eeuu_nlo) {iterations=5:50000:"gw"}
```

We use default values for all parameters. The process is

$$e^+e^- \rightarrow u\bar{u}$$

at a collider energy of 250 GeV.

Run WHIZARD by the command

```
$ whizard nlo.sin
```

After `GoSam` has constructed the virtual matrix element code, you will get a result that consists of three components. The first component is the Born cross section, the second component is the (subtracted) real emission, and the third component is the (also subtracted) virtual correction. The latter should be negative.

Add the the results of the real and virtual corrections and divide by the Born value. The result should be  $\alpha_s/\pi$ , that is for the given parameters: 0.0375.