ILC XSEC Data Base

C. Calancha (KEK)
calancha@post.kek.jp

June 21, 2014

Motivation

Importance of databases

- Knowing where is stored the information is just half of the story.
- Fast and efficient access to such information is also very important.
- Make info available to a broader audience (i.e. not ILC community members).

ILC XSEC DB

- It is extremelly useful tool (and necessary i would say).
- Before ilc-xsec-db, every member in our physics group collecting all xsec.
  - Unefficient, painful,duplicate of efforts, dangerous (easy to introduce typos)...
  - Clearly this task should be centralized.
  - People doing analysis should focus on analysis strategy.
- Having a tool telling you how much H bosons you produce at given L & beam pol. is nice.
  - Useful for people deciding ILC stagging scenarios?
- Or giving you the weighted cross section at your beam pol. is also nice.
  - Please, do not multiplicate by head on XXI century.

- Original design as simple ASCII tables.
  - Totally overcome.
  - Not: flexible enough, powerful enough, safe enough, fun enough.
  - In summary, not enough.

General Design targets

- It should be fast and scalable.
- Easy to use.
- Extensible when new production samples comming.
- It should report clear error messages about wrong input.
- Nice an elegant (makes maintenance more fun).

Command Line Tool: *The girl of my eyes (*)*

- Normal user should be available to run it.
  - Not enough privileges to run sql queries on kekcc.
- Thought as a tool for interactive session
  - No intention to implement it within Marlin framework.
  - Not strictly necessary in order to record xsec in your final ntuple (see my last example).
- Portability: target is to run on kekcc cluster, portability is not a goal.
  - Anyway it uses standard tools often installed by default on `Unix` enviroments.
  - Running under `Windows` using `Cygwin`? (it should, if satisfied all dependences).

(*) Sorry $\alpha, \beta, \gamma, \ldots, \epsilon, \ldots$ (this is a joke, just ignore it).

Web Interfaz.

Our memory is limited: Do not wasted it!

- This is the tool for coffee break discussions.
- You are on workshop, and one LHCb buddy asks you:
- how many H do you expect on the channel X with L=Y at beam pol Z?
- You dont remember and you feel a little shame...

ilc-xsec-db is here to rescue you! (*)

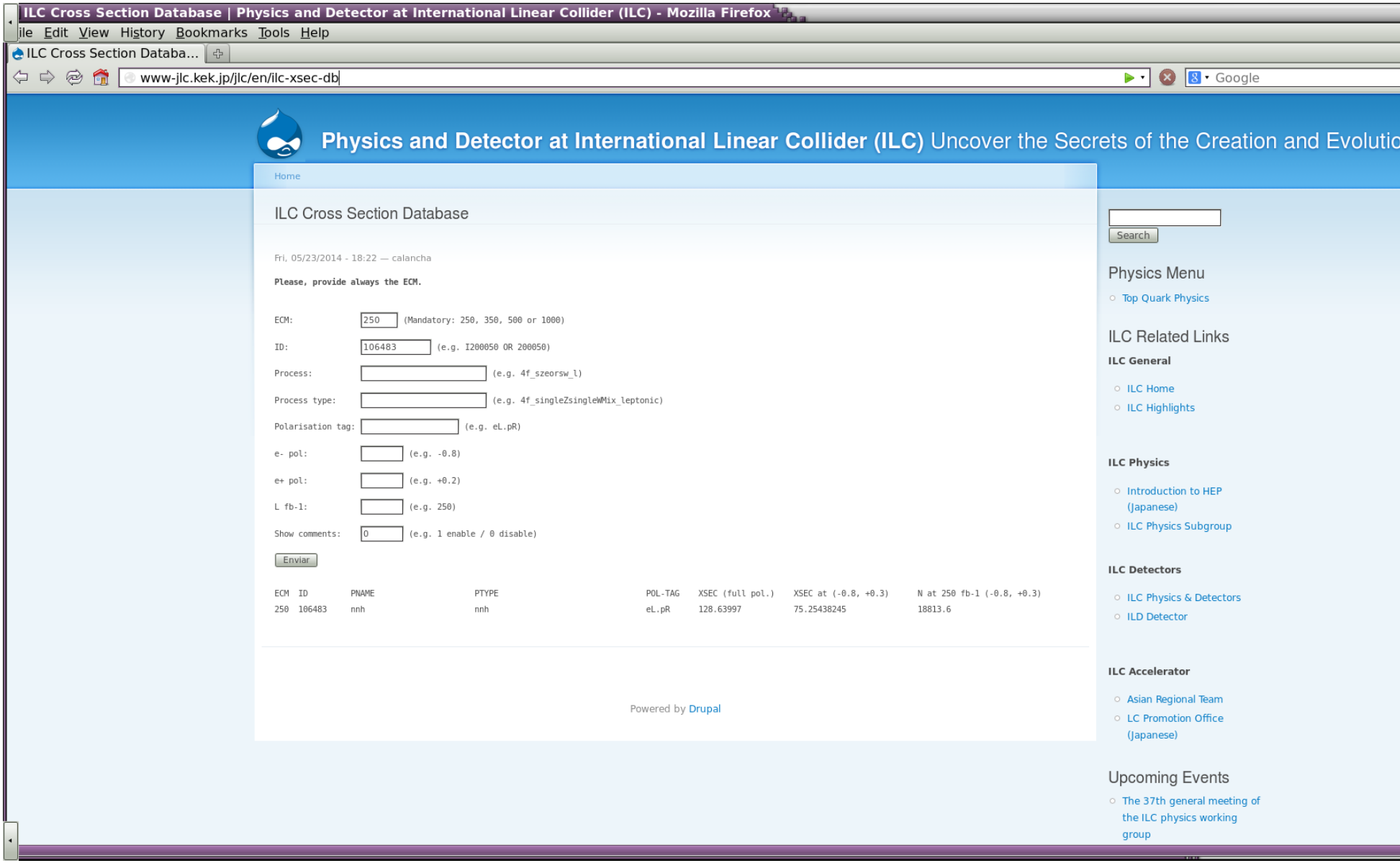
- i dont have such problems:
  - Still single.
  - i use the ilc-xsec-db.

http://www-jlc.kek.jp/jlc/en/ilc-xsec-db

- Tested on `Mozilla Firefox` v24 (working well).
- ♠ Seems not working on `Opera` v12.16 and `Internet Explorer`.
- ♠ Not working for text-based browsers (anyone here use them?).
- ? Sorry i dont use neither `Chrome` nor `Mac` products.
- ? Any cell phone user can test it in such devices? (Sorry i dont use them!)

(*) Sorry, ilc-xsec-db only contains ILC physics records.

Record this link in your web browser favourites



Command Line Tool

Latest version of command line here: http://www-jlc.kek.jp/jlc/en/subg/physics

Anyway, preferred method is to call this tool from kekcc (kekcc db always updated).

How to use

ilc-xsec-db

- Include in your PATH the directory were ilc-xsec-db is located:
  - export PATH=/group/ilc/soft/samples/ilc-xsec-db/bin:$PATH
- Interfaz accept several options to set parameters and perform finer query.
  - Both `UNIX` and `GNU` option styles are supported.
- Tested on kekcc cluster and it works.
- If no options provided (or called with '-h') documentation on the options is displayed.
  - Please, read this documentation before to use this tool.

How it performaces

- list all 'nnh' processes at ECM = 250 GeV:
  - Start: Fri Jun 13 13:24:57.431057000 JST 2014
  - End: Fri Jun 13 13:24:57.638698000 JST 2014
- Wow! that was really fast!
- List all samples contained in the db:
  - Start: Fri Jun 13 13:17:44.978739000 JST 2014
  - End: Fri Jun 13 13:17:51.055819000 JST 2014
- 6 seconds: ok for an interactive session.
- The user interface consume most of the time:
  - From the bash manual: 'BUGS: It's too big and too slow.'

Both, web and command line versions are scalable

- The web browser interfaz is sql based (scalability not a problem).
- The core implementation of the terminal tool is based on hash tables.
  - Average cost independent of the number of elements stored.
  - Including future `TDR` samples will not introduce any performance penalties.

Some Examples

- Current version only accept single-value arguments (–name=nnh)
- One natural extension would be accept something like: –name=(nnh,4f_sznu_l)
  - Considering to support this in the future (if i have time).
- With current version, you can 'simulate' such things calling program in a loop:

Example I

Example I (*)

```
echo ;(
  names="4f_sznu_l nnh"
  ecm="250 500"
  for n in $names
  do
    for e in $ecm
    do
      ilc-xsec-db --name=$n --ecm=$e
    done
  done
)
```

Easy copy and paste

- Calling inside loop your output dont show the terminal prompt.
- You can pass -v argument to get the name of the columns.
- All process name/type are from official metadata files.

(*) Just an example, the string 'names' can contain arbitrary number of process names.

Output Example I

xterm

```
File Edit Options Buffers Tools Help
START: Sat Jun 21 09:15:08.560869000 2014 JST
250  106589     4f_sznu_l              4f_singleZnunu_leptonic          eL.pR     192.75282          112.760400          28190.1
250  106590     4f_sznu_l              4f_singleZnunu_leptonic          eR.pL     39.31864           1.376152            344.0
500  250054     4f_sznu_l              4f_singleZnunu_leptonic          eL.pR     278.797            163.096245          81548.1
500  250056     4f_sznu_l              4f_singleZnunu_leptonic          eR.pL     14.8503            0.519760            259.9
250  106483     nnh                    nnh                              eL.pR     128.63997          75.254382           18813.6
250  106484     nnh                    nnh                              eR.pL     65.098189          2.278437            569.6
250  108029     nnh                    nnh_aa                           eL.pR     0.295872           0.173085            43.3
250  108030     nnh                    nnh_aa                           eR.pL     0.149726           0.005240            1.3
250  108061     nnh                    nnh_zz                           eL.pR     3.434687           2.009292            502.3
250  108062     nnh                    nnh_zz                           eR.pL     1.738122           0.060834            15.2
250  108045     nnh                    nnh_za                           eL.pR     0.199392           0.116644            29.2
250  108046     nnh                    nnh_za                           eR.pL     0.100902           0.003532            0.9
250  108077     nnh                    nnh_zz_4n                        eL.pR     0.137259           0.080297            20.1
250  108078     nnh                    nnh_zz_4n                        eR.pL     0.06946            0.002431            0.6
250  108013     nnh                    nnh_mumu                         eL.pR     0.028429           0.016631            4.2
250  108014     nnh                    nnh_mumu                         eR.pL     0.014387           0.000504            0.1
500  106523     nnh                    nnh                              eL.pR     289.08088          169.112315          84556.2
500  106524     nnh                    nnh                              eR.pL     21.559948          0.754598            377.3
500  108189     nnh                    nnh_aa                           eL.pR     0.664886           0.388958            194.5
500  108190     nnh                    nnh_aa                           eR.pL     0.049588           0.001736            0.9
500  108221     nnh                    nnh_zz                           eL.pR     7.718459           4.515299            2257.6
500  108222     nnh                    nnh_zz                           eR.pL     0.575651           0.020148            10.1
500  108205     nnh                    nnh_za                           eL.pR     0.448075           0.262124            131.1
500  108206     nnh                    nnh_za                           eR.pL     0.033418           0.001170            0.6
500  108237     nnh                    nnh_zz_4n                        eL.pR     0.308449           0.180443            90.2
500  108238     nnh                    nnh_zz_4n                        eR.pL     0.023004           0.000805            0.4
500  108173     nnh                    nnh_mumu                         eL.pR     0.063887           0.037374            18.7
500  108174     nnh                    nnh_mumu                         eR.pL     0.004765           0.000167            0.08339
END: Sat Jun 21 09:15:09.329887000 2014 JST

-UU-:----F1  ex1           All L30    [(Fundamental)] ----------
```

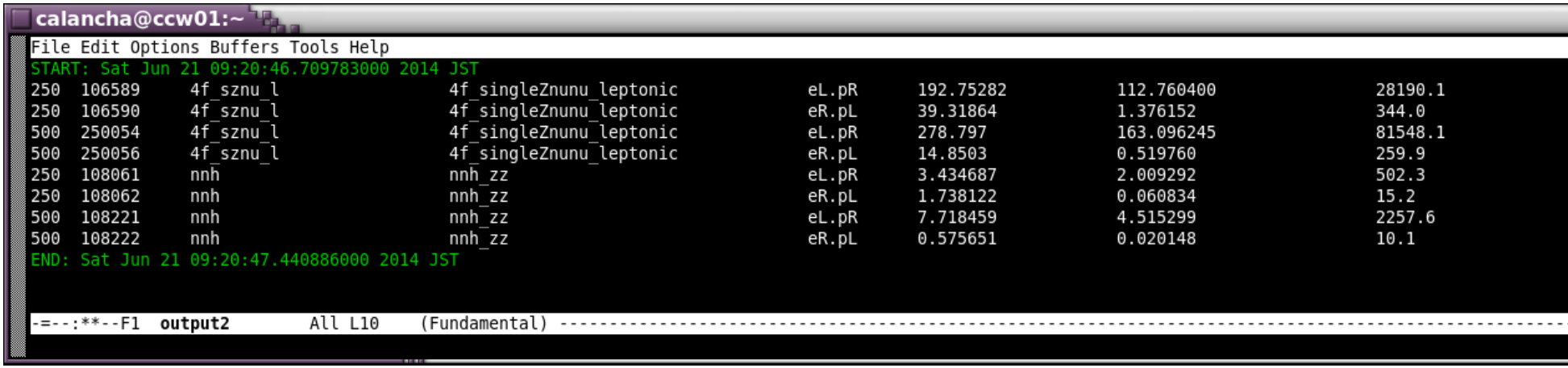
- Maybe user just interested in one specific Higgs mode in the nnh (see next example).

Example II: Restrict one of the processes

Example II

```
echo;(
  names="4f_sznu_l nnh"
  ecm="250 500"
  for n in $names
  do
    for e in $ecm
    do
      [[ "$n" == "nnh" ]] && ilc-xsec-db -n$n --ecm=$e --type=nnh_zz
      [[ "$n" == "nnh" ]] || ilc-xsec-db -n$n --ecm=$e
    done
  done
)
```

Output Example II



Unvaluable Tool when preparing submit scripts

- Its useful to store the xsec in your final ntuple.
- I do it passing xsec as a parameter to my jobs.
- ilc-xsec-db help me a lot of to do this fast and safe.

/group/ilc/soft/samples/mc-dbd/ild/dst-merged/500-TDR_ws/5f/ILD_o1_v05/v01-16-p05_500

- Example: prepare job submission of above kekcc dir.
- 5 fermions at 500 GeV (198 samples, that is, 198 DB queries to get the xsecs).

Output: Took less than 10 s !!! (*) (error free) *I'm lovin' It*

```
START: Fri Jun 20 20:40:18 JST 2014
xsec=0.001051
./sub $DIR rv01-16-p05_500.sv01-14-01-p00.mILD_o1_v05.E500-TDR_ws.I37270.Pae_evvvv.eB.pL- \
$xml 1 1 1 $xsec
xsec=0.017090
./sub $DIR rv01-16-p05_500.sv01-14-01-p00.mILD_o1_v05.E500-TDR_ws.I37282.Pae_evvxx.eB.pL- \
$xml 1 1 1 $xsec
.
.
.
xsec=0.266465
./sub $DIR rv01-16-p05_500.sv01-14-01-p00.mILD_o1_v05.E500-TDR_ws.I37379.Pae_ellyy.eW.pR- \
$xml 1 1 1 $xsec
xsec=0.062400
./sub $DIR rv01-16-p05_500.sv01-14-01-p00.mILD_o1_v05.E500-TDR_ws.I37383.Pae_eyyyy.eW.pR- \
$xml 1 1 1 $xsec
END: Fri Jun 20 20:40:26 JST 2014
```

(*)Here queries ran natively on `elips` (no external process calls). Running in standard way (`BASH` wrapper) would take much longer time. The safe way is ALWAYS running on the standard way.

Summary/Plan

Summary

- Developed ilc-xsec-db: a data base with cross sections of the official ILC MC samples.
- Web Interfaz to use with web browsers.
- Command line tool running at kekcc cluster.
  - Fast enough for interactive session.
  - Reliable, easy to use, easy to extend with new samples, scalable, elegant.

Plan

- Keep update the database (adding future official samples).
- Keep small maintanance for fix eventual bugs found by users.
- I could extend capabilities if i found the time
  - Already have ideas about possible extensions but my time is limited.

BACK UP