The FINUDA pre-analysis monitor

Diego Faso (faso@to.infn.it)

Last update: Tuesday, October 16, 2007

Main purposes

- **BEAM status:**
  - Luminosity (average/run and integrated/run)
  - Interaction point position and statistics
  - Center of mass energy

- **Event quality:**
  - Number of reconstructed K-/K+ / target and space
  - Pattern-Recognition-Error-Code evaluation

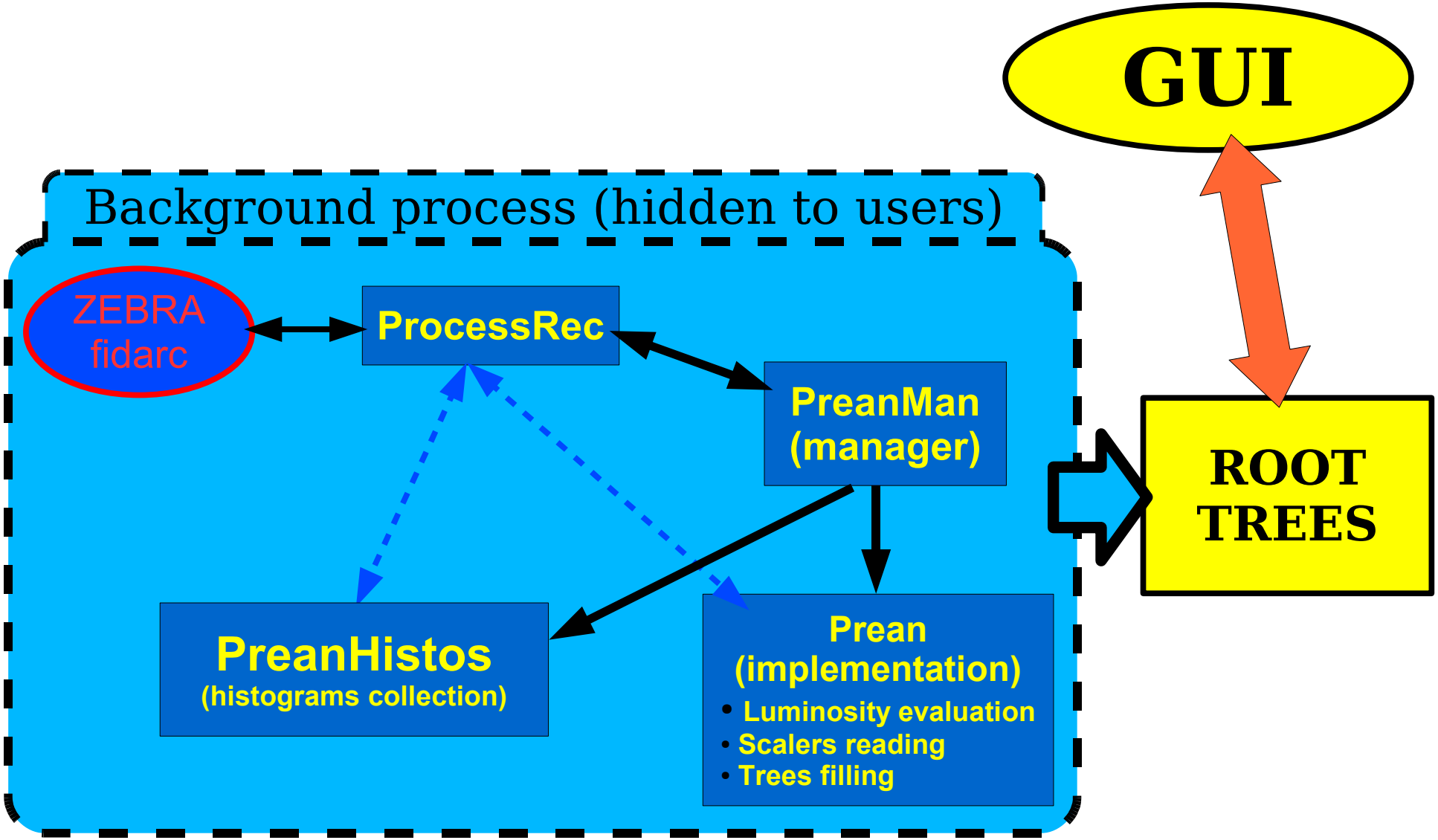
- **Reconstruction efficiency:**
  - Momentum resolution (μ+)
  - Momentum resolution (μ+) depending on the path
  - Support for short tracks

- **Stability:**
  - Trend of considered variables vs run.

Background process:

- **Start New Run (or new online-monitoring)**
  - Open raw file (offline mode)
  - Get raw event from UDP socket (online mode)
  - Get the run number from DAQ (online mode)
  - Reset counters
- **Event Loop**
  - Every event:
    - Perform pre-analysis step1 *[and stop here if needed]*
    - Fill histograms (if needed)
    - Call rdtupk (from fidarc)
    - Perform pre-analysis step2 *[and stop here if needed]*
    - Perform the complete fidarc reconstruction process
    - Perform pre-analysis step3 *[and stop here if needed]*
    - Fill histograms (if needed)
- **End Of Run**
  - Save Trees (used by the User interface for filling histograms).

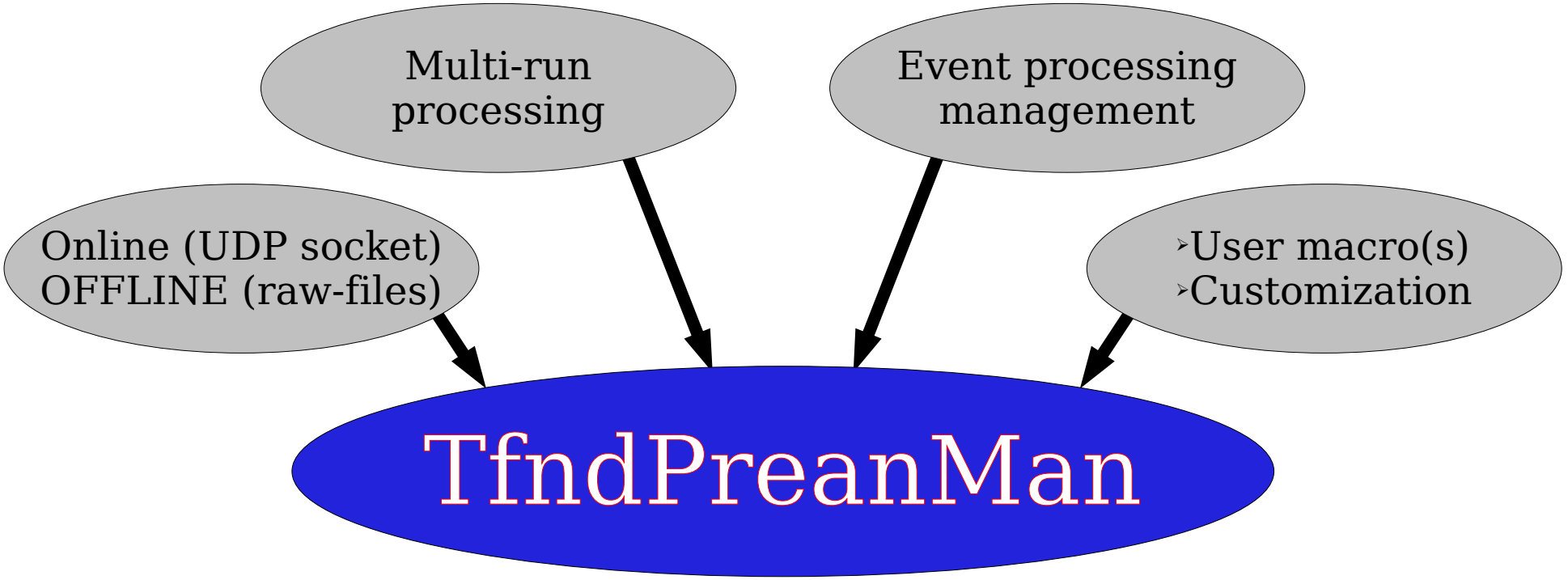
Structure (block diagram)



“Prean Manager”: features

The “TfndPreanMan” has been developed within the froot environment, thus exploiting all advantages provided by the steer class “TFndRun”.

The event processing can now be easily handled by writing a simple **(f)root** macro in which a pointer to TFndPreanMan is used.



Preanalysis steps

The **event-processing** has been **splitted** into several steps, in order to allow a complete **customization** for the running process(es)

1) **Get** the **event** (optionally fill froot HDT structure).
2) **Perform pre-analysis (step 1)**
3) **Perform** a **custom reconstruction** (optional)
4) **Build** standard **ZEBRA-structure** (read-lib is used)
5) **Perform pre-analysis (step 2)**
6) **Perform** a **custom reconstruction** starting from geometrical hits reconstructed by fidarc (optional)
7) **Perform** the FINUDA **standard reconstruction** (fidarc official FORTRAN code is used)
8) **Perform pre-analysis (step 3)**

The event-processing can be stopped at any step: the preanalysis can be customized according to:
cpu-performances, required statistics, DAQ rate, ...

Preanalysis steps

- The access to raw-event is handled by:
  - the read-lib (fin_open / fin_open03):
    the fidarc program is controlled via the *froot-fidarc* interface (TFndProcessRec class).
- User can decide to fill the froot HDT
  - A pointer to the raw-event is stored as data-mamber of TFndRun (base of TFndPreanMan)

1) Get the event
2) Pre-analysis (step 1)
3) Custom reconstruction
4) Build ZEBRA (FGES)
5) Pre-analysis (step 2)
6) Custom reconstruction
7) Standard reconstruction
8) Pre-analysis (step 3)

C++ steers the whole process

Preanalysis steps

Some important informations can be extracted directly from the pointer to the raw-event:

- Run completed

PreanMan

- Run number
- Event number
- Trigger status
- Scalers

Prean

Event reconsrtuction not involved yet

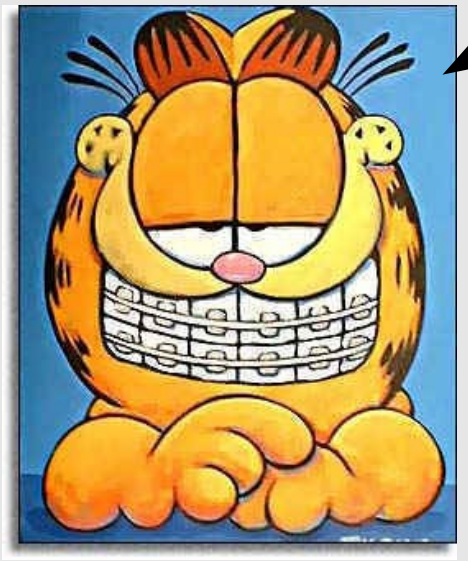
1) Get the event
2) Pre-analysis (step 1)
3) Custom reconstruction
4) Build ZEBRA (FGES)
5) Pre-analysis (step 2)
6) Custom reconstruction
7) Standard reconstruction
8) Pre-analysis (step 3)

Preanalysis steps

(This step is optional)

The raw-event can be directly used
for custom checks or analysis:

The pre-analysis manager can return the pointer to the raw-event.....
...this could make happy some users!



1) Get the event
2) Pre-analysis (step 1)
3) Custom reconstruction
4) Build ZEBRA (FGES)
5) Pre-analysis (step 2)
6) Custom reconstruction
7) Standard reconstruction
8) Pre-analysis (step 3)

The pre-analysis manager works as an interface for getting raw-events without the need to write new code!

Preanalysis steps

The read-lib is used to fill ZEBRA FGES structure:

- Geometrical hits are reconstructed starting from physics signals (rdtupk)
- ZEBRA-FGES structure is filled with:
- Geometrical reconstructed hits
- Detectors ADCs & TDCs

read-lib

C++ is steering the step, but no C++ code is required to implement these functions

1) Get the event
2) Pre-analysis (step 1)
3) Custom reconstruction
4) Build ZEBRA (FGES)
5) Pre-analysis (step 2)
6) Custom reconstruction
7) Standard reconstruction
8) Pre-analysis (step 3)

updates/changes in the fidarc code do not require any change in the C++ code!

Preanalysis steps

Some important informations are extracted directly from the content of the ZEBRA FGES structure

- Number of hits/layer (related to noise)
- Multiplicity
- Further ideas?

Prean

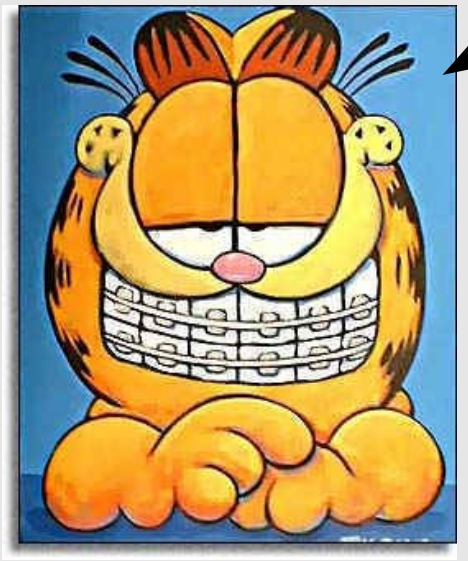
1) Get the event
2) Pre-analysis (step 1)
3) Custom reconstruction
4) Build ZEBRA (FGES)
5) Pre-analysis (step 2)
6) Custom reconstruction
7) Standard reconstruction
8) Pre-analysis (step 3)

Preanalysis steps

(This step is optional)

ZEBRA-FGES structure can be accessed for custom checks or analysis:

...this could make happy some users!



1) Get the event
2) Pre-analysis (step 1)
3) Custom reconstruction
4) Build ZEBRA (FGES)
5) Pre-analysis (step 2)
6) Custom reconstruction
7) Standard reconstruction
8) Pre-analysis (step 3)

Any C++/based reconstruction can be written inside a standard root macro without modifying the compiled code

User macros dependend on the zebra-FGES structure

Preanalysis steps

The fidarc (FORTRAN) complete event reconstruction is performed.

- The content of ZEBRA-FGES is used to reconstruct the event
- The ZEBRA-DST structure is filled

(all hidden into **one** user command)

fidarc

C++ is steering the step, but no C++ code is required to implement these functions

- NOTE:
  The content of ZEBRA-FGES can not be used anymore

1) Get the event
2) Pre-analysis (step 1)
3) Custom reconstruction
4) Build ZEBRA (FGES)
5) Pre-analysis (step 2)
6) Custom reconstruction
7) Standard reconstruction
8) Pre-analysis (step 3)

The C++ code steering this step is completely independent on any update/change in fidarc

Preanalysis steps

Some important informations are extracted directly from the content of the ZEBRA FDST structure

- Trigger flag: **Bhabha**
  - Momentum of reconstructed tracks
  - angle between reconstructed tracks
  - Position of the interaction point
  - Invariant mass
- Trigger flag: **HYPE**
  - (*) Number of reconstructed tracks / event
  - Number of stopped kaons/target
  - Momentum of μ^+
  - (*) Momentum of π^-

(*) in progress

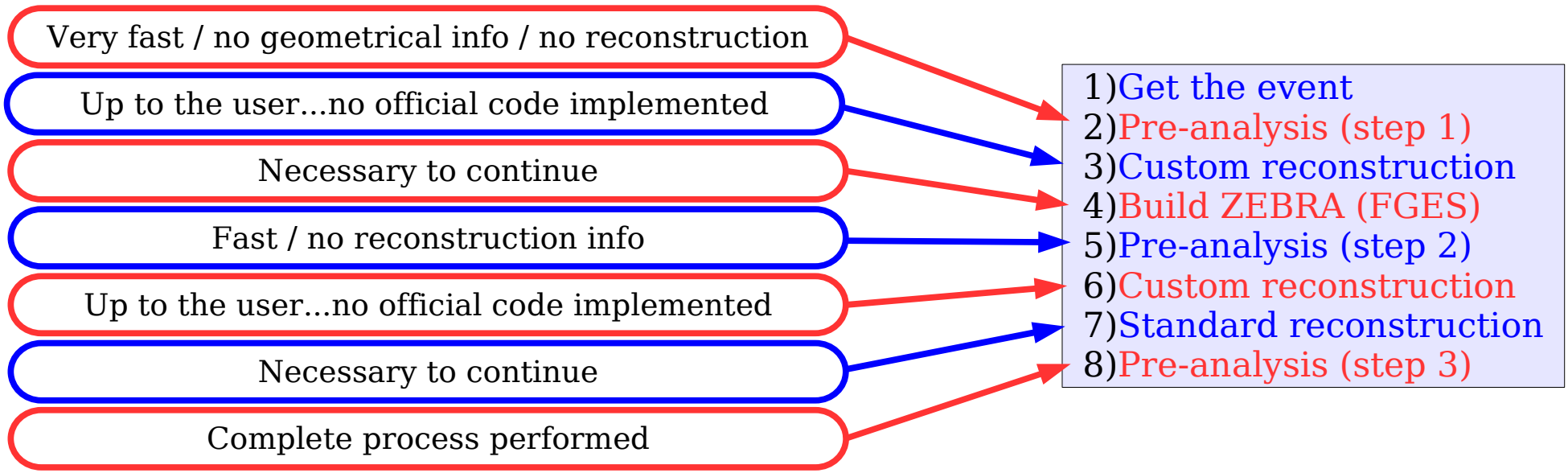
1) Get the event
2) Pre-analysis (step 1)
3) Custom reconstruction
4) Build ZEBRA (FGES)
5) Pre-analysis (step 2)
6) Custom reconstruction
7) Standard reconstruction
8) Pre-analysis (step 3)

- No need to store luminosity parameters into a huge database table
- Possibility of evaluating several parameters in a single step!

C++ code dependends on the zebra-FDST structure

Preanalysis customization

- The event-processing can be stopped at any step.
- The user can select the “stop-step” event by event.
- Users' compiled code can be used (ACLiC).
- Thinking about a configuration file (or better... a configuration class) for:
  - Default event processing customization
  - Possibility of applying cuts (chi-squared...)



Preanalysis customization

Event-processing example:
Low statistics on reconstructed informations / Fast scalers-reading / Fast processing

1) **Get the event**

2) **Pre-analysis (step 1)**

Every event

3) **Custom reconstruction**

4) **Build ZEBRA (FGES)**

5) **Pre-analysis (step 2)**

Every 3 event

6) **Custom reconstruction**

7) **Standard reconstruction**

8) **Pre-analysis (step 3)**

Every 9 event

Status of the art (Oct 16, 2007)

- **ProcessRec** (from the TProcess of Filippini/Panzarasa):
  - Completed and tested (multi-run processing available)

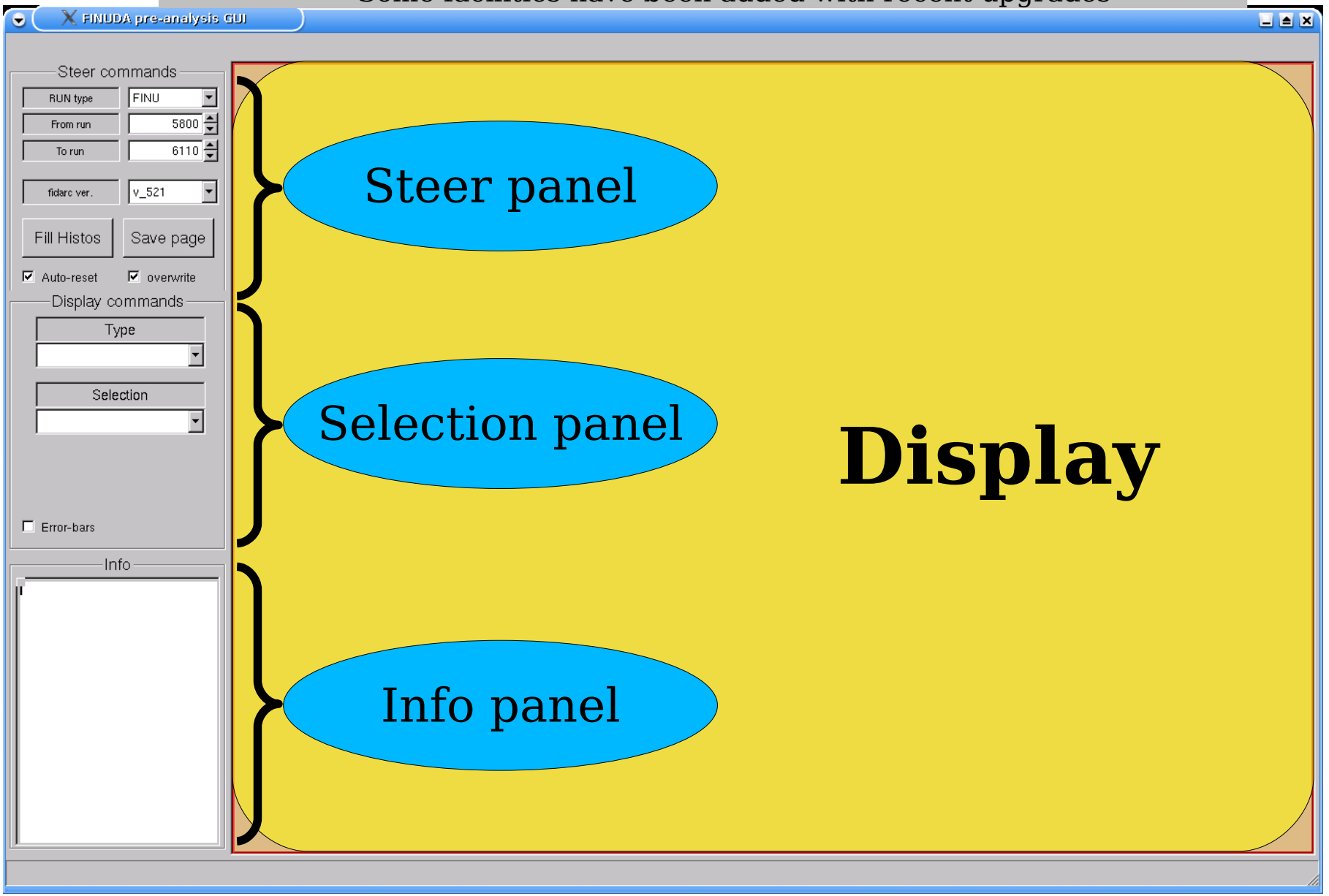
- **PreanMan:**
  - Implemented and tested (online-mode available)

- **PreanHistos:**
  - Completed

- **Prean:**
  - Completed
  - Implementation:
    - Read scalers: **OK (used for preanalysis purposes: not for DAΦNE)**
    - Luminosity evaluation (from Bhabha trig.): **OK**
    - Luminosity evaluation (from Hype trig.): **OK**

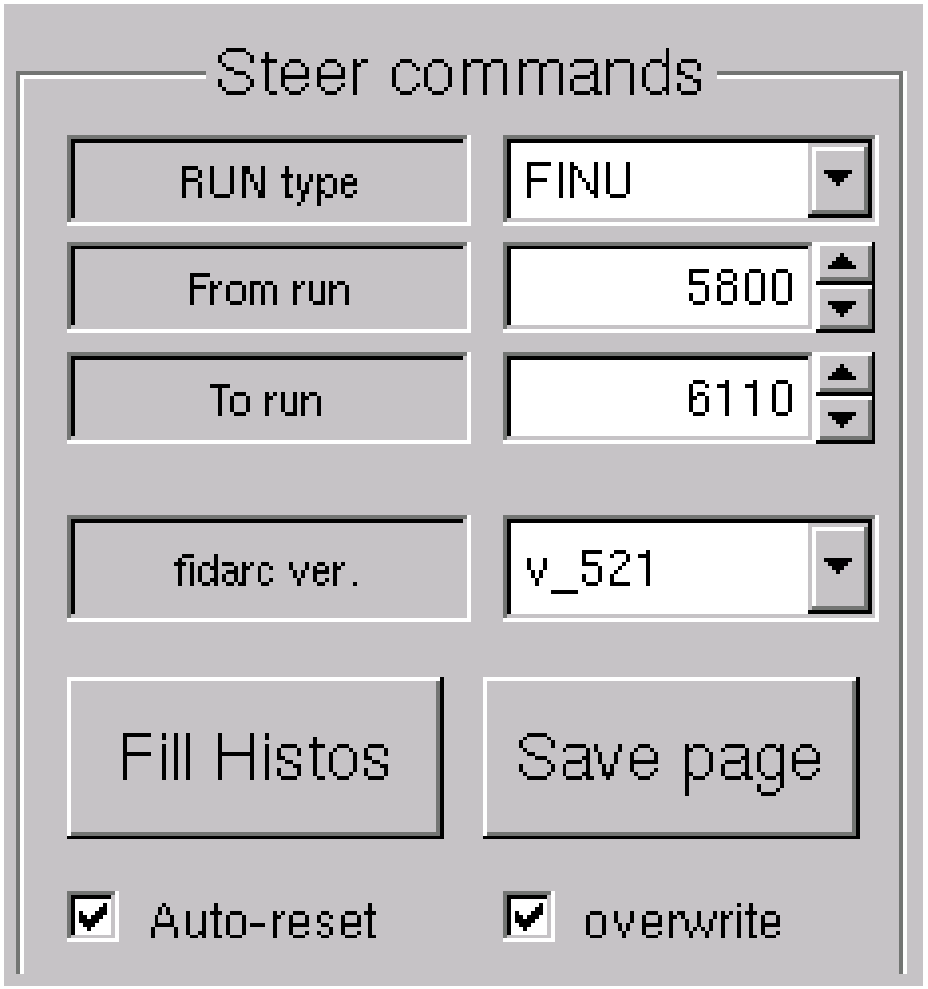
Pre-analysis GUI

Some facilities have been added with recent upgrades



Pre-analysis GUI (STEER)

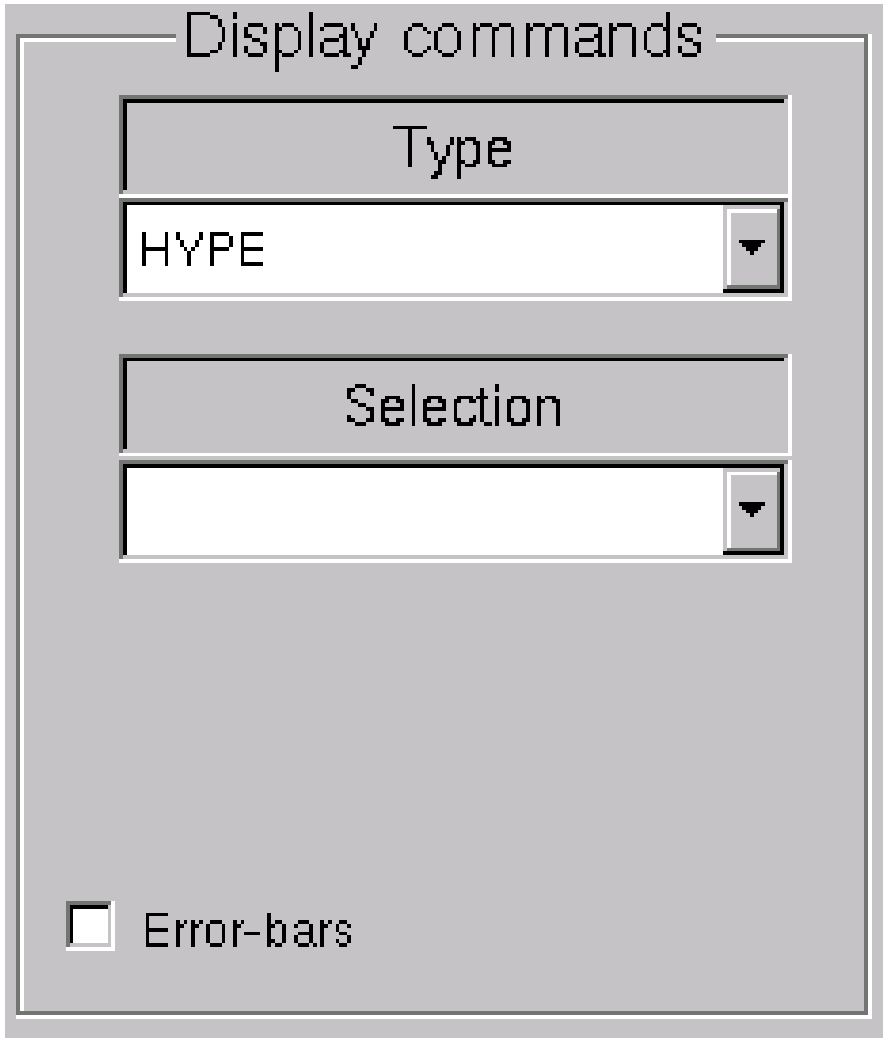
Some facilities have been added with recent upgrades



- Run type:
  "FINU" only is avalable
- From run / To run:
  Press "enter" after having set the required start/stop numbers
- Fidarc ver.:
  Is the version of fidarc to be used
- Fill Histos:
  By pressing this button all pre-analysis histograms are filled, according to the selected range of runs. (if the "Auto-reset" checkbox is disabled new histograms will be added to current ones)
- Save Page:
  A "pdf" file of current display is created (named according to the version-range setting)

Pre-analysis GUI (Selection)

Some facilities have been added with recent upgrades



... histograms browser ...

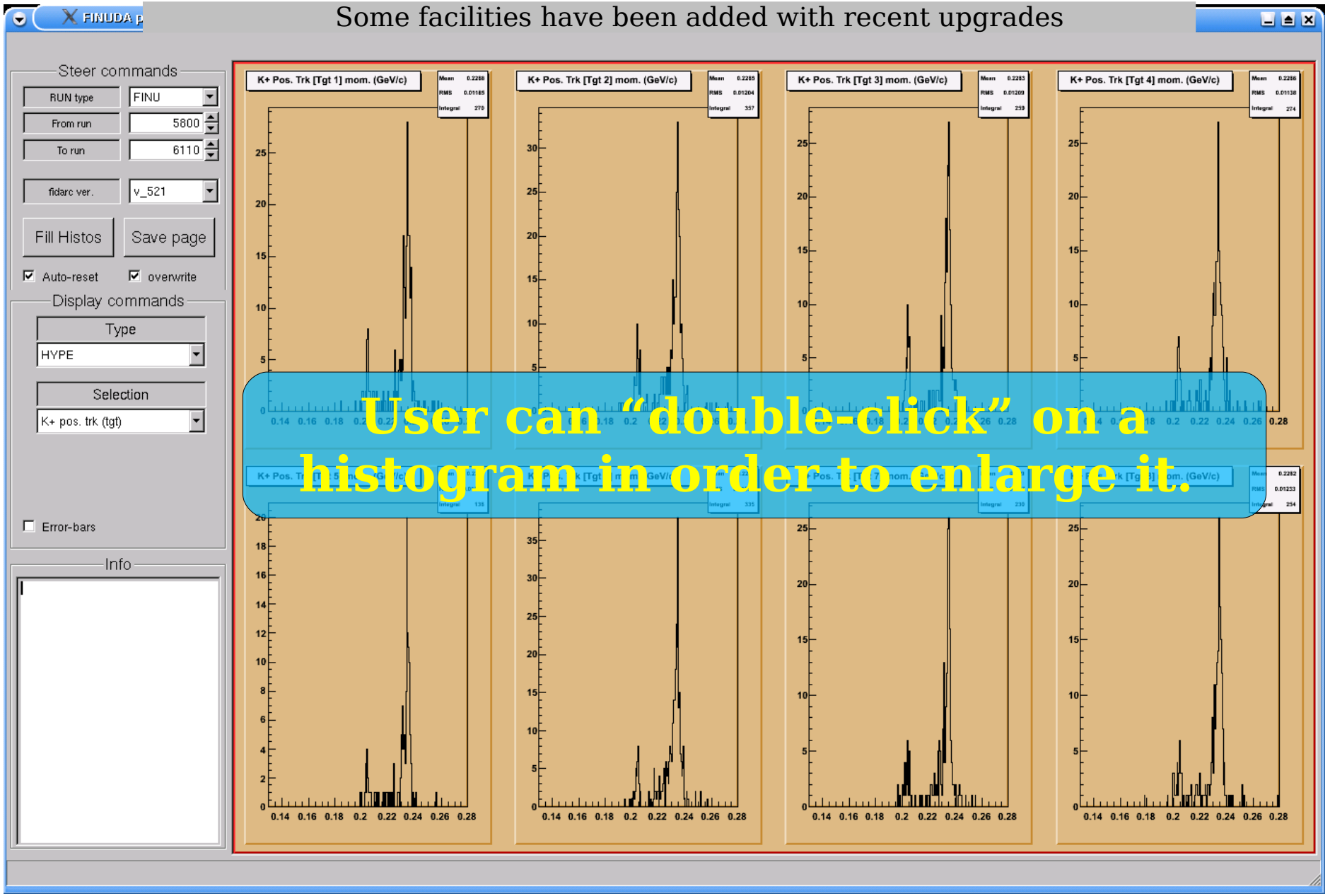
- Type:
  - GES (Global Event Structure)
  - BHABHA
  - HYPE
  - STATISTICS
- Selection:

  The content depends on the "Type" setting

  The "Error-bars" checkbox can be used in order to compute and draw errors.

Pre-analysis GUI (Display)

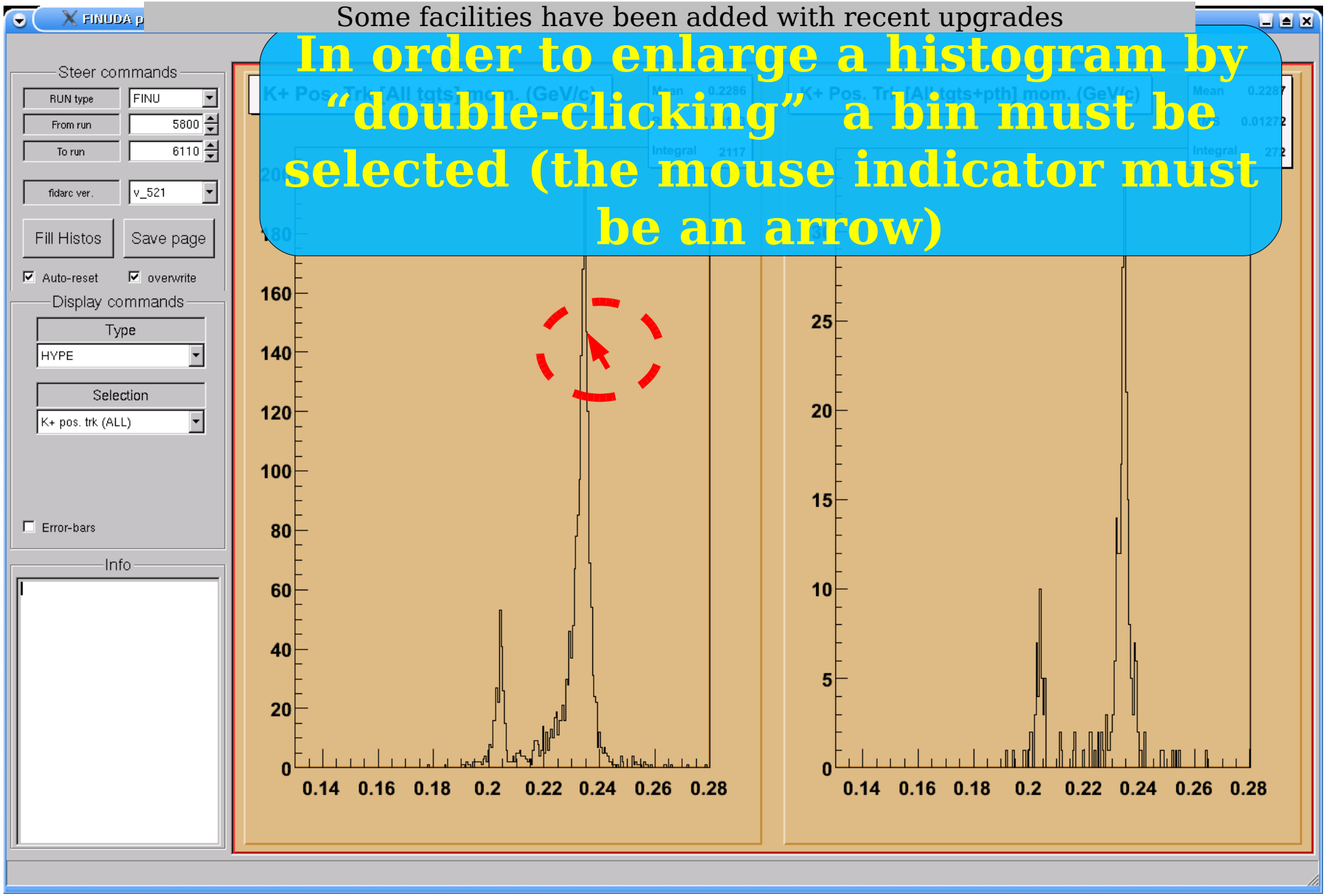
Some facilities have been added with recent upgrades



Pre-analysis GUI (Display)

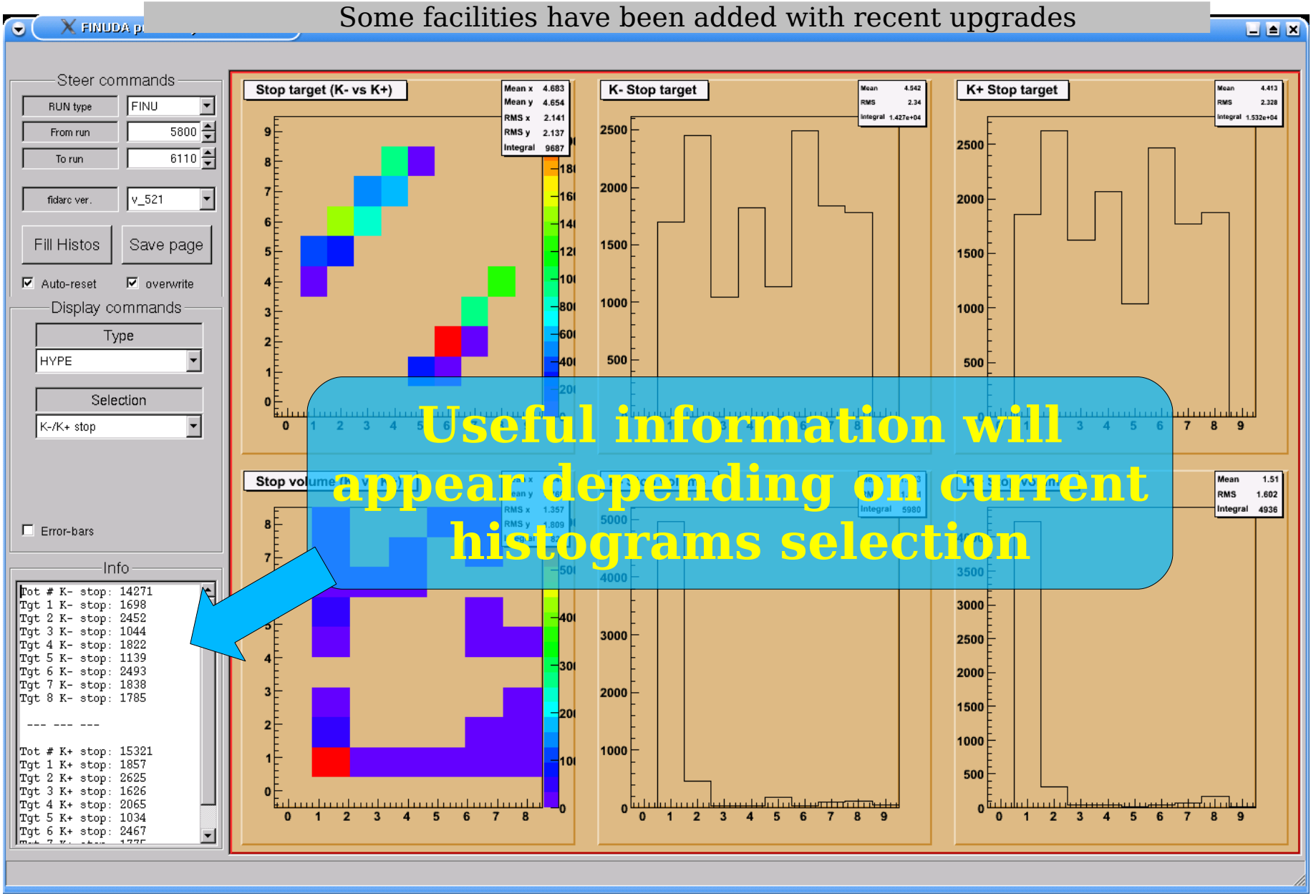
Some facilities have been added with recent upgrades

In order to enlarge a histogram by "double-clicking" a bin must be selected (the mouse indicator must be an arrow)



Pre-analysis GUI (Info)

Some facilities have been added with recent upgrades



Pre-analysis GUI (available information)

BHABHA

- Particles
  - Reconstructed momentum for e+ and e- (1D and 2D)
  - Invariant mass (different hypothesis)
  - Angle between reconstructed tracks
- Interaction point
  - Reconstructed position [x,y,z] (1D, 2D and 3D)
- TOF *(useful in order to check calibrations)*
  - reconstructed TOF for particles:
    - 2D: Bhabha TOF e^+ vs e^-
    - 1D: e^+ (e^-) TOF
  - Check for symultaneous events:
    - Time difference between slabs associated to e^+ and e^-

Pre-analysis GUI (available information)

HYPE

- Kaon stopping point (K^+ and K^-)
  - 2D stopping position (inside targets or other materials)
- Kaon recognition/reconstruction (“K^+/K^- stat”):
  - Kaon Pattern-Recognition Error-code (descr. for fida-ver > 521) :
    - see zebra blankdec
  - Kaon stopping code:
    - see zebra blankdec

Pre-analysis GUI (available information)

HYPE

- PHI decay point:
  - PHI decay reconstructed position (according to the K^+K^- recognition)

- *"K^+ pos trk (ALL)"*:
  momentum of positive tracks from K+ overall targets (with and without path selection [*])

- *"K^+ pos trk (tgt)"*:
  momentum of positive tracks from K+ target by target (considering both forward and backward tracks without applying qny quality cut)

- *"K^+ pos trk (pth)"*:
  momentum of positive tracks from K+ target by target (considering forward tracks only with path selection applied [*] ; no quality cuts)
- *"K^+ K^- TOF* :
  Useful for evaluating TOFINO performances on high-threshold.

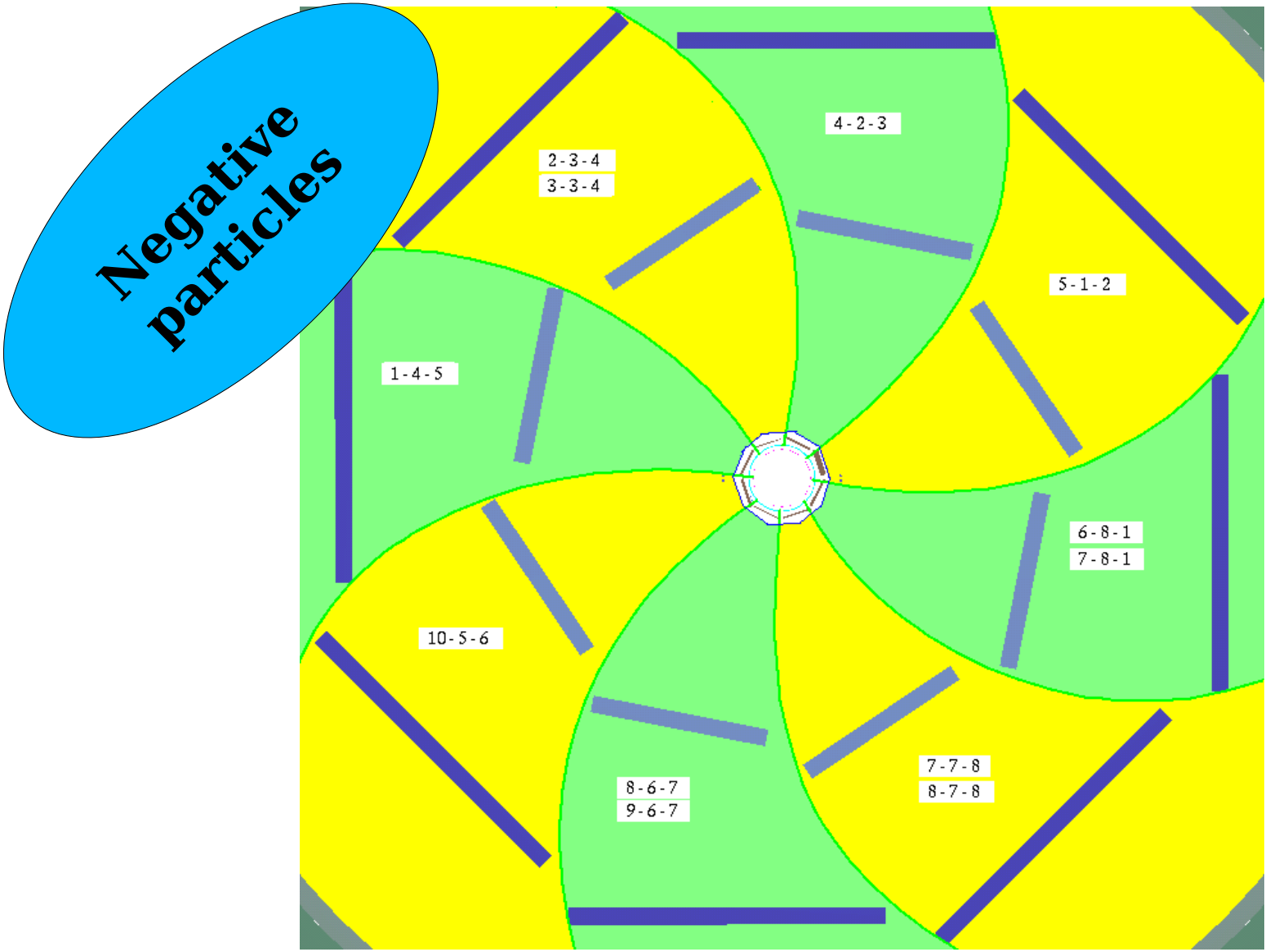
[*]: path selection described in following pages

Pre-analysis GUI (available information)

STATISTICS

- Lumin/run:
  - Luminosity (run average) vs run number
- Int. Lumin/run:
  - Integrated Luminosity vs run number
- Tracks momentum-integral/run:
  - Number of tracks and tracks momentum vs run number

Pre-analysis (available path-selection)



Pre-analysis (available path-selection)