Learning how to run FLUKA: Introduction to Flair

FLUKA Beginner's Course

Why is UI design important

- User Interfaces are what allows end users to interact with an application.
- A good UI will make an application intuitive and easy to use
- Excellent applications without good UI will be less popular than inferior ones with a good UI

What makes a good UI?

General:

- Simple
- Intuitive
- Respects the commonly accepted conventions
- Visually organized
- Native look
- Easily install and setup
- Extensible / Programmable

FLUKA:

- Do not hide the inner functionality
- Provide a platform for working/analyzing results

Language Choice

	Python	Java	Root/cint	C/C++
Distribution	Fedora: Pre-Installed M$ Win: installer, cygwin	Linux: package M$ Win: Installer, no-gygwin	Linux: package M$ Win: procedure no-cygwin	Linux: Pre-installed M$ Win: cygwin, djgpp
Flavors	Single	Several	Single	Many
Interpreted	√	√ VM	√	
Compiled		√ VM	√	√
Source Portability	√	√	√	
Binary Portability	√	√		
Interactive	√		√	

What is Python?

Python is a scripting language which is:

- interpreted
- interactive
- object-oriented
- like pseudo code
- dynamically typed
- available for many platforms
- extensible with C-API

Free from: http://www.python.org

GUI toolkits for Python

1st Choice

- Tkinter — default GUI toolkit for Python.
  Good for simple UIs.
  Portable, wrapper around tk/tcl
- wxPython — Most popular.
  Good for complex UIs.
  Wrapper on Win32, GTK
- JPython — Access to the Swing library
- PyGTK — Access to the well-known GTK toolkit
- PyQt — Access to the well-known Qt library
- win32all — Access to MFC from python (MS-Win only)
- WPY — MFC style, both also available for UNIX
- X11 — Limited to X Windows.

Plotting Engine

matplotlib — python 2D plotting library
http://matplotlib.sourceforge.net

gnuplot-py — Python interface to gnuplot
http://gnuplot-py.sourceforge.net

pyROOT — Python interface to ROOT

About

/fleə(r)/ n [U,C] natural or instinctive ability (to do something well, to select or recognize what is best, more useful, etc.

[*Oxford Advanced Dictionary of Current English*]

What is flair [1/2]

FLUKA Advanced Interface [http://www.fluka.org/flair]

- **All-in-one** User friendly graphical Interface;
- Minimum requirements on additional software;
- Working in an intermediate level
  Not hiding the inner functionality of FLUKA

Front-End interface:

- Fully featured **Input file Editor**
  - Mini-dialogs for each card, allows easy and almost error free editing
  - Uniform treatment of all FLUKA cards
  - Card grouping in categories and card filtering
  - Error checking and validation of the input file during editing
- **Geometry:** interactive visualization editing, transformation, optimizations and debugging (tomorrows talk);
- **Compilation** of the FLUKA Executable;
- **Running** and **monitoring** of the status of a/many run(s)

What is flair [2/2]

Back-End interface:

- Inspection of the output files (core dumps and directories)
- Output file(s) viewer dividing into sections
- Post processing (merging) the output data files
- Plot generation through an interface with gnuplot;

Other Goodies:

- Access to FLUKA manual as hyper text
- Checking for release updates of FLUKA and flair
- Nuclear wallet cards
- Library of materials
- Database of geometrical objects (Not yet completed)
- Programming python **API**
- Everything is accessible with keyboard shortcuts

Concepts: Flair Project

- Store in a single file all relevant information:
  - Project notes
  - Links to needed files: input file, source routines, output files …
  - Multiple runs from the same input file, as well running status
  - Procedures on how to run the code
  - Rules on how to perform data merging
  - Information on how to post process and create plots of the results
- You can consider Flair as an editor for the project files.
- Can handle any FLUKA input format (reading & writing), but internally it works using the names format for the input, free with names for the geometry (Recommended way of working)
- The format is plain ASCII file with extension: .flair

Note: If you want to copy a project you need to copy also all linked files especially the input and source routines!

Installation

- Flair web site to download code and documentation

  http://www.fluka.org/flair

- Installation procedures:
  - RPM/DEB method (Linux): strongly recommended! on systems that support the RPM/DEB. The package will create all file association, menu items and keep track of updates and files installed.
    The package will install the program to: /usr/local/flair
    and will create the following launcher programs:
    - /usr/local/bin/flair — flair program
    - /usr/local/bin/fm — FLUKA manual
    - /usr/local/bin/pt — Periodic Table
    - /usr/local/bin/fless — FLUKA output viewer

  - tar.gz method (MacOS, MS-Windows). Please follow the instructions on:
    http://www.fluka.org/flair/download.html
    and for special instruction on the FAQ:
    http://www.fluka.org/flair/faq.html

Starting flair

Programs Menu (Linux)

- Click the icon of Flair from the programs menu;
- Flair is registered under the Science/Physics category but depending your Linux distribution and window manager it might appear in different sub-menus (i.e. Applications, Education, Science or Others).

Window Manager (Linux, only via RPM or DEB installation)

- Flair makes an association of the following extensions:

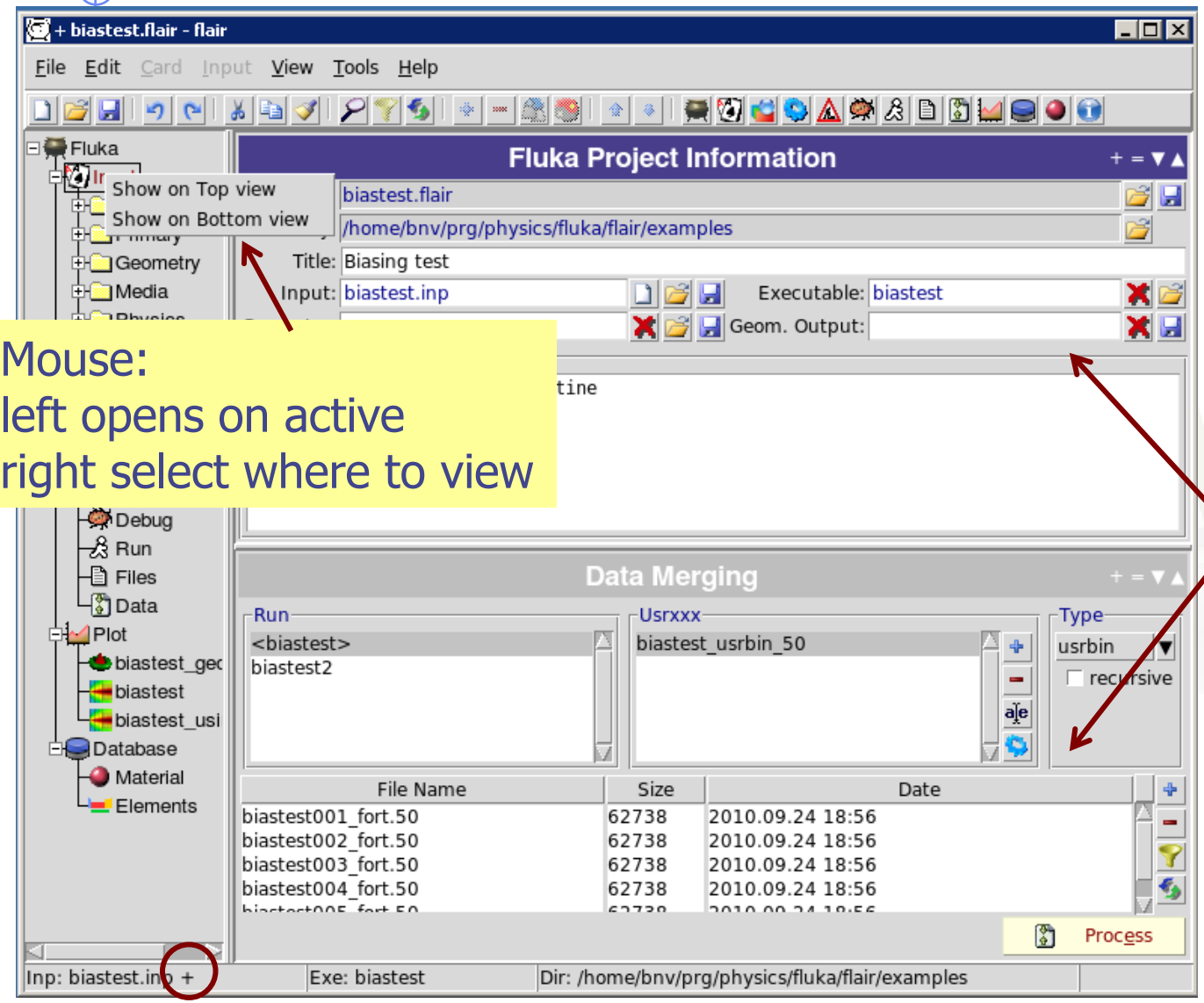
 ***.flair**

 ***.fluka *.inp**

Console

- Type the command flair. Remember to place in your **`$PATH`** the directory where flair is installed!

Interface



Mouse:
left opens on active
right select where to view

active

+ vertical/horizontal
= equalize
▼ minimize
▲ maximize

2 working frames

inactive
click to activate

input modified and not saved

Menus

File Edit Card Input View Tools Help

File	I/O, export to other formats, printing, recent projects
Edit	Common editing features: Cut, Paste, Add, Del, Clone, Filter
Card	Add or change cards in input; grouped by Categories
Input	Commands to manipulate input cards
View	Accessing various views of flair
Tools	General purpose commands: Terminal, Browser, Preferences
Help	Access to help, check for updates, web page, about dialog

Keyboard Short cuts (Linux/Windows): F10 or Alt+*F, E, C, I, V, T, H*

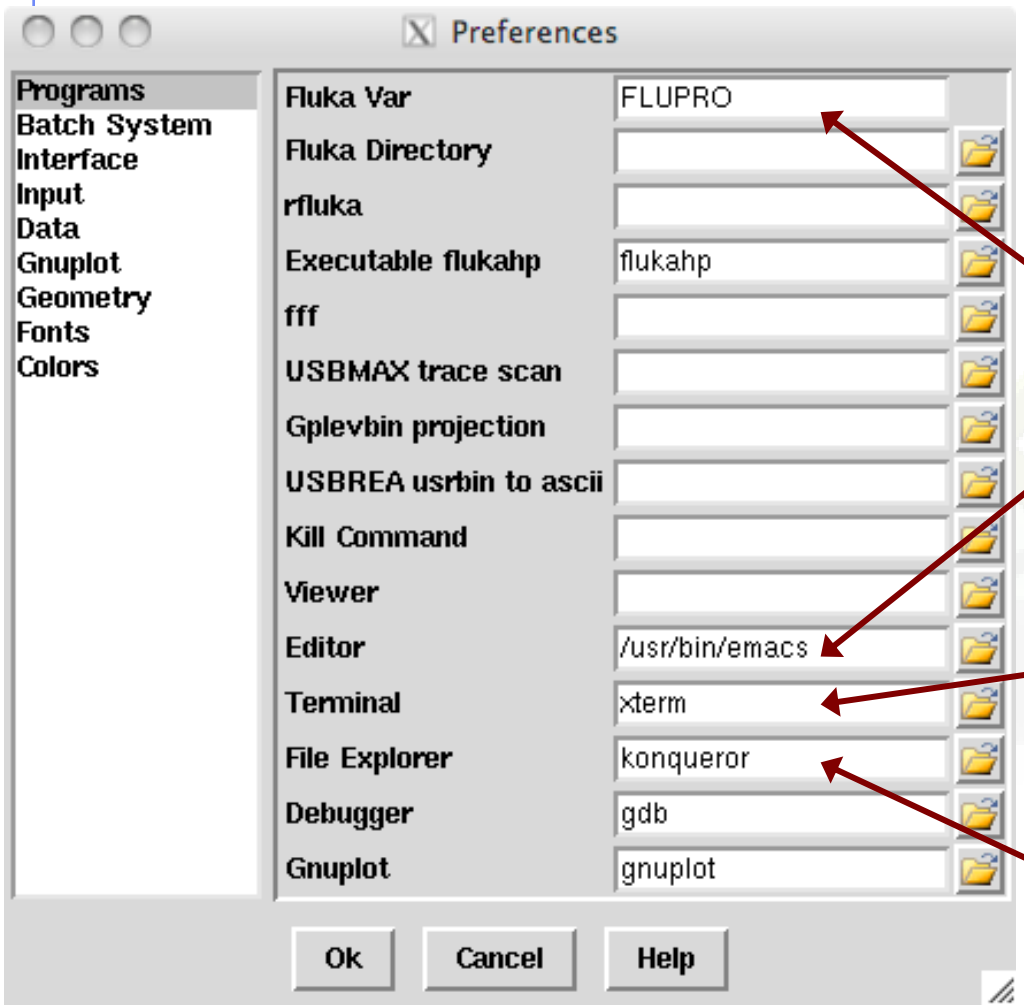
Toolbar

Move Up/Down

Quick Access to:

1. Project Frame
2. Input Editor
3. Geometry Editor (if installed)
4. Process Summary
5. Compile executables/Add user routines
6. Debug Geometry
7. Run/monitor simulations
8. View output files
9. Data merging
10. Plots
11. Databases (not yet functional)
12. Material Database
13. Help

Basic Preferences



The program tab of preferences allows the user to set the default programs and directories

Set your FLUKA directory, to override $FLUPRO

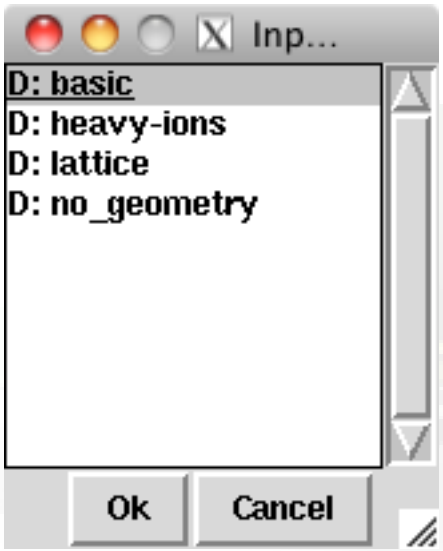
Set your favorite editor

Set your favorite console program (xterm, nxterm, kconsole…)

Set your favorite file browser for example:
- "konqueror" for Linux with KDE
- "open" for the finder in MacOS

Input Templates

- When requesting a new input or a new project flair will prompt to select an input template:



Default template: basic.inp

```
TITLE

GLOBAL                                              1.0       1.0
DEFAULTS                                                          NEW-DEFA
BEAM
BEAMPOS
GEOBEGIN                                                          COMBNAME
    0    0
* Black body
SPH blkbody    0.0 0.0 0.0 10000000.0
* Void sphere
SPH void       0.0 0.0 0.0 1000000.0
* Cylindrical target
RCC target     0.0 0.0 0.0 0.0 0.0 10.0 5.0
END
* Black hole
BLKBODY      5  +blkbody -void
* Void around
VOID         5  +void -target
* Target
TARGET       5  +target
END
GEOEND
* ..+....1....+....2....+....3....+....4....+....5....+....6....+....7..
ASSIGNMA    BLCKHOLE    BLKBODY
ASSIGNMA      VACUUM       VOID
ASSIGNMA      COPPER     TARGET
RANDOMIZ         1.0
START
STOP
```

- Flair default templates are prefixed with "D:"
- User templates will be prefixed with "U:"

The user can create his own set of input templates. They are normal FLUKA input files and they have to be placed in the directory ~/.flair/templates (create the directory if not existing)

Input Editor

- With the input editor the user can manipulate the input cards:
  - Add card to input;
  - Edit & modify existing ones;
  - Copy & Paste;
  - Clone (Duplicate);
  - Import from other input files;
  - Validate the correctness of the cards;
  - Error filtering;
  - Rearrange order;
- The editor will always try to rearrange the input cards (only if needed) to create a valid FLUKA input file.
  e.g. body cards outside the GEOBEGIN/GEOEND parts will be moved inside

Note: Automatic rearranging of cards cannot work if "**#include**" cards are present. The user have to do it manually.

Card Categories

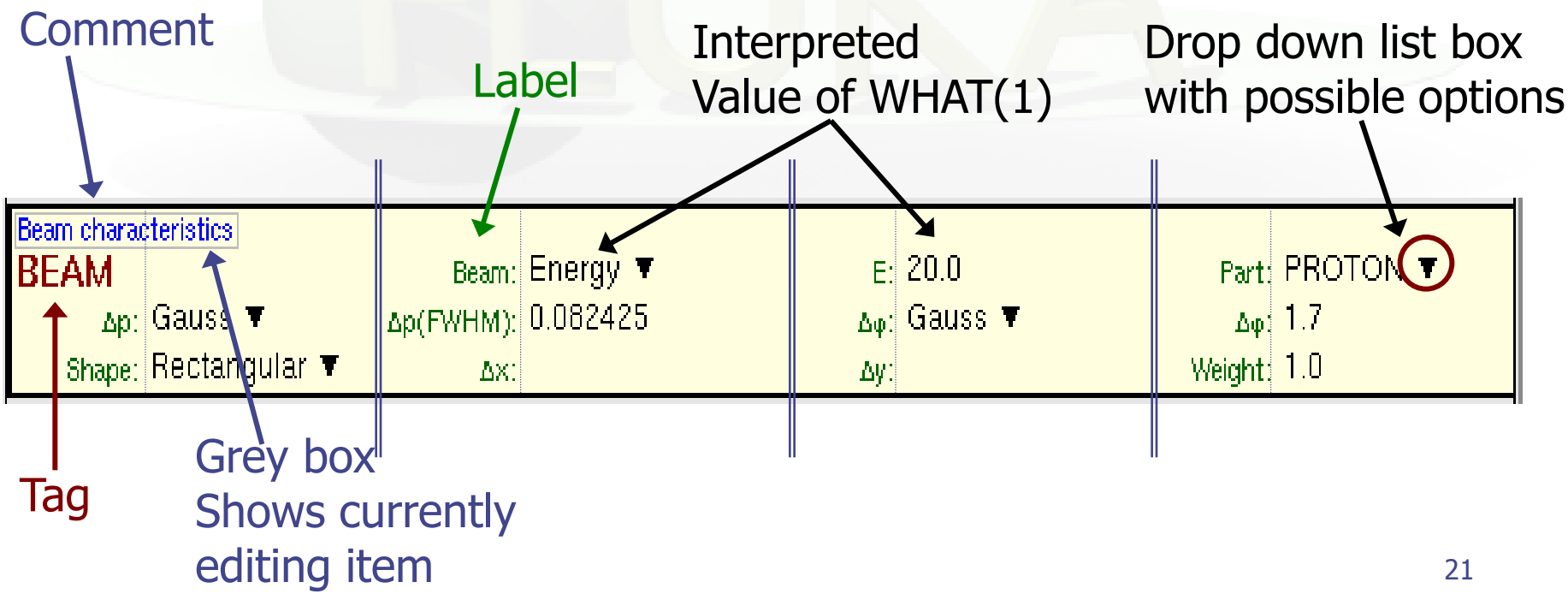
For easier access, cards are groups in the following categories:

- **General** General purpose (TITLE, DEFAULTS, GLOBAL…);
- **Primary** Definition of the primary starting particles;
- **Geometry** Cards related to the definition of the geometry bodies/regions/lattices plotting and rotations/translations;
  - **Bodies** Subcategory containing only the bodies definition;
  - **Transformations** Subcategory containing only the geometrical directives;
- **Media** Definition and assignment of materials;
- **Physics** Setting physics properties of the simulation;
- **Transport** Modify the way particles are transported in FLUKA;
- **Biasing** Cards for importance biasing definition;
- **Scoring** Cards related to scoring;
- **Flair** flair special cards;
- **Preprocessor** Definitions for creating conditional input files.

Anatomy of a card mini-dialog [1/2]

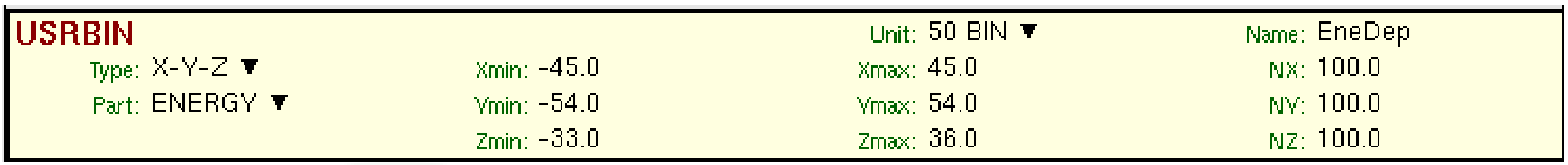
- For each extended card flair has a mini dialog (currently in 4 columns), interpreting all information stored in the card

```
* Beam characteristics
BEAM              -20.0 -0.082425      -1.7                          1.0PROTON
```

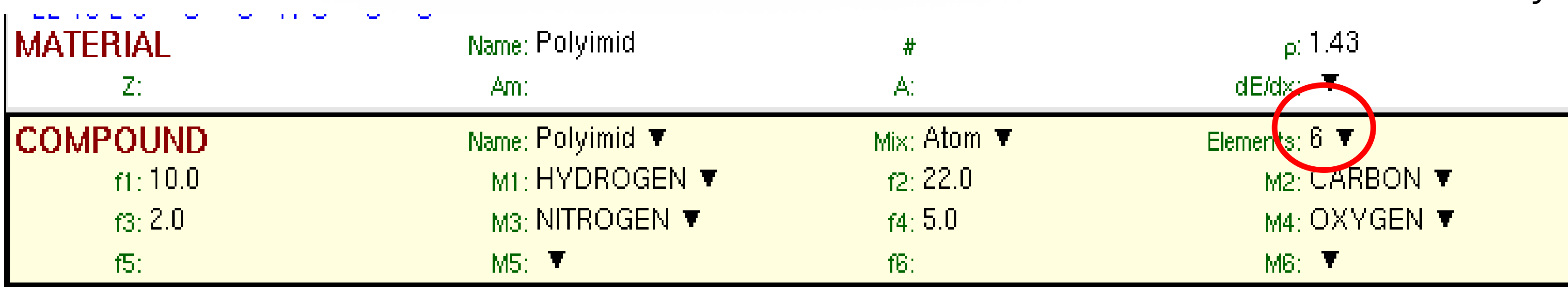


Anatomy of a card mini-dialog [2/2]

```
* Energy deposition in 3D binning
USRBIN          10.0    ENERGY     -50.0      45.0      54.0      36.0EneDep
USRBIN         -45.0     -54.0     -33.0     100.0     100.0     100.0&
```




```
* Polypyromellitimide Polyimide, Kapton
* Chemical       O = C   H-C   C = O
* Formula           / \  // \ / \       H-C - C-H         H-C - C-H
*                  /   \C    C   \     //     \\         //     \\
*            ---- N     |    ||   N - C         C - O - C         C ----
*                  \   /C    C   /     \     /           \     /
*  C  H  N O        \ /  \\ / \ /       H-C = C-H         H-C = C-H
*   22 10 2 5    O = C   H-C    C = O
MATERIAL                                  1.43                            Polyimid
COMPOUND        10.0  HYDROGEN      22.0    CARBON       2.0  NITROGENPolyimid
COMPOUND         5.0    OXYGEN                                           Polyimid
```



Validating input and Error correction

- Flair validates the input file while loading and each card during editing;
- Errors are highlighted in **red**;
- For the moment only syntactical errors are checked, and a few logical errors;
- Popup-menu option "Show errors" displays a short message on what is expected as correct value;
- Menu item "Input / Filter Invalid" shows only the invalid cards from the last filtered view.

Material Database

- Flair contains an internal database of ~500 predefined materials and/or compounds;
- Some (~300) with the Sternheimer parameters;

Please use these data as Reference only!

- Validate **always** the correctness of the data;
- If errors found please contact the author;

- The database can be edited, and populated with your own materials. In this case a local copy of the database will be made in ~/.flair directory.

Managing a projects

FLAIR allows to:

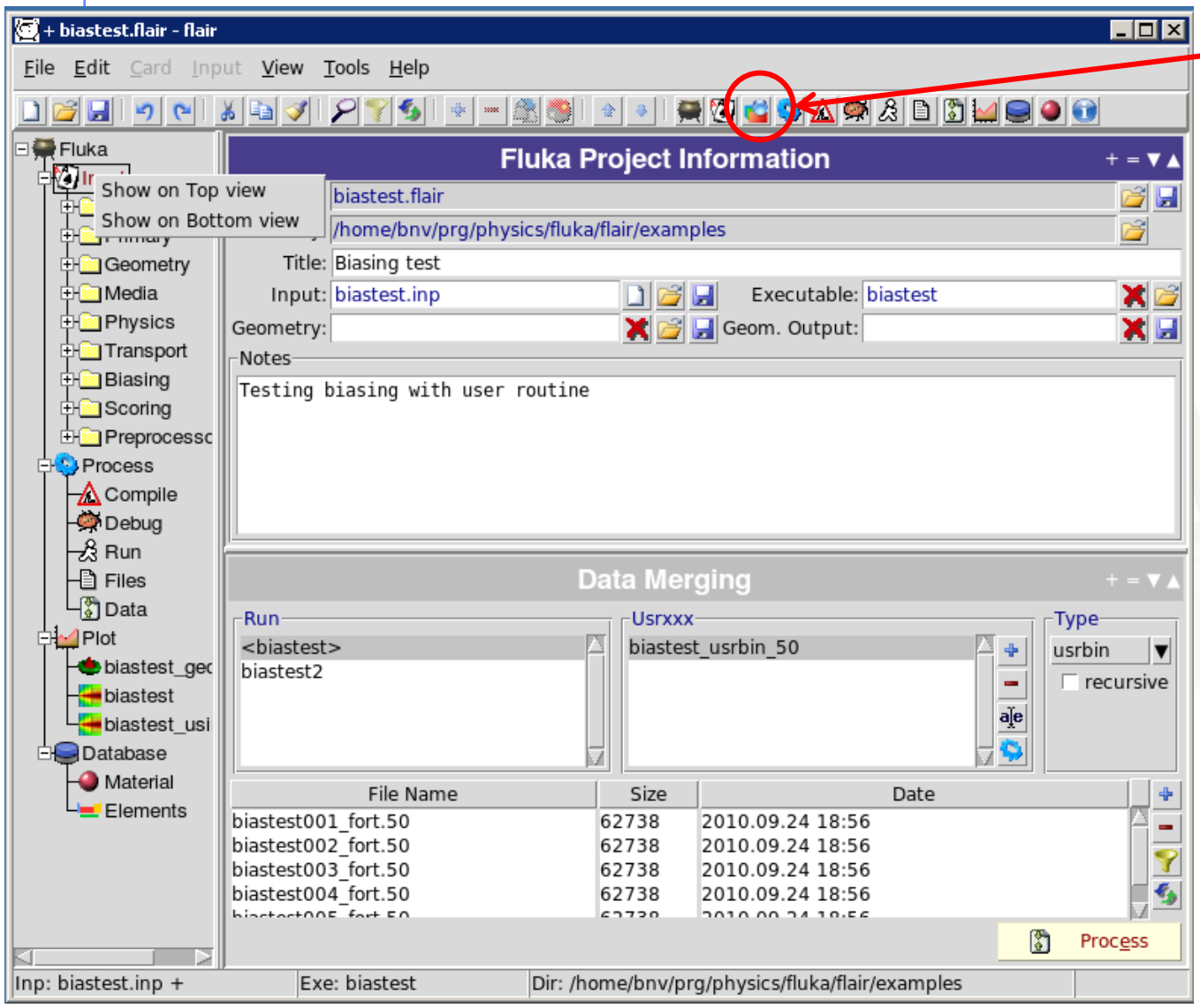
- Attach, Edit, Compile User routins and link to a FLUKA executable
- Monitor the behaviour of a run
- Manage, inspect, edit all output files
- Manage post-processing of output files
- Produce and export Plots of results

Other goodies

Flair has a lot of functionality that is not covered in this tutorial:

Geometry Editor to view, build and debug Geometry

Starting the Geometry Editor



Click on icon

or from Menu
→ View
→ Geometry Editor

or with
[F4] shortcut

- Either start flair with option -g

Geometry editor

- Working on 2D cross sections of the geometry;
- Interactive visual editing of the geometry in 2D;
- Debugging bodies/regions in a graphical way;
- Fast 3D rendering of the geometry;

Pros

- Fast display of complex geometries;
- Many user-customizable layers;
- Graphical editing of the bodies with snapping mechanism to generate accurate coordinates;
- Visual selection and editing of zones w/o the need to know the orientation of bodies;
- Use full analytical curve of bodies with no conversion to vertices/edges;
- Interactive debugging with information of problematic bodies, regions and/or zones;

Cons

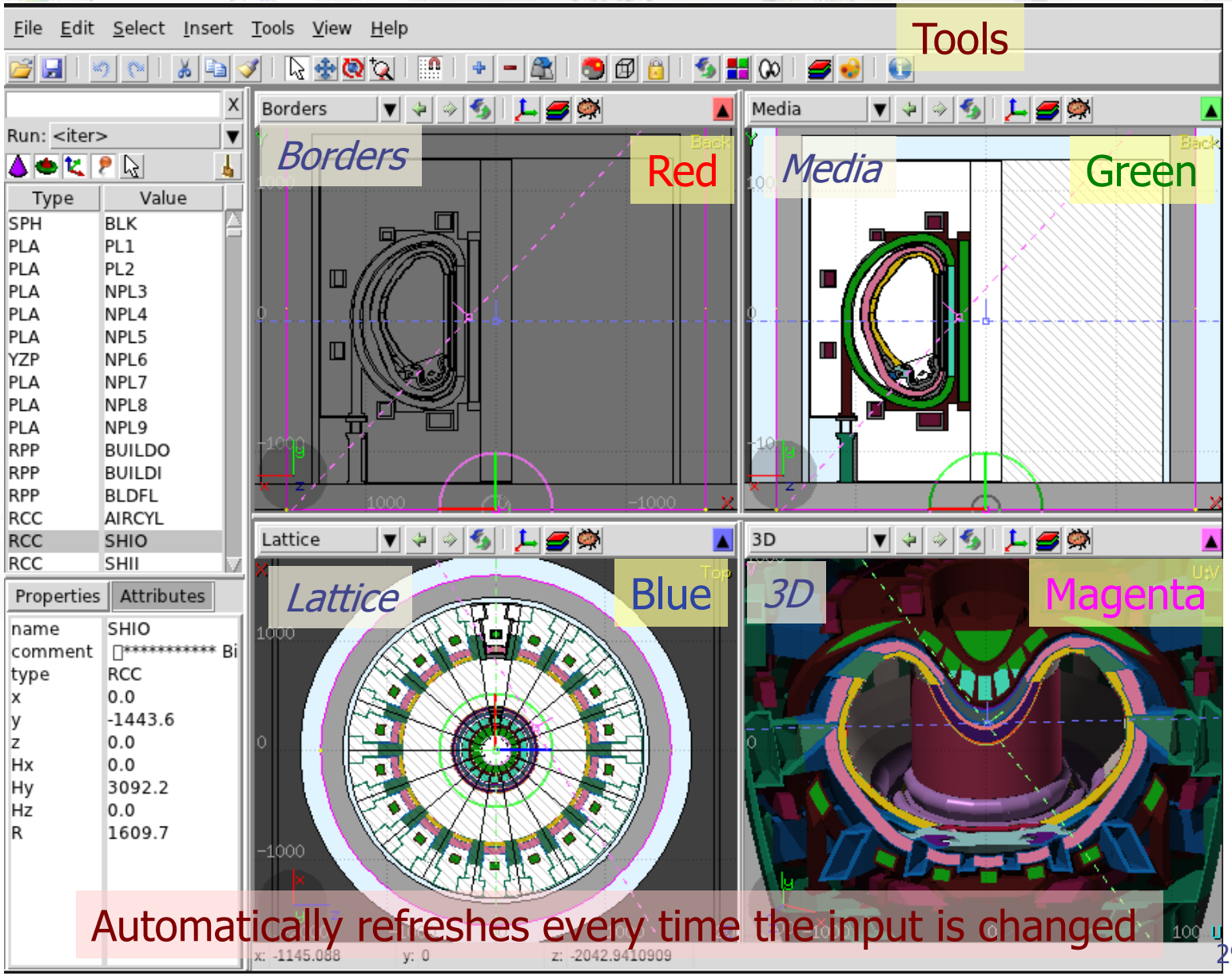
- Tricky to orientate in an unknown geometry;
- Difficult to find region using the expression;

Geometry Editor: Interface

Filter

Filtered Objects

Properties & Attributes



Automatically refreshes every time the input is changed