Introduction to Flair

Beginners' FLUKA Course

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*]

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?

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

What is flair [1/2]

- **FLUKA Advanced Interface** [http://www.fluka.org/flair]
- **All-in-one** User friendly graphical Interface
- With 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:** transformation, optimizations and debugging
- **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 viewer dividing into sections
- Post processing (merging) the output data files
- Plot generation through an interface with gnuplot
  Input information, USRxxx, RESNUCLEI and geometry
- 3D photo-realistic images with PovRay (ToDo)

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 (ToDo)
- Programming python **API**

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 procedure
  - RPM/DEB method: 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: Manual installation on platforms like (MS-Windows, Mac OS…). Please follow the instructions on the 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

- Click the icon of Flair from the programs menu. Flair is registered under the Science/Physics category
  Note: freedesktop however places this category in various places depending your linux distribution and window manager
  Typical places:
  - Applications
  - Education
  - Edutainment / Science
  - Scientific
  - Other

Window Manager (only via RPM or DEB installation):

- Flair makes an association of the following extensions:

 *.flair  *.fluka, *.inp

Console

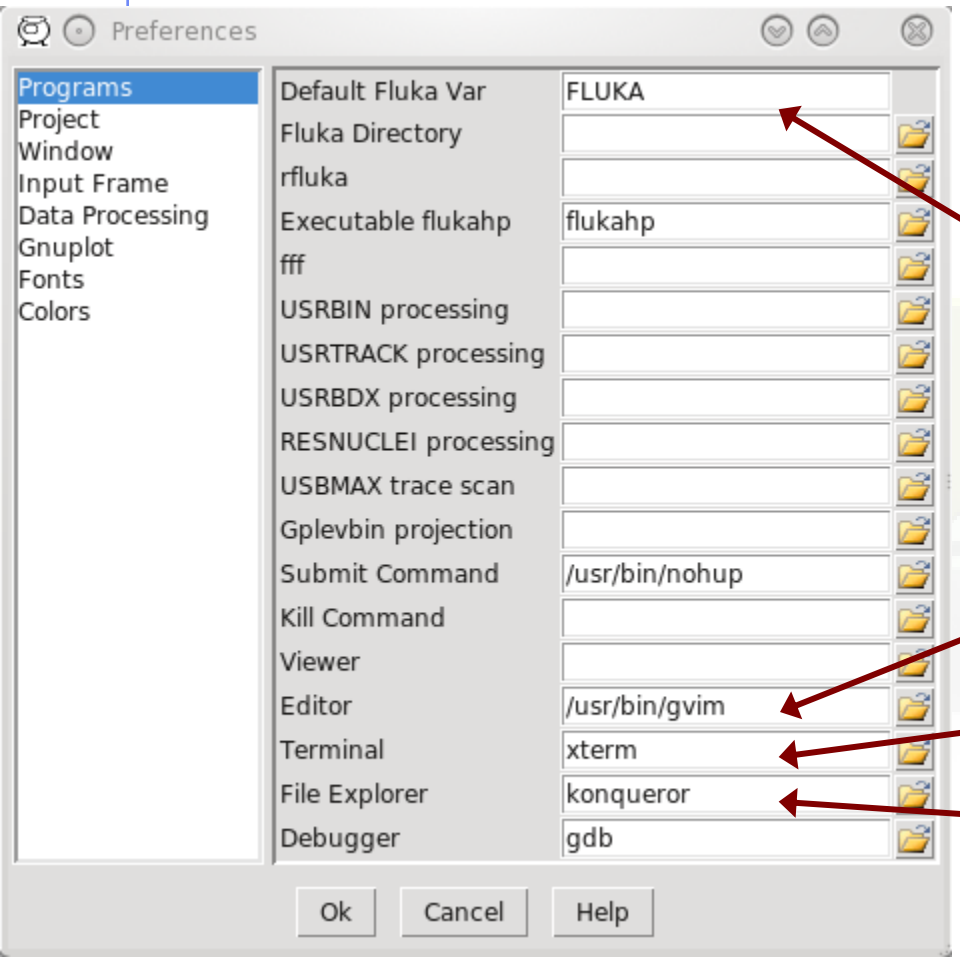
- Type the command /usr/local/bin/flair
- It is recommended to place in your PATH the /usr/local/bin directory

Startup

During startup flair will perform the following:

- Open an "xterm" to be used as the Output window
- Display the About Dialog
- Check for the existence of FLUPRO variable, and if not found will open the Preferences dialog to set explicitly the FLUKA path.
  WARNING: Window managers (GNOME, KDE…), as well command shells (bash, tcsh, ash…) have a different configuration file where they expect the environment variable.
- Display the "Tip of the Day" dialog
- Open the "Check for Updates" dialog (every 30 days interval)

Basic Preferences



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

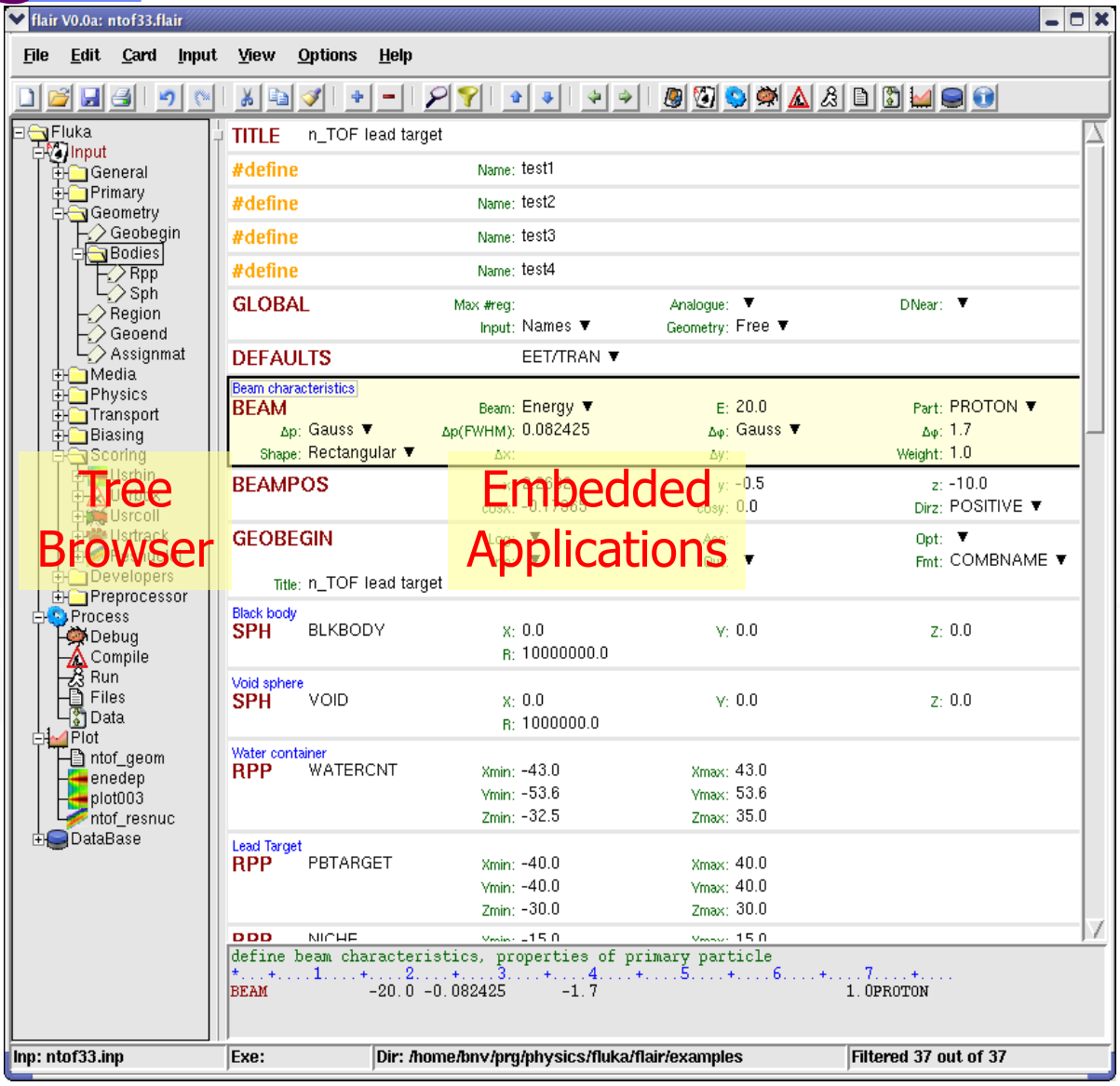
Set your FLUKA directory, to override $FLUPRO

Set your favorite editor

Set your favorite console program

Set your favorite file browser

Program Interface



Wrapper of standalone applications

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 cut: F10 or Alt-*F, E, C, I, V, T, H*

Toolbar

Undo/Redo

Find/Filter

Project Control I/O

Cut/Copy/Paste

Add/Del/Clone/Toggle

Move Up/Down

Quick Access to:

1. Project Frame
2. Input Editor
3. Process Summary
4. Compile
5. Debug Geometry
6. Run
7. View output files
8. Data merging
9. Plots
10. Databases
11. Help

Input Templates

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

- 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 (create the directory if not existing)

~/.flair/templates

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 will be moved inside

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
- **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
- **Developers** — *…reserved for FLUKA developers…*
- **Preprocessor** — definitions for creating conditional input files

Concepts: Extended Cards [1/2]

- Flair is treating the input file as a list of extended cards
- Each extended card contains:
  - Comment: All commented lines preceding the card(s) as well the inline comments
  - Tag: The 8 character word identifying the card. All tags not recognized by flair will be converted to #error
  - Whats: Multiple number of whats (0=sdum, 1-6 first line, 7-12 continuation line...)
  - Extra: multi line string of extra information for special cards like REGION, TITLE, PLOTGEOM etc.
  - State (Enable/Disable)

Concepts: Extended Cards [2/2]

- Flair has introduced a couple of special cards to homogenise the FLUKA input file representation:
  - REGION, referring to a region declaration in geometry
  - COMPOUND, all compound cards related to one material are joined in one card
  - END card for bodies/regions is no longer required

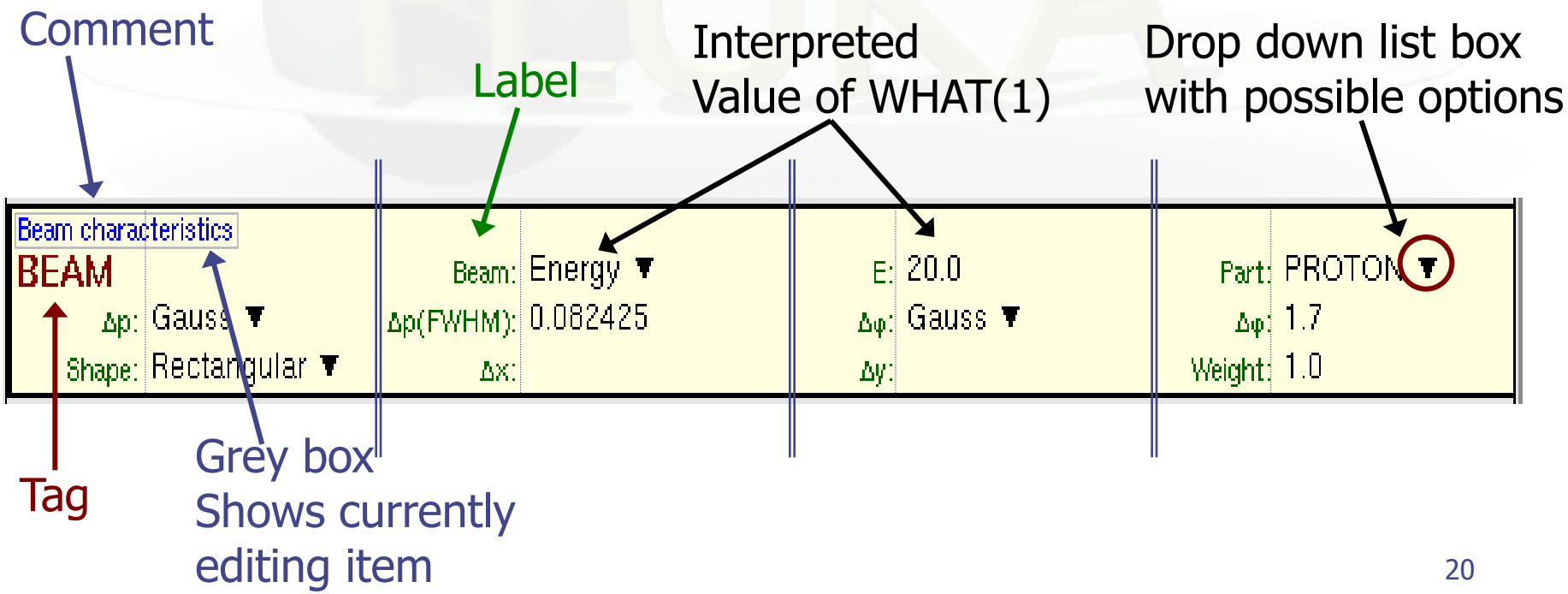
- Cards can be edited with the flair editor through the use of the minidialogs, forcing the user to enter the "correct" information. Or through the Edit dialog (Ctrl-E) that the user has full control of the card

- Flair will try to find the best floating point representation of each number, to ensure the maximum accuracy; number of digits that fits in the specific width (10 for the fixed format, 22 for the free format)

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

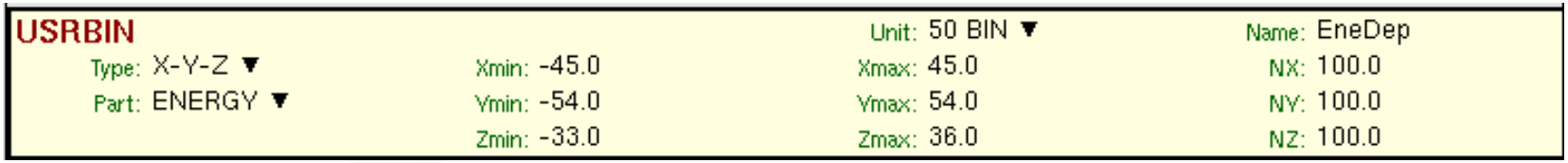
- For each extended card flair has a mini dialog (currently in 4 columns), interpreting all information stored in the card

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

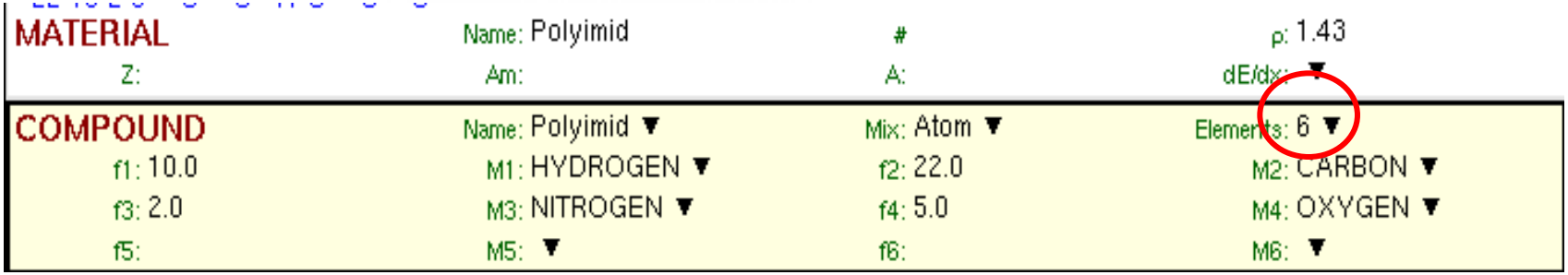


Anatomy of a card mini-dialog [2/2]

```
* 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
```



Editing Cards

While in input editor you are working in two modes

1. Card mode: where you manipulate the cards as a unit. Copy, paste, delete, change order of cards
2. Edit mode: where you manipulate the contents of the card. Edit mode is activated immediately after adding a new card or by hitting Enter or with the mouse click. To leave edit mode click the Esc or with the mouse click somewhere else.
   The active item (what) is highlighted with a grey rectangle and highlighted also in the card viewer below the editor.

- Shift+Mouse or Shift+Up/Down arrows selects a range of cards

Validating input and Error correction

- flair validates the input file while loading and each card during editing.
- Errors are highlighted with **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 / 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

Starting a Run

- Flair can start a single run based on the input file, or multiple runs by overriding some options, like #defines, title, random number seed and number of starting particles
- Flair will try to "attach" to a run. Using only the information from the output files generated by FLUKA, flair will try to identify the directory where the run takes place and monitor the progress of the Run.
- During the execution of the run the user can view the output files in the "Files Frame"

Tips & Tricks

- Mouse
  - right-click — opens the popup-menu with the most important actions
- Keyboard — Check the accelerators on the menus
  - Ctlr-Enter — Performs the default action in every frame. Add a card in the Input Editor
  - Ctrl-Space — Access popup-menu (like right-clicking)
  - Listboxes — All listboxes in flair are searchable and case insensitive. Type the first characters of the string you are searching and the closest match will be highlighted. Ctrl-G repeats the search. Space selects/deselect item
  - +, -, Ins — While editing the REGION expression shows a list of all available bodies

Known Bugs / Limitations

- Unicode / International characters do not work well and should be avoided
- Gnuplot <4.2 has a bug in the number of palette colors, and on the cblabel for the wxt terminal
- Inline comments, and comments inside REGION definition are treated as one comment preceding the input card
- REMEMBER always that the .flair and .inp are different thing. Do not save the project as .inp or the input file as .flair

Other goodies

Flair has a lot of functionality that is not covered in this tutorial. I would advice the users to go through the various menus and help page and try it out.

FLUKA

Software choices [1/2]

Requirements:

- Open source software
- Multiplatform with easy installation
- Minimum requirements on other package
- Large community of users and years of development

Python [http://www.python.org]
is a scripting language which is:

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

Software choices [2/2]

- Tkinter [http://wiki.python.org/moin/TkInter]
  default GUI toolkit for Python.
  Good for simple UIs.
  Portable, wrapper around tk/tcl

- Gnuplot [http://www.gnuplot.info]
  is a command-line driven, interactive function plotting program specially suited for scientific data representation. Gnuplot can be used to plot functions and data points in both two and three dimensions and in many different formats.

- Povray [http://www.povray.org]
  POV-Ray™ is short for the Persistence of Vision™ Raytracer, a tool for producing high-quality computer graphics. POV-Ray™ is copyrighted freeware. POV-Ray is the worlds most popular raytracer.

Version Numbering

- Flair version numbering, has the form: **flair-M.m-R**
- **M: Major version**, this number is increased by one only when a major modification in the structure of the program takes place. During the initial phase of development is 0.
- **m: Minor version**, this number is increased by one every time an addition in the program's functionality is made. e.g. Adding new plotting forms, mechanism, databases etc.
- **R: Patch level**, this number is increased when bug fixes take place or minor changes in the functionality. e.g. Addition of extra fields in a form etc.
- The About Dialog of the program displays the Major and minor version as well the CVS release number, every time changes in the program are committed to the CVS server.
- The Update dialog of flair will ask the user to check on the web every 30 days (default) for a new flair version (M.m) and FLUKA release.

How to Contribute

- Python programming
  - Parsing and processing output files
  - Web based database for sharing resources with other users
  - ...
- Input Editor
  - Cards Layout, in other formats (from 3 up to 8 columns)
  - Labels have to be intuitive, if something is not comprehensible please propose an alternative
- Manual, Online documentation, Tips database
- Icons for tool bars and cards
- Gnuplot scripts or ideas for better presentation
- Comments & Ideas, on new features that one wants to see
- **Testing, Bug reporting**
- ...

ToDo: Features to be added

- Interface
  - Exportation of processing scripts and various formats (Geant...)
  - Import from other codes (MCNP…)
- Input Editor
  - A group of flair cards for manipulating the input file. Including dynamic transformations, variables, vector operations, etc.
- Post Processing
  - Re-binning or USRBINs
  - Maximum trace
- Plotting:
  - Information of Input File
  - 3D Ray Tracing interface to povray
  - Particle tracks