Flair Advanced Features

Advanced FLUKA Course
/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]
Mouse:
left opens on active
right selects where to view

2 working frames

active
+ vertical/horizontal
= equalize
▼ minimize
▲ maximize

input modified and not saved
Interface

Keyboard:
- Almost everything is possible with the keyboard see manual for shortcuts
- Ctrl-Enter: Execute most important action
- Ins/Del: Add or Delete

Mouse:
- right-click anywhere to get a popup menu

Listboxes:
- all listboxes are searchable. Typuing only the characters (A-Z) and numbers (0-9) all other are ignored

LabelFrames:
- can collapse/expand by clicking on the label
Anatomy of a card mini-dialog

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

<table>
<thead>
<tr>
<th>Beam characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>BEAM</td>
</tr>
<tr>
<td>-20.0</td>
</tr>
<tr>
<td>-0.082425</td>
</tr>
<tr>
<td>-1.7</td>
</tr>
<tr>
<td>1.0 PROTON</td>
</tr>
</tbody>
</table>

Comment
Tag
Grey box Shows currently editing item

Label
Interpreted Value of WHAT(1)
Drop down list box with possible options
highlight differences during editing
- Drag’n’drop from the TAG of the cards
- Double click on card TAG to select all similar cards
- Editing multiple cards: select cards and modify the value in one card will propagate the change to all similar selected cards
- Ctrl-Duplicate-Click Show/Hide selected cards
- #if..#endif, $transform, $translat or $expand flair will enclose the selected cards with the #if #endif, or $start_xxx, $end_xxx transformation cards
- Popup Balloon tooltip displays short help:
  - for every option on every card
  - body description in the REGION expression
- Right-click: shows popup-menu
  - Quick filtering by REGION, MATERIAL, scoring etc...
- Easter Eggs: AWARI by Double-Right-Click on dialog showing the card representation as text at the bottom of the screen
Input Editor - 3

- Automatic indentation of nested #if..#endif and $start..$end directives.
- To refresh the display type Ctrl-R
- Each REGION can be split into many cards if needed to be used with preprocessor commands.
- Use as a name “&”

<table>
<thead>
<tr>
<th>Void around</th>
</tr>
</thead>
<tbody>
<tr>
<td>REGION VOID</td>
</tr>
<tr>
<td>Expr: +void -target</td>
</tr>
<tr>
<td>#if BIAS ▼</td>
</tr>
<tr>
<td>REGION &amp; cont: -bias</td>
</tr>
<tr>
<td>#endif</td>
</tr>
</tbody>
</table>
Filter Cards dialog allows a more advanced selection of cards to be displaced, by showing only the cards that match the selected options.
Manual editing of the card
Scan input and display only active cards (not excluded by the preprocessor)
Show cards containing problems/errors
Display a message with the errors identified in the cards
Open the geometry transformation dialog
Expand parenthesis in the region (only logical optimization will performed)
Manual Card Editing

Accessible: **Ctrl-E, right-click → Edit, Menu → Input → Edit**

Lines: How many lines the card extends

Extra: additional information for a card like title string for TITLE, or region expression for REGION

Dropdown box: shows with categories all items defined in the input (bodies, regions, materials, particles...)

![Edit Card interface with primary particle definition and BEAM card example]
Bodies Transformation

Transformation Types:
- T: translate along a vector
- TX, TY, TZ: translate along axis
- RX, RY, RZ: rotation (degrees)
- S: scaling

- Applies a user transformation to the selected bodies on the input editor.
- Convert transformations to/from ROT-DEFini cards
- Zero: limit below which to be considered as zero
- Accuracy: Numeric digits
- Infinite: infinite bodies when converted to which size to use
- Use QUA: convert infinite cylinders to infinite QUAdrics

Remember:
- When transforming bodies for use with LATTICE card, use the maximum precision
Color Palette

Accessible: Menu → View → Palette

- Edit colors used for material display in Geometry plots and GeometryEditor
- Global colors are saved inside flair.ini and are shared between all projects
- Local colors are initially randomly assigned and saved inside the project file
Compiling

Filetypes accepted:
- Fortran: .f, .F, .for, .FOR
- C/C++: .c, .cpp, .cxx, .cc
- Libraries: .a, .so

Automatic scanning of necessary user routines and copying them to project folder.

Build: behaves like a "makefile" compiles based on files timestamp when are newer
Compile: Forces compile of the selected files
Clean: cleanup of all produced files

When you are unsure, click on “Clean” before “Build”
Running

- Monitors the status of the run by inspecting the FLUKA output files. If `timeout` occurs try to re-Attach to the running process.
- The timeout is user-definable in the Preferences dialog.

`<inputname>` refers to the input file AS IT IS in the input editor.
Create additional runs based on the same input file by overriding:
- Title
- Preprocessor definitions
- Random number seed
- Starting particles
- Execution timeout
- Executable
Running: How to use multicore CPU’s

- Create clones of the current input e.g. `test.inp` named: `test1.inp, test2.inp, test3.inp` ...
- Assign a different random number seed on each run (Rnd entry)
- Select all in the listbox and click Run

Multiple Selection:
- To modify many runs at the same time, select them in the listbox
- The options will be “disabled”
- Right-click on the options you want to enable and modify them
- Modify the filters in Data processing for summing up all cycles from all runs (see later)
Output Files

Inspect Output files generated by FLUKA classified per:

- Run/Cycle

As well special output files from compilation data processing plotting and temporary

Double clicking opens:
- Files in the file Viewer
- coredumps in debugger

Right click can convert USRBIN’s from formatted to unformatted

Delete selected files
Data Processing

Process all scoring BINARY output files for each Run.
Name rules are defined in Preferences

Automatically scan input for scoring cards

+/- Modify file list by adding / removing items

Dialog for editing scanning rules for files.

Use the rules to merge from multiple runs. e.g. add a \d in the target like +\d\d\d\d\d\_fort.\U

To modify the rules for multiple scoring cards, select all Usrxxx before

The default rules can be modified in the Preferences Dialog
### Plot List

#### Plot List

<table>
<thead>
<tr>
<th>File</th>
<th>Title</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>geometry</td>
<td></td>
<td>Geometry</td>
</tr>
<tr>
<td>enedep</td>
<td>Deposited Energy</td>
<td>USRBIN</td>
</tr>
<tr>
<td>fluence</td>
<td>Particle Fluence</td>
<td>USR-1D</td>
</tr>
<tr>
<td>resnuc</td>
<td>Residual Nuclei</td>
<td>RESNUCLE</td>
</tr>
<tr>
<td>USRBIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>USR-1D</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RESNUCLE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Plots can be created in the “Plot” list frame. Either Add new plots or Clone from existing ones.**
- **It is important to set a unique filename for each plot. This filename will be used for every auxiliary file that the plot needs (the extension will change)**
- **The Filter button creates automatically one plot for each processed unit**
- **Hit Enter or click the Edit icon to display the plotting dialog**
- **Fast Double click on item to open the corresponding dialog**
- **Slow Double click to modify the value**

#### Plot Types

- **Geometry** For geometry plots
- **USRBIN** For plotting the output of USRBIN
- **USR-1D** To plot single differential quantities from cards USRBDX, USRTRACK, USRCOLL, USRYIELD
- **USR-2D** To plot double differential from USRBDX
- **RESNUCLE** To plot 1d or 2d distributions of RESNUCLEi
- **USERDUMP** To plot the output of USERDUMP. Useful for visualizing the source distribution (ToDo)
Plotting Frames

All plot types share some common fields:
Title + options, Filename, Axis Labels, Legends (Keys) and Gnuplot Commands.

Plot button (Ctrl-Enter) will generate all the necessary files to display the plot, ONLY if they do not exist.

Re-Plot will force the creation of all files regardless their state.

Check the gnuplot manual to provide additional customization commands: e.g. To change the title font to Times size=20, add in the Opt: field the command: font ‘Times,20’

Look in the flair manual for a short reference of gnuplot commands.
General Tips

- In the Configuration Dialog you can set global commands to execute before or after any plot.
- The output window displays all the commands that are sent to gnuplot. As well as the errors. In case of problem always consult the output window!
- In the Gnuplot commands you can fully customize the plot by adding manually gnuplot commands:
  - Special commands:
    - `plot, splot` with no options, defines the order where flair should insert the plot or splot command.
    - `replot <plot-cmd>` append extra plots to the one generated by flair.
**USRBIN Plots -1**

### Binning Detector
- **File:** ntof33_usrbin_50
- **Title:** n_TOF lead target
- **Cycles:** 5
- **Primaries:** 15900
- **Weight:** 15900.0
- **Time:** **** Sum file ****

### Binning Info
- **Det:** 1 EneDep
  - **X:** [-40 .. 40] x 100 (0.8)  
    - **Min:** 1.95034673E-07
  - **Y:** [-40 .. 40] x 100 (0.8)  
    - **Max:** 0.0254351143
  - **Z:** [-30 .. 35] x 100 (0.65)  
    - **Int:** 11.0419018

### Projection & Limits
- **Type:** 2D Projection
  - **Color Band**
    - **Min:**
    - **Max:**
  - **CPD:** 5
  - **Colors:** 30
  - **Palette:** FLUKA
  - **Round:**

### Gnuplot commands
- **Rebinning**
- **Swap axes**
- **Get limits from gnuplot using right-mouse**
- **Draw errors. (combined with log)**
- **Correct only if one slice is used**
Normalization could be used as:

- number or expression evaluating in a number $65e-3/2.7$
- function with $x$ as variable. e.g. $E2T(x*65e-3/2.7)-293$

with the function defined in the Gnuplot commands:

$$E2T(x) = ((3.00629e-08*x-0.000108436)*x+1.01097)*x+311.839$$
Normalization could be plotted:

- 2D projection, 1D projection
- Trace of the maximum
- Full width at half maximum
Geometry plot overlay (useful for LATTICE’s):

- **Auto** - generates automatically from FLUKA a geometry at the middle position of the projection

otherwise you can use any existing geometry plot from the drop down list.

Be careful to properly match the axes that you are using.
Configuration Dialog: Programs

- Set FLUKA directory
- Override default programs to use
- Processing programs are in the “Data” section
Configuration Dialog: Interface

- General interface settings
- Keep backups when files are saved as (file~)
- Automatically Cleanup temporary files. Disable only if you want to inspect files after Debug or Plot when an error occurs
- Key time to reset the type-in search in listboxes
- Balloon delay time
- Time format for files (follows python&C syntax)
- Time out to attach to a running simulation
- Automatic refresh interval of information
Configuration Dialog: Input Editor

- Show alignment scale
- Automatically insert comment
- Always display preprocessor cards
- Enable drag’n’drop
- Automatic body insertion while editing the region expression
- Sort the region and material list
- Display card interpretation at the bottom of the screen
Define how to generate the automatic filenames

\( \backslash I \) will be replaced by input

\( \backslash T \) by card name

\( \backslash t \) by card character

usrbdx \( x \)

usrbin \( b \)

usrcoll \( c \)

usrtrack \( t \)

usryield \( y \)

resnuclei \( r \)

\( \backslash U \) the abs(unit-number)
Configuration Dialog: Gnuplot

Terminal:
additional options to supply to default terminal

Global Commands:
gnuplot commands to be executed before any plot

File Types:
Right-click: to Add/Delete/Modify file types.
Configuration Dialog: Geometry

Laptop Mode:
check to swap middle with right mouse buttons. Middle button is used in GeometryEditor for panning, zooming, rotating etc...

Zero:
Infinite:
Accuracy:
same as in the Bodies Transformation dialog
### Materials Database

**Material Database**

<table>
<thead>
<tr>
<th>Material</th>
<th>Density</th>
<th>Stoichiometry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mercury</td>
<td>13.546</td>
<td>Hg</td>
</tr>
<tr>
<td>728 Cyclohexanone</td>
<td>0.9478</td>
<td>H-10, C-6, O-1</td>
</tr>
<tr>
<td>Skeletal Muscle (W&amp;W type 1)</td>
<td>1.05</td>
<td>H-10.1, C-17.1,</td>
</tr>
<tr>
<td>Lead</td>
<td>11.35</td>
<td>Pb</td>
</tr>
<tr>
<td>Thallium</td>
<td>11.72</td>
<td>Ti</td>
</tr>
<tr>
<td>Cyclobutane</td>
<td>0.00125</td>
<td>H-8, C-4</td>
</tr>
<tr>
<td>1-Chlorobutane</td>
<td>0.8862</td>
<td>H-9, C-4, Cl-1</td>
</tr>
<tr>
<td>Sodium nitrate Na_N_O3</td>
<td>2.261</td>
<td>N-16.5, O-56.5,</td>
</tr>
</tbody>
</table>

**Material Properties**

**Title:** Mercury

**Notes:**

**Names:**

**Stoichiometry**

<table>
<thead>
<tr>
<th>Composition:</th>
<th>mass</th>
<th>liquid</th>
<th>13.546</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z</td>
<td>A</td>
<td>El</td>
<td>Name</td>
</tr>
<tr>
<td>80</td>
<td>Hg</td>
<td>Mercury</td>
<td>1.0</td>
</tr>
</tbody>
</table>

**Properties**

**Group:** Elements

---

**WARNING:** When modifying the database a local copy will be created in `~/.flair` folder!!!
Periodic Table

Hg

Atomic Weight: 200.59 (2)
Density: 13.546 g/cm³
Melting: -38.83 °C
Boiling: 356.73 °C
Oxidation: +1, +2

Table of Elements

<table>
<thead>
<tr>
<th>Group</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>Period</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>H</td>
<td></td>
<td></td>
<td>Li</td>
<td></td>
<td>Be</td>
<td>8</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>Li</td>
<td>4</td>
<td>Be</td>
<td>11</td>
<td>Na</td>
<td>12</td>
<td>Mg</td>
<td>19</td>
<td>K</td>
<td>20</td>
<td>Ca</td>
<td>37</td>
<td>Rb</td>
<td>38</td>
<td>Sr</td>
<td>55</td>
<td>Cs</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>73</td>
<td>Tb</td>
<td>74</td>
<td>W</td>
<td>75</td>
<td>Re</td>
<td>76</td>
<td>Os</td>
<td>77</td>
<td>Ir</td>
<td>78</td>
<td>Re</td>
<td>87</td>
<td>Fr</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>71</td>
<td>Lu</td>
<td>72</td>
<td>Hf</td>
<td>73</td>
<td>Ta</td>
<td>74</td>
<td>W</td>
<td>75</td>
<td>Re</td>
<td>76</td>
<td>Os</td>
<td>77</td>
<td>Ir</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>39</td>
<td>Y</td>
<td>40</td>
<td>Zr</td>
<td>41</td>
<td>Nb</td>
<td>42</td>
<td>Mo</td>
<td>43</td>
<td>Tc</td>
<td>44</td>
<td>Ru</td>
<td>45</td>
<td>Rh</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>77</td>
<td>La</td>
<td>58</td>
<td>Ce</td>
<td>59</td>
<td>Pr</td>
<td>60</td>
<td>Nd</td>
<td>61</td>
<td>Pm</td>
<td>62</td>
<td>Sm</td>
<td>63</td>
<td>Eu</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>89</td>
<td>Ac</td>
<td>90</td>
<td>Th</td>
<td>91</td>
<td>Pa</td>
<td>92</td>
<td>U</td>
<td>93</td>
<td>Np</td>
<td>94</td>
<td>Pu</td>
<td>95</td>
<td>Am</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>103</td>
<td>Lu</td>
<td>104</td>
<td>Yb</td>
<td>105</td>
<td>Dy</td>
<td>106</td>
<td>Ho</td>
<td>107</td>
<td>Er</td>
<td>108</td>
<td>Tm</td>
<td>109</td>
<td>Yb</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>117</td>
<td>Lr</td>
<td>118</td>
<td>Rf</td>
<td>119</td>
<td>Db</td>
<td>120</td>
<td>Sg</td>
<td>121</td>
<td>Bh</td>
<td>122</td>
<td>Hs</td>
<td>123</td>
<td>Mt</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>175</td>
<td>Hg</td>
<td>176</td>
<td>Tl</td>
<td>177</td>
<td>Pb</td>
<td>178</td>
<td>Bi</td>
<td>179</td>
<td>Po</td>
<td>180</td>
<td>At</td>
<td>181</td>
<td>Rn</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>176</td>
<td>Tl</td>
<td>177</td>
<td>Pb</td>
<td>178</td>
<td>Bi</td>
<td>179</td>
<td>Po</td>
<td>180</td>
<td>At</td>
<td>181</td>
<td>Rn</td>
<td>182</td>
<td>Rn</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>177</td>
<td>Po</td>
<td>178</td>
<td>Bi</td>
<td>179</td>
<td>Po</td>
<td>180</td>
<td>At</td>
<td>181</td>
<td>Rn</td>
<td>182</td>
<td>Rn</td>
<td>183</td>
<td>Rn</td>
</tr>
</tbody>
</table>

** Lanthanides

** Actinides

H - gas
Li - solid
Br - liquid

Non-Metals
Transition Metals
Rare Earth
Alkali Metals
Alkali Earth Metals
Other Metals

At 20 C.
Import / Export

**Importing**
- **Input**: merge parts or entire input file with the current
- **Mcnp**: import mcnp geometry into FLUKA. (experimental)

**Exporting**
- **Gnuplot**: save active plot to a gnuplot script
- **Makefile**: create a makefile for compiling the executable
- **Mcnp**: save input in MCNP format: Geometry, Materials, Importances
- **Povray**: save geometry into povray 3D format
Geometry Editor 2D

- Debugging and editing bodies/regions in a graphical way
- Working on 2D cross sections of the geometry. Not a real problem since most of the objects are 2D extruded in the 3rd dimension

**Pros**

- Fast display of complex geometries
- Visual selection and editing of zones
- Use real curve of bodies with no conversion to vertices/edges
- Interactive debugging with information of problematic body regions and zones
- No use of any additional hardware (plain X11 libraries)

**Cons**

- No interactive 3D display
- Blind in 3rd dimension [could be compensated with raytracing]
- Tricky to orientate in an unknown geometry
Geometry Editor: Interface

Filter

Filtered Objects

Properties

Automatically refreshes when the input is changed
Geometry Editor: Mouse / Keyboard

General:
- **Ctrl** “controls” or changes the action
- **Shift** aligns to grid
- **Escape** cancels the active action

Mouse:
- **Left button**: User selectable action from the tools
- **Middle button**
  - default: Pan/Move viewport
  - **Shift**: select rectangle region and zoom into
  - **Shift-Ctrl**: select rectangle region and zoom out
  - **Ctrl**: rotate projection using a virtual trackball
  - **Ctrl-Shift**: rotate projection using a virtual trackball with steps of 15
- **Right button (or Ctrl-Spacebar)**: pop-up menu

When laptop mode is enabled in the Preferences then the middle and right buttons are swapped
Tools

- **select h** bodies, regions, or modify viewports
- **pan x** move viewport
- **zoom z** zoom in/out. Clicking will zoom by 2 or draw a rectangle. To zoom-out use Ctrl
- **trackball t** rotate viewport
- **refresh Ctrl-R** refresh all viewports
- **toggle Shift-Z** change view type: bodies, region, material
- **layout v** rotate various layouts
- **errors** show dialog with geometry errors
- **axes o** show dialog to select projection
- **zone Ctrl-Click** show zone description using selected bodies
- **views 1..9** change view projection X-Y, Y-Z,...
Geometry Editor: Viewports

Other viewports are visible with dashed lines.
Manipulating Viewport

- Dashed lines represent viewports
- Center is represented with a square
- When the other-viewport is outside the view window, the viewport-line will be displayed on the closest edge

**Actions** (select-tool + left mouse)

- **drag the center** square to reposition the viewport
- **drag the line close to the center** to reposition the viewport along the vertical axis
- **drag the extremities** of the viewport-line to rotate the viewport
Geometry Errors

- "Errors in Geometry" notifies that are possible errors in the geometry.
- Clicking the icon displays the dialog with the errors.
- Touching surfaces are checked against 10 significant digits
### Geometry Errors

**x, y, z** Coordinates of the error (on the surface of body)

**body** Body with the x, y, z point on surface generating the error

**+body** Regions that are on the + side of the body.

- Regions where the body should be **subtracted** to remove the error

**-body** Regions that are on the – side of the body.

- Regions that the **body** should be **intersected** to remove the error

**+/- of body** are defined according to the normal on the surface.

+ refers to outside, - to inside

---

**Input File Errors**

---

![Geometry Errors](image-url)
Programming Interface: API

There is work presently going on to decouple the functionality from the interface, some of the basic classes can be used to input processing

file: Input.py - to manipulate input files
  import Input
  Input.init([database]) to initialize the database of cards
Most commonly used classes:
  Card containing the description of each card
  Input manipulating the FLUKA input file

file: Project.py - to manipulate project files
API: class Card

Constructor: Input.Card(tag, what [,comment [,extra]])
what is a list starting with what[0]=sdum

Important Methods:
setWhat(n, value) set value to what#n
nwhats() return number of whats
what(n) return value of what#n
numWhat(n) return numeric value of what#n
intWhat(n) return integer value of what#n
clone() return a copy of the card
setEnable(e) enable/disable card
API: class Input

Constructor: Input.Input()
initialize the structure to hold an input file

Important Variables:
cardlist a list with pointers to cards
cards a dictionary with pointers to cards grouped per tag

Important Methods:
read(filename) read input from file
write(filename) write input to filename
addCard(card,pos) add card to position pos (or end of file)
delCard(pos) delete card from position pos
preprocess() preprocess input to check for active cards
setEnable(e) enable/disable card
API: class Project

Constructor: Project.Project()
initialize the structure to hold a project file

Important Methods:
- clear() to re-initialize project
- load(filename) load project from file filename
- save([filename]) save project to filename
- runCmd(run) create run command
API: example

Read an input file and modify the random number seed

```python
import Input
Input.init()
input = Input.Input()
input.read("test.inp")
try:
    rndcard = self.cards["RANDOMIZ"][0]
    rndcard.setWhat(2, 5723)
except:
    print "No RANDOMIZe card found"
    sys.exit(0)
input.write("test2.inp")
```
API: .flair file structure

# comments

Variable: Value

Notes:
  multi-line values are terminated with Ctrl-L

Run: name
  ... Block of Run related information

Data:
  ... Including Data processing information

EndData

EndRun

Plot: name
  ... Plot related informations

EndPlot