Table of Contents

Plot Tool for

The tool is a python script `plots.py` located in `Alignment/Escher/scripts`. The relevant commit is tagged as `nchiapol_20100618`.

The command line to run the tool:

```
python -i plots.py
```

By default the tool uses `CondDB` and `DDDB` with tag `head-20100518` and creates some simple plots for TT. These plots are saved to pdf-Files. The tool provides two command line options. `-t` to change the filetype and `-d` to add database layers. The class definitions for this tool are all in `libPlots.py`. I recommend to have a look at that file.

Types of plots

There are two types of plots

- normal plots (class `Plot`)
- arrow plots (class `ArrowPlot`) displaying an arrow between the nominal position and the aligned position.

Adjusting plots

Plots can be added or adjusted directly in the code in `Alignment/Escher/scripts/plots.py`. The file `examplePlots.py` contains a large number of predefined plots you can copy and paste into `plots.py`. Plots are represented either by instances of the class `Plot` or by instances of the class `ArrowPlot`:

two examples

```
ArrowPlot("VeLo-ZXarr",".*",lambda do:[do.Z,do.Z+10*do.TZ,do.X,do.X+10*do.TX],lambda do:do.Z<500)
Plot("TT-ZtX","/TT(?!T)(?!.*Sensor.*)",lambda do:[do.Z,do.TX])
```

- The first argument is the name and will be the base name of the saved file.
- The second object is a regular expression to select the objects to plot.
- The third argument is a function taking an object of type `ElementGeometry` and returning two respectively four numbers. All members of `ElementGeometry` can be used as return values.

Any function can fulfilling these requirements can be used. For simple things lambda forms (see below) are very useful. For examples look at `examplePlots.py` and the class `PlotFunctions` in `libPlots.py`

- The fourth argument is optional and allows to select spacial regions to plot. Its again a function taking an instance of `ElementGeometry` and returning a bool this time.

When a Plot (or Arrow Plot) object is appended to `plots` the script will assure its correctly filled and the output is saved. Different plots can be combined by using the function `drawCombinedPlot`. An example of how to use it can be found in the code as well.

lambda forms

Lambda forms are a shorthand to create anonymous functions. The following two lines create an identical function `f(x)`

```
>>> # using lambda forms
>>> f = lambda x:x+1

>>> # using normal functions
>>> def f(x): return x+1
```

The python reference for lambda forms can be found at:
http://docs.python.org/reference/expressions.html#lambda

-- AlbertBursche - 27-May-2010 -- NicolaChiapolini - 18-Jun-2010

This topic: LHCb > PlotAlignmentDB
Topic revision: r4 - 2010-06-18 - NicolaChiapolini