

GraphvizGL

A Integration of graphviz library in OpenScientist.

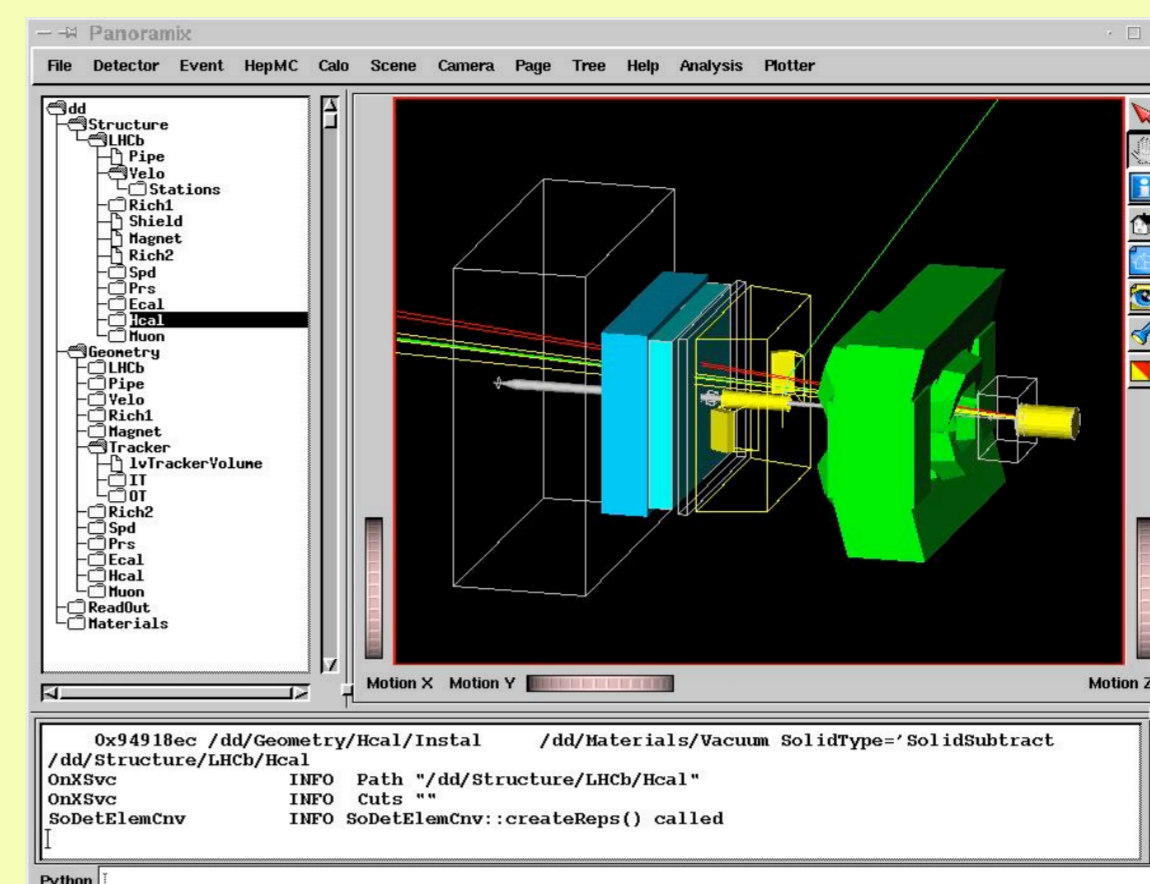
http://www.lal.in2p3.fr/OpenScientist

Laurent Garnier LAL / IN2P3 / CNRS

GOAL : We wanted to connect the visualization part of OpenScientist (OnX) to graphviz by not passing by intermediate image files but by using the native library directly.

OpenScientist : <http://openscientist.lal.in2p3.fr>

- It is an integration of open source products working together to do scientific visualization and data analysis, in particular for high energy physics (HEP).
- Environment of OpenScientist use OpenGL (<http://www.opengl.org>) as a rendering layer, which is now a de facto standard that comes on all interactive platforms.
- Thanks to its modularity, OpenScientist can be connected to several packages. For this project, we will add two new packages to render OpenGL:
 - Graphviz : to render graphs and routes algorithms
 - FTGL : to render freetype fonts



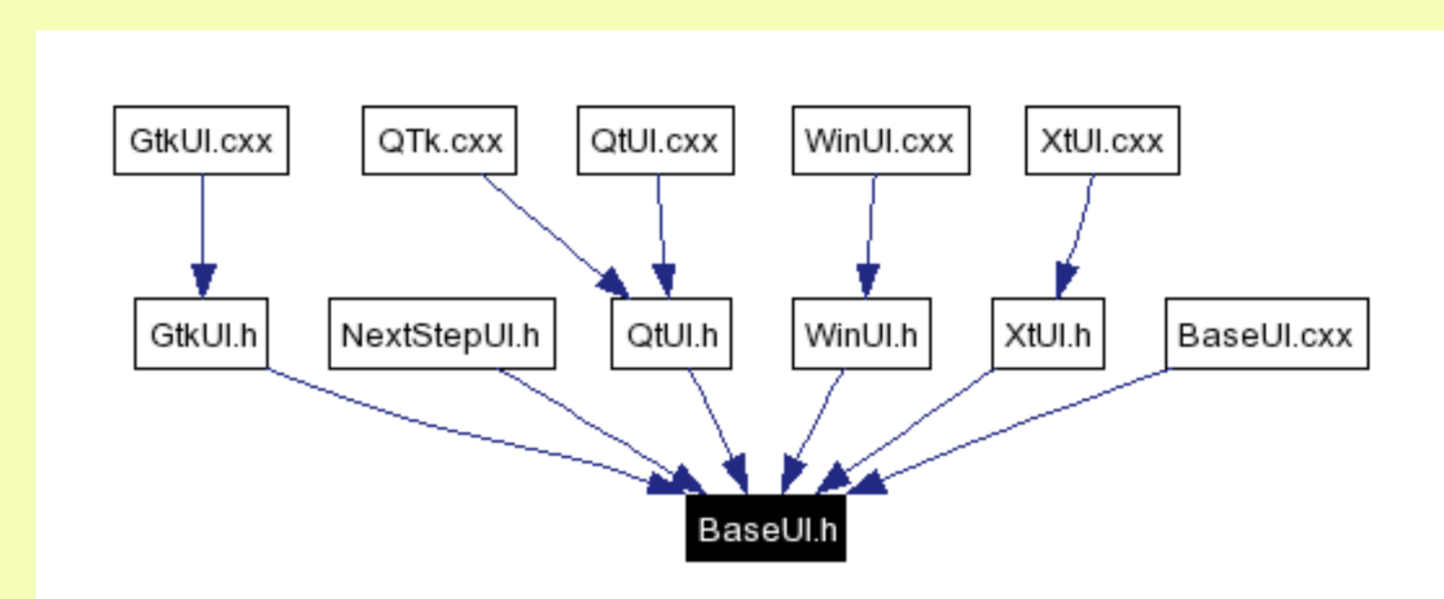
A picture produced with the LHCb Panoramic event display built with OnX

Graphviz Library : <http://www.graphviz.org>

Graph visualization is a way of representing structural information as diagrams of abstract graphs and networks. Automatic graph drawing has many important applications in software engineering, database and web design, networking, and in visual interfaces for many other domains.

- It is open source graph visualization software use for example in doxygen graph generation
- Good at visualizing a scene containing boxes connected by lines
- Containing routing algorithms that permit to connect boxes

For other use, we will have to connect another library with graphviz, in order to render font in OpenGL format : FTGL

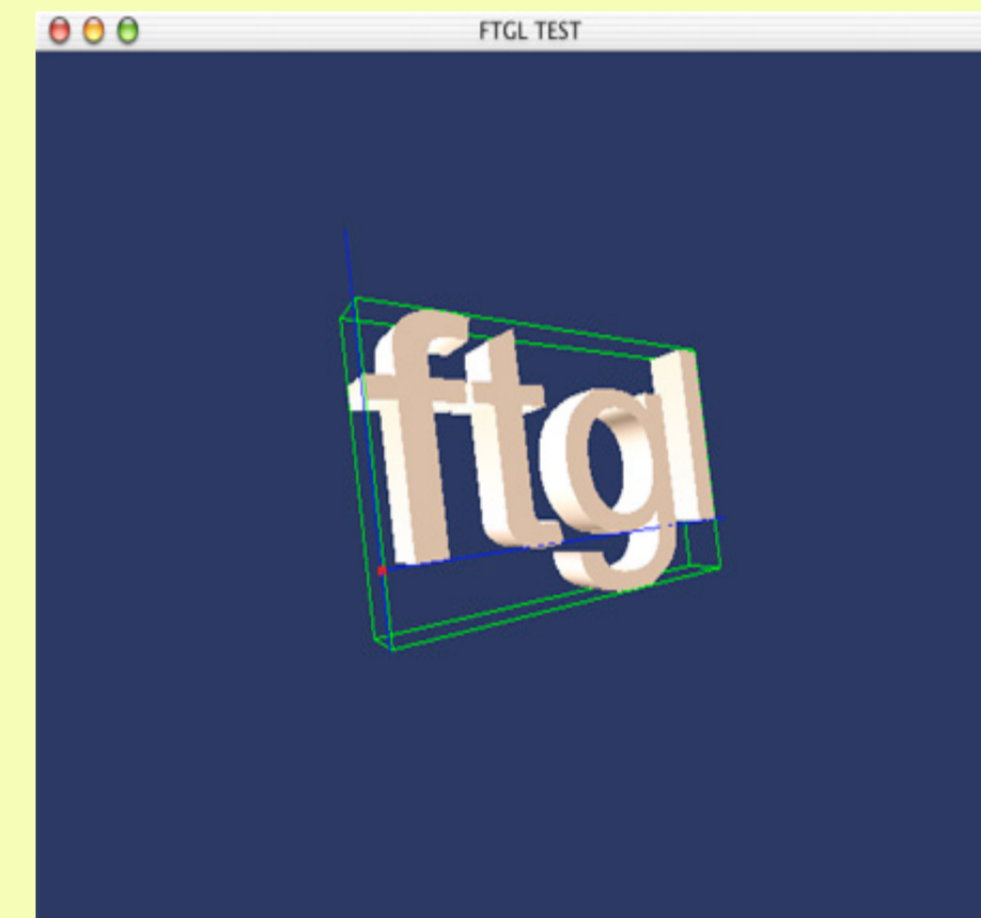


Example of graphviz output on OnX GUI Component

FTGL Library

<http://homepages.paradise.net.nz/henryj/code/#FTGL>

- FTGL is a free, open source library to enable developers to use arbitrary fonts in their OpenGL (www.opengl.org) applications.
- Unlike other OpenGL font libraries FTGL uses standard font file formats so it doesn't need a preprocessing step to convert the high quality font data into a lesser quality, proprietary format.
- FTGL uses the FreeType (<http://freetype.sourceforge.net/>) font library to open and 'decode' the fonts. It then takes that output and stores it in a format most efficient for OpenGL rendering.



The FTGL test example

Implementation details

Graphviz can generate graphs from two ways, with a dot file, or a cxx code. The cxx code below has been used for generation of the OnX result on the right.

.dot file :

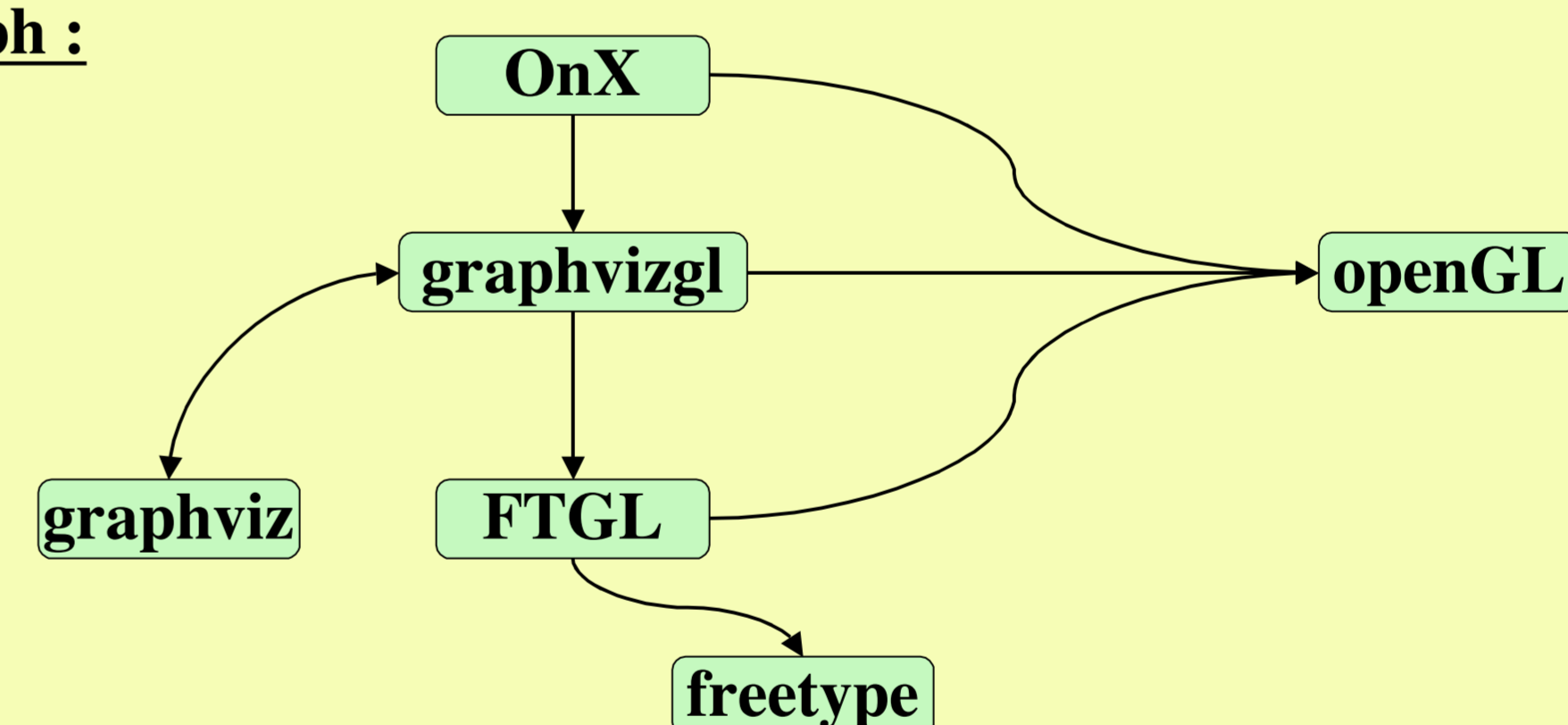
```
digraph g {
  tree [color=red];
  item 1";
  item 2";
  item 3";
  item 4";
  item 5";
  item 6";
  tree -> "item 1";
  tree -> "item 2";
  tree -> "item 3";
  tree -> "item 4";
  "item 1" -> "item 5";
  "item 1" -> "item 6";
  "item 6" -> "item 2";
  "item 6" -> "item 4";
  "item 4" -> "item 5";
  "item 5" -> "item 4";
}
```

cxx code :

```
n = agnode(fGraph, "tree");
n1 = agnode(fGraph, "item 1");
n2 = agnode(fGraph, "item 2");
n3 = agnode(fGraph, "item 3");
n4 = agnode(fGraph, "item 4");
n5 = agnode(fGraph, "item 5");
n6 = agnode(fGraph, "item 6");
e = aledge(fGraph, n, n1);
e1 = aledge(fGraph, n, n2);
e2 = aledge(fGraph, n, n3);
e3 = aledge(fGraph, n, n4);
e4 = aledge(fGraph, n1, n5);
e5 = aledge(fGraph, n1, n6);
e6 = aledge(fGraph, n6, n2);
e7 = aledge(fGraph, n6, n4);
e8 = aledge(fGraph, n4, n5);
e9 = aledge(fGraph, n5, n4);
agsafeset(n, "color", "red", "");
```

It can also export graphs in a lot of format like « jpg », « ps », « svg »...thank to plugins.
As we wanted to use graphviz as a library, dot file is not the good solution.
We have implemented a new plugin which made the link between OnX and graphviz.

Library call graph :



Results of graphviz integration in OnX :

