

## Graphvizgl

L. Garnier

► **To cite this version:**

L. Garnier. Graphvizgl. XV International Conference on Computing in High Energy and Nuclear Physics (CHEP-06), Feb 2006, Mumbai, India. pp.337-338. in2p3-00168140

**HAL Id: in2p3-00168140**

**<http://hal.in2p3.fr/in2p3-00168140>**

Submitted on 27 Aug 2007

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# GRAPHVIZGL

L. Garnier, LAL, Orsay, France

## Abstract

Graphvizgl is a library connecting the well known graphviz library [1] to the OpenScientist[2] software.

## 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

## OPENSIENTIST

It is an integration of open source products working together to do scientific visualization and data analysis, in particular for High Energy Physics (HEP).

The environment of OpenScientist use OpenGL[3] 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 graphvizgl library, we will add two new packages to render OpenGL :

- Graphviz : to render graphs and route algorithms
- FTGL [4]: to render freetype[5] fonts

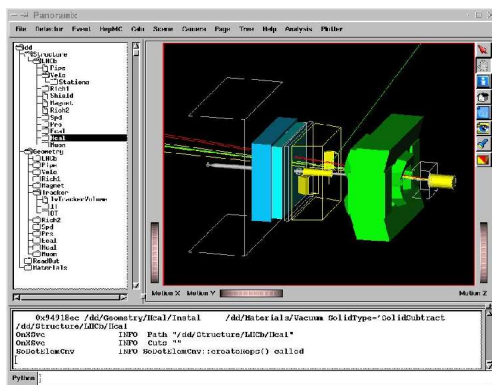


Figure 1 : A picture product with the LHCb Panoramix event display built with OnX

## GRAPHVIZ LIBRARY

Graph visualization is a way of representing structural informations as diagrams of abstract graph 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.

Graphviz is an Open Source graph visualization software use for example in doxygen graph generation. Thanks to multiples routing algorithms, it is very good at visualizing scene containing boxes connected by lines.

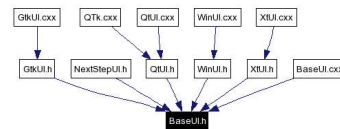


Figure 2 : Example of graphviz output on OnX GUI component.

For ower use, we will have to connect another library with graphviz, in order to render fonts : FTGL

## FTGL

It is a free, open source library to enable developers to use arbitrary fonts in their OpenGL applications.

Unlike other OpenGL font libraries, FTGL uses standart font flie formats so it does not need a preprocessing step to convert the hight quality font data into a lesser quality, proprietary format.

FTGL uses Freetype font library to open and “decode” the fonts. It then takes that output and stores it in a format most efficient for OpenGL rendering.

## IMPLEMENTATION DETAILS

Graphviz library can generated graph from two ways :

- You could give a dot file witch will look like :

```
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";
}
```

- Or directly code your graph description in the .cxx file. The code bellow render the same graph. See

\*garnier@lal.in2p3.fr

figure 3.

```
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", "");
```

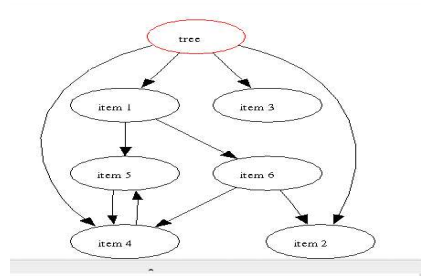


Figure 3 : graph render by code describe previously

## RESULTS

Now, in just a few line of code to call the graphviz library, and a few lines to initialize the graph, we are able to draw in grapps in OpenGL into openScientist.

## REFERENCES

- [1] <http://www.graphviz.org>
- [2] <http://openscientist.lal.in2p3.fr>
- [3] <http://www.opengl.org>
- [4] <http://homepages.paradise.net.nz/henryj/code/#FTGL>
- [5] <http://www.freetype.org>

It can also export graphs in a lot of formats like “jpeg”, “ps”, “vrml”, “svg”, thanks to a plugins mechanism.

As we wanted to use graphviz as a library, dot file is not the good solution. We have implemented a new plugin witch made the link between OnX and graphviz library.