

G4TWIST

The OpenScientist Implementation of Geant 4

Maher Quraan

March 22, 2005

1 Introduction

G4TWIST is an implementation of GEANT 4 in the OpenScientist environment. OpenScientist allows the integration of several packages, and as such takes advantage of many open-source packages, particularly, visualization packages. The combination of these visualization packages with GUIs and scripting allows the creation of an environment that is user friendly and easy to use. This environment is particularly handy when building a detector geometry. In addition to the vivid 3-d images, the user can move, rotate, zoom in and out continuously, all with the click of a mouse. This makes the task of verifying and debugging complicated geometries very straight forward. Scripts can also be used to create buttons that allow the user to display each detector component with the click of a button, thereby, further simplifying the task of geometry verifications.

Although the OpenScientist implementation allows other functionalities, the G4TWIST code is restricted to the geometry part, and in this sense this implementation is only intended for visualization.

2 Directory Organization & Contents

The G4TWIST package resides in cvs under triumph/G4TWIST, and can be checked out using

```
> cvs checkout -d devel triumph/G4TWIST
```

and is meant to reside in the same directory as mofia, geant, and geant4 (i.e. \$HOME/e614soft/triumf/G4TWIST). Several directories appear under G4TWIST; the following is a brief description of the main directories.

G4TWIST/devel/cmt

This is the directory containing the setup and make files. CMT is a Configuration Management Tool that is package oriented. A lot of the setup and configuration has been automated in this package, and it has been used for other HEP experiments such as ATLAS. To use G4TWIST the user needs to execute the setup scripts

```
> cd $HOME/e614soft/triumf/G4TWIST/devel/cmt
```

```
> source /OSC/setup_maher.csh
> source setup.csh
```

and then compile

```
> make
> make Python
```

The make Python command creates the Python libraries that are needed for the Python-scripting part of the code.

The Libraries reside in

G4TWIST/devel/Linux-i686

The source code resides in

G4TWIST/devel/source

and the include files in

G4TWIST/devel/include/G4TWIST.

Note that namespaces are used here, and the namespace for the TWIST code is G4TWIST.

The directory

G4TWIST/devel/scripts

contains three scripting directories

scripts/Geant4

contains G4 macros

scripts/Python

contains Python scripts. The relevant scripts here are the ones that are used to access the geometry for the various detector components for visualization purposes. Changing these scripts or adding more scripts for geometry components that already exist in C++ does not require re-compiling.

scripts/OnX

contains OnX scripts. OnX is a hub for other packages and handles interactiv-

ity scripting and visualization. The directory contain scripts for the GUI buttons that are created for the various geometry components which in turn invoke the corresponding Python scripts in the **scripts/Python** directory.

3 Running G4TWIST

Before running this package make sure to source the environment scripts

```
> cd $HOME/e614soft/triumf/G4TWIST/devel/cmt
> source /OSC/setup_maher.csh
> source setup.csh
```

then go to the run directory

G4TWIST/devel/run

and type

```
> onx
```

This will create a GUI with pull-down menus and buttons. To start the program click on the pull-down menu **Geant4** and click on **Init**. The initialization takes a couple of minutes. Once the initialization is completed you can go to **Geant4** and choose **Geometry**. There you will see many buttons corresponding to the various detector components. Clicking on any of these buttons displays the corresponding component. To clear the page go to **page** and click on **clear page**. One can also create multiple regions using this pull-down menu, etc. Cut views can also be created using the **scene** pull-down menu. various other functionalities exist and are fairly straight forward to use. Moving and zooming the display object can be achieved using the turning wheels on the GUI or using one of the three mouse buttons.