How to use event dispay with BT

This page gives some tips for using FRED with the Beam test software releases, but also with any other GLAST software release.

Disclaimer 1:

This page is not a tutorial about FRED and FRED installation. Detailed information of FRED can be found at :

- http://www.fisica.uniud.it/~glast/FRED/
- http://glast-ground.slac.stanford.edu/workbook/
- http://www-glast.slac.stanford.edu/IntegrationTest/SVAC/Instrument_Analysis/UsefulStuff/eventDisplay.html;

Disclaimer 2:

I am not an expert neither in FRED nor in the different software packages to analyze data. There are people with a lot more expertise than me on these technical issues (Anders, Michael, Leon...). I joined the collaboration just few months ago. Since i am working with both beamtest and full lat data, i spent some time trying to set some simple procedures which allow me to move from one software package to another easily and in a more or less controlled way. This is what is presented in this web page.

The page is devided in 3 sections.

- Setting the environment variables to use a specific software package
- Useful macro to select the events you want to display with FRED
- Setting FRED GLAST panel to visualize the desired data

Important Remark

The scripts and macros use paths which are extracted from the machines at SLAC. In order to work on other cluster of computers one needs to replace the pathnames.

1 - Setting the environment variables to use a specific software package

The software to analyze GLAST data is pretty complex. It involves several (similar) release packages (BeamtestRelease, EngineeringModel, GlastRelease and ScienceTools) and many different release versions. In order to add more complexity to the issue, different release packages run with different root versions, and different programs might be running with different shell flavours. That implies that one needs to be careful when changing software package; the chances of working with a not properly set package (i.e. wrong libraries) are not negligible.

In this section you can find a set of very simple scripts and a even more simple root macro that help setting the environment variables in rather easy and systematic way.

1.1 - Scripts to set some environment variables

The script .SetSoftRelease(.SetSoftRelease_bash for bash or sh shell) sets the environement variables CMTCONFIG (to "rh9_gcc32opt"), GLAST_EXT (to "/afs/slac/g/glast/ground/GLAST_EXT/$CMTCONFIG") and the variable CMTPATH. The first 2 point to the directory where all the software packages hang, therefore they DO NOT need to be replaced. On the other hand, the variable CMTPATH points to the specific software package to be used, and consequently it needs to be modified every time you want to change software package or release. Because of that, the script has 2 arguments:

1. **Software package;** BeamtestRelease, EngineeringModel...
1. **Release version**: *v3r0907p2, v6r070329p17 ...*

An example of how to use it follows:

source .SetSoftRelease *BeamtestRelease v3r0907p2*

I have this script in my home directory, which allows me to call it from everywhere in the following way:

source $HOME/.SetSoftRelease BeamtestRelease v3r0907p2

It is important to note that different software packages might be using different root versions. That is the case (at August 2006) for the BeamtestRelease and the EngineeringModel, which uses root 5 and root 4 respectively. That implies that the root version (and related environment variables) have to be also modified when switching between these two software packages.

The script SetSoftCode_Generic.csh (SetSoftCode_Generic.sh for bash or sh SHELL) can be used to set both, the software package environment variables and the root environment variable ROOTSYS (updating also LD_LIBRARY_PATH and PATH appropriately). An example of how to use it follows:

source SetSoftCode_Generic.csh BeamtestRelease *v3r0907p2*
Here i also provide few scripts which can be used to set some (useful) selected combinations of software package and release version.

- BeamtestRelease-v3r0907p2.csh (BeamtestRelease-v3r0907p2.sh for bash)
- BeamtestRelease-v4r0909p2.csh (BeamtestRelease-v4r0909p2.sh for bash)
- EngineeringModel-v6r070329p16.csh (EngineeringModel-v6r070329p16.sh for bash)

An example of how to use them follows:

source BeamtestRelease-v4r0909p2.csh

Important Remarks

- The above mentioined scripts use the script .SetSoftRelease, and they all assume that this script is located in the $HOME directory. Therefore, if you want to place.SetSoftRelease in any other place, you have to change the scripts accordingly...
- I do have csh and sh scripts because, even though I particularly use csh, FRED wants to use bash (see section 3), and therefore I need both. I do not know whether this applies to you... but that might be useful.

1.2 - Root macro to load root libraries

The macro LoadLibraries.C can be used to load (within a root session) the following root libraries:

- libcommonRootData.so
- libmcRootData.so
- libdigiRootData.so
- libreconRootData.so
- libPhysics.so

The only argument of this macro is a string containing the software package and the release version to be used, in the following way:

"SoftPackage-Version"

An example of how to use it follows:

root [0] .x LoadLibraries.C ("BeamtestRelease-v3r0907p2")

As it will be shown in the next section, one can also include this macro in another macro and execute it from there:

```
#include "LoadLibraries.C"

void sillymacro(TString softrelease = "BeamtestRelease-v3r0907p2")

{

 LoadLibraries(softrelease.Data())

 etc,etc, etc...

}
```

Important Remark

Since the macro is executed within root, it cannot modify the root version accordingly to the software package to be used. Therefore, before executing this macro, check that you are using the appropriate root version for the software package you want to use.

2 - Useful macro to select the events you want to display with FRED

Dipslaying events with FRED is not very quick; especially when you run GLEAM and FRED in a remote machine and you have to "download" the information from the event display to your local machine. Therefore, it is better to select the events you want to display before starting running FRED over them.

In this section you can find macros which can help you selecting the type of events you want to display with FRED.

Let me note that Riccardo Giannitrapani is working in releasing the FRED code so that we can compile it in our local machines. Running FRED in your local machine has the advantage that you only have to download the information of the event to be displayed from the machine where GLEAM runs. That should make things faster. Hopefully the code will be available soon.

The macro doing the main job is EventFilter.C, which was written by *W. Focke H. Kelly and A. W. Borgland*. This macro creates a new digi.root OR recon. root OR mc.root file with a sub-sample of the initial events, which are selected according to a cut in the "merit" or "svac" variables. This macro runs very nicely, but has two small drawbacks:

1. If you want to process the 3 files (digi, recon and mc) which is usually the case, you have to run the macro 3 times
2. You have to be sure you are loading the libraries from the right software package and release version before running it

The macro FilterRootFiles.C is a very simple macro which makes the event filtering a bit easier. With this macro, the user can specify a software package as argument (like *"BeamtestRelease-v3r0907p2" or "EngineeringModel-v6r070329p16*") and the macro loads the appropriate root libraries (using the macro *LoadLibraries.C)* and then executes EventFilter.C for the 3 types of files sequentially(in case they are specified as arguments).

An example of running the macro FilterRootFiles.C is given by macro FilterEvents_BeamTest_CR_CERN322.C, which is the macro that defines the software package to be used, the input and output filenames and the selection cut to be applied to the events. In partituclar this macro uses the software package "BeamtestRelease-v3r0907p2" to select events from beam test run 700000322 according to the selection criteria:

```
char* cuts = "TMath::Abs((acos(Tkr1ZDir)*(180./3.14159))-180.) < 5 && CalEnergyRaw > 10";
```

You can easily modify this macro to fit your needs.

3 - Setting the FRED GLAST panel to visualize the desired data

FRED is installed in the public domain of the noric machines:

/afs/slac.stanford.edu/g/glast/applications/FredRelease/

There are several releases (v0r97p1 v0r98 v0r99).

You can start FRED by executing the ruby program fred.rb from the release you want to use. I am using v0r99, which is the last one installed at SLAC.

This is done by the script

/afs/slac.stanford.edu/g/glast/applications/install/@sys/usr/bin/fred

However, I would suggest to run it with the flag --log DEBUG, so that a logfile ($HOME/fred.log) is created. This log file is very useful when FRED crashes or does not want to run. Besides that logfile, a directory $HOME/.fred is created and contains other logfiles. But those ones did not help me when FRED was not working due to configuration errors...

You can use the simple script myfred (from csh or tcsh) as an example of how to run the fred version you want (in debug mode) with the ruby package you want.

FRED will be acting as a client of Gleam which will be the engine processing the events.
The configuration of the execution of Gleam is performed via the FRED GLAST pannel.

Run FRED by executing the script fred (or myfred) in the noric machines. You will get a FRED display. In order to configure FRED to display the data you want to see you have to open the FRED GLAST pannel, which is the button with a satellite image in the upper part of FRED.

There are 5 fields that need to be filled. They are listed below:

1. **A personal script defining some environment variables related to the software to use (CMTPATH, GLAST_EXT...)**. One can use the scripts which were listed in section 1 of this web page (examples: BeamtestRelease-v3r0907p2.sh, BeamtestRelease-v4r0909p2.sh , EngineeringModel-v6r070329p16.sh)

1. **A "Gaudi" configuration script.** The name of this script is ***setup.sh*** and its location depends on the software package to be used. An example follows: *BeamtestRelease-v3r0907p2:* /nfs/farm/g/glast/u09/builds/rh9_gcc32opt/BeamtestRelease/BeamtestRelease-v3r0907p2/Gleam/v6r18/cmt /setup.sh ; *EngineeringModel-v6r070329p16*: /nfs/farm/g/glast/u09/builds/rh9_gcc32opt/EngineeringModel/EngineeringModel-v6r070329p16 /Gleam/v6r8p1/cmt/setup.sh

1. **A "Gaudi" executable.** This is the Gleam executable, ***Gleam.exe***; and its location depends on the software package that is used. *BeamtestRelease-v3r0907p2:* /nfs/farm/g/glast/u09/builds/rh9_gcc32opt/BeamtestRelease/BeamtestRelease-v3r0907p2/Gleam/v6r18/rh9_gcc32opt/Gleam.exe; *EngineeringModel-v6r070329p16:* /nfs/farm/g/glast/u09/builds/rh9_gcc32opt/EngineeringModel/EngineeringModel-v6r070329p16/Gleam/v6r8p1 /rh9_gcc32opt/Gleam.exe

1. **A job options ascii file.** This file defines, among other things, the input data and geometry to be used. It is slightly different for different software package versions. Examples follow: jobOptions_beamtest.txt, jobOptions_beamtest_montecarlo.txt, jobOptions_EngineeringModel.txt

1. **A working directory.** Any (valid) directory will work

Important Remark

FRED requires to run the script in a BASH shell, therefore the sripts used in the GLAST panel must BASH scripts.

The images BeamTest GLAST panel and EngineeringModel GLAST panel are snapshots of the FRED GLAST panel set to work with beamtest files and engineering model files respectively.

It is worth to say a couple of words about the differences between BeamtestRelease and EngineeringModel job options files. There are two differences.

The first difference is the geometry, which is obviously different. The geometry is set by an xml file, which is especified by the variable **GlastDetSvc.xmlfile**

- BeamtestRelease: *GlastDetSvc.xmlfile = "$(XMLGEODBSROOT)/xml/cu06/cu06SegVols.xml";*
- EngineeringMode: *GlastDetSvc.xmlfile = "$(XMLGEODBSROOT)/xml/latAssembly/latAssemblySegVols.xml";*

The variable *XMLGEODBSROOT* is initialized in when executing the setup.sh script (field 2 in FRED GLAST panel).

The second difference is the definition of the variable **ApplicationMgr.ExtSvc:**

- BeamtestRelease: *ApplicationMgr.ExtSvc* += "EventSelector/EventSelector","EventCnvSvc";
- EngineeringMode: *ApplicationMgr.ExtSvc* += "GlastEventSelector/EventSelector","EventCnvSvc"

Now we are at the point in which we have to push the RUN button. If the files you typed in exist and make sense, then you will be able to start displaying events with FRED.

Important Remark

Occasionally one finds events which cannot be properly displayed and SCREW UP the session. Leon found a problem in AcdReconFiller (2006/07/13) which caused many of those problems (see corresponding discussion in the beam test mailing list). This is fixed in HepRepSvc v0r18p3. Yet it seems that some large events still have problems to be displayed... There are two "dirty" ways of avoiding the problem:

- Not reading the mc.root file; this seems to prevent the problem to happen.
- Skip those "special" events that cause the problem in a particular run. This can be done if the events are selected by typing the event number in the event number display of FRED (upper part of panel) instead of scanning them with the "play" button.

I think that's all what you need to know to inspect the data with FRED... at least that is all I know about displaying data with FRED. Hope that is useful. Do not hesitate in contacting me if there are things which are not clear or you have questions about this stuff (dpaneque@slac.stanford.edu).