Running valgrind on linux

Set up a working area on linux

- I set up an area in my Glast Users area. For example

```
 $USHERNFS/MemLeak
```

  where USHERNFS is /nfs/slac/g/glast/users/glground/usher

- Use this space to store the top level job options file for controlling the job (I still rely on the basicOptions file as much as possible)

Setting up the environment

- Set CMTPATH. Here it is convenient to point to one of the builds already done for us by the Release Manager. You can find the specific paths on the Release Manager page . Pick one of the builds and you will see the path to it on the resulting page. Set the CMTPATH:
  in csh:

```
setenv CMTPATH /nfs/farm/g/glast/u09/builds/rh9_gcc32/GlastRelease/GlastRelease-HEAD1.420
```

  in bash:

```
CMTPATH=/nfs/farm/g/glast/u09/builds/rh9_gcc32/GlastRelease/GlastRelease-HEAD1.420; export CMTPATH
```

- Setup the Gleam environment. Batch jobs inherit the environment of the submitting process. So, "source" the setup file (note that you will need to check the Gleam version number to do this):
  in csh:

```
source $CMTPATH/Gleam/(gleam version)/cmt/setup.csh
```

  in bash:

```
source $\{CMTPATH}/Gleam/(gleam version)/cmt/setup.sh
```

- Setup the top level job options environment variable to point to your top level job options file. For example:
  in csh:

```
setenv JOBOPTIONS $USHERNFS/RealData/readdigi_runrecon.txt
```

  in bash:

```
JOBOPTIONS=$\{USHERNFS}/RealData/readdigi_runrecon.txt; export JOBOPTIONS;
```

- Make sure you can run Gleam:

```
$CMTPATH/Gleam/(gleam version)/rh9_gcc32/Gleam.exe
```

Looking for memory leaks

- To run valgrind

```
valgrind -v --leak-check=yes --show-reachable=yes --logfile=log $CMTPATH/Gleam/(gleam version)/rh9_gcc32
/Gleam.exe
```

References

- Valgrind web page is here

TkrGroup