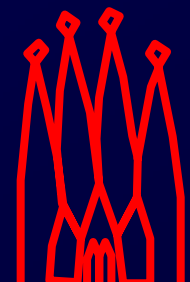
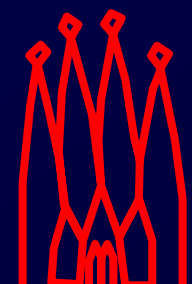
5

Histograms And N-tuples



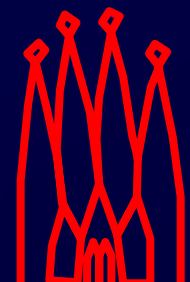
Histograms & N-tuples

- One of the key tools in HEP
- HBOOK was one of the best packages in CERNLIB
- Usage and function is obvious
  - We did not reinvent the wheel
- In Gaudi it's the same concept
  - First book then fill, requires explicit use of histogram pointer (c.f. HFF1)
  - Simplification in GaudiHistoAlg, combine in a single call, and hide pointer handling in base class



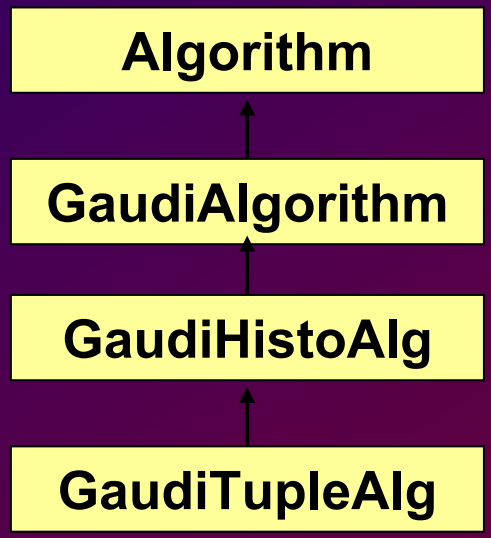
Histograms - Good To Know...

- Histograms are kept in memory
- If not saved - they are lost
- Like all other data -
  they reside in a Data Store
  - Same access mechanism
- Persistency is configuarable
  - HBOOK, ROOT

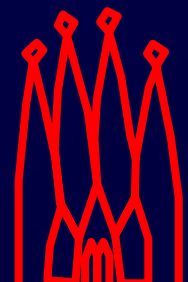


GaudiHistoAlg and GaudiTupleAlg

- Specialisations of GaudiAlgorithm



- Simplify handling of histograms and N-tuples



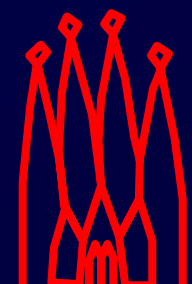
Booking and filling 1D histograms

- **Booking done automatically on first filling call**

```
plot( energy,
      12,
      "Primary particle energy (GeV)",
      0.,
      100.,
      100 );
```

Variable to plot
Integer ID
Title
Low edge
High edge
Number of bins

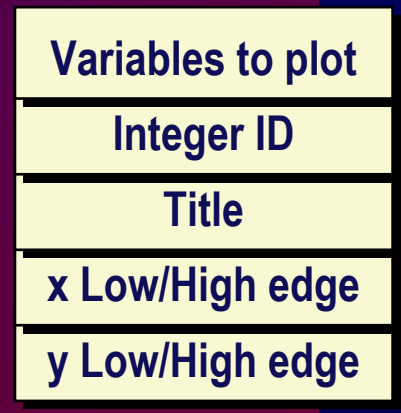
- **Location is set by job options**
  - HistoTopDir
    (default is "", recommend using "sub-detector name" + "/")
  - HistoDir (default is algorithm name)



Booking and filling 2D histograms

- similar to 1D

```
plot2D( xVtx, yVtx
        1001,
        "Primary vertex position",
        -1., 1.,
        -1., 1. );
```



- In this example, number of bins and weight are taken from defaults
  - see doxygen
  - 3D histograms, 1D profiles and 2D profiles are also available

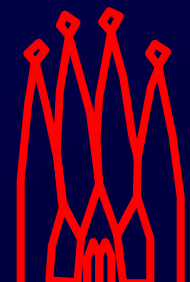
Histogram Persistency

Job options

use HbookCnv v*

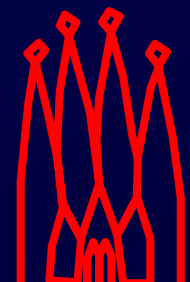
```
// Set up the service using standard options
// Choose Hbook or Root
#include "$STDOPTS/RootHist.opts"
//#include "$STDOPTS/Hbook.opts";

// Output filename (use .hbook extension for HBOOK)
HistogramPersistencySvc.OutputFile = "histo.root";
```



N-tuples - Good To Know...

- Cannot be kept in memory
  - Grow and grow and grow...
- Like all other data - reside in a Data Store
  - Same access mechanism
  - Usage simplified by GaudiTupleAlg

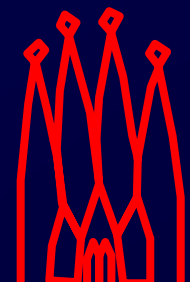


Book and fill an N-tuple

- **Book, declare items and fill all in one go:**

```
// Book the N-tuple. If already booked, retrieve its pointer
Tuple myTuple = nTuple( 100, "An example nTuple" );

// Declare the columns and fill
myTuple->column( "Ntrack",   numTracks );
myTuple->column( "NeutralE", neutralEnergy/TeV );
// Commit the entry
myTuple->write();
```



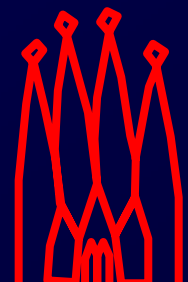
N-tuple Persistency

- **Job options**

```
NTupleSvc.Output = { "FILE1
                     DATAFILE='../job/tuples.root'
                     OPT='NEW'"};
Myalg.NupleLUN = "FILE1" ;
// Convention for file extension is .root or .hbook
```

- **Default persistency is Root. For Hbook:**

```
ApplicationMgr.HistogramPersistency = "HBOOK";
ApplicationMgr.DLLs += { "HbookCnv" };
```



Histogram and Ntuple IDs

Numerical and Alpha Numerical IDs possible

E.g.

```
plot1D( energy, 12, "Particle Energy", 0,100,100 );
```

"HistoTopDir/HistoDir/12"

```
plot1D( energy, "en1", "Particle Energy", 0,100,100 );
```

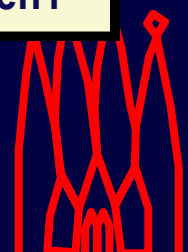
"HistoTopDir/HistoDir/en1"

```
plot1D( energy, "subdir/12", "Particle Energy", 0,
  100, 100 );
```

"HistoTopDir/HistoDir/subdir/12"

```
plot1D( energy, "subdir/en1", "Particle Energy", 0,
  100, 100 );
```

"HistoTopDir/HistoDir/subdir/en1"



Extensive Examples

use GaudiExamples v*

$GAUDIEXAMPLESROOT/

- src/Histograms/GaudiHistoAlgorithm.{h,cpp}
- src/TupleEx/TupleAlg.cpp
- src/TupleEx/TupleAlg2.cpp
- options/Histograms.opts
- options/TupleEx.opts