



HEPAnalLib for DØ data analysis



Jeremiah Mans
Princeton University

■ Overview

- Design of HEPAnallib
 - Philosophy
 - Major groups of classes
 - Analysis structure
- Interface with DØ data (TMBTree)
 - Using DØ Code
 - Culled files

■ Philosophy of HEPAnalLib

- Fast analysis – fast cycling over MC, data
- Delegate to other packages as much as possible
 - Lorentz vectors, matrices, and random numbers from CLHEP
- Allow multiple implementations of I/O
 - HBOOK and ROOT histograms
 - HBOOK-Ntuple, ROOT Tree, and other sources of events

■ Classes provided by HEPAnalLib

- Generic Histogramming package
 - Storage in ROOT or HBOOK files
- Utility classes
 - Constrained fitter – (originally from Scott Snyder)
 - Kernel-estimation smoothing (KEYS)
 - Normalization counters

Analysis Structure

- An analysis package consists of several classes:
 - A main analysis class, derived from HEPD0Anal
 - An event class, derived from HEPD0Event
 - If desired, a set of helper classes which encapsulate preselection and selection stages.
- Events characterized by an integer EventMask
 - Used to build sets of histograms
 - Counting for normalization
 - Stored when culling TMBTrees

■ Accessing the Data

- HEPAnalLib has been interfaced with the TMBTree.
 - Uses the general released TMBTree classes, and adds a few simple classes to interface the files into HEPAnalLib
- Code which has been written to run on just the TMBTree (requiring no core DØ software) can run in HEPAnalLib
 - EMCandidate
 - MuCandidate
 - lm_access

■ Culled files

- Significant increases in analysis speed can be achieved by applying a simple broad preselection to the data and culling out only those events which pass to another file.
- Direct cloning of the TMBTrees fails because of TRefs
- We have written code to fix the TRefs during copying
- Potential analysis speedup is large (reduce time by more than 3x)



■ More information

- More information is available at <http://flywheel.princeton.edu/HEPAnalLib>