D0 Monte Carlo Event Selection System
An Intro to the d0_mess package

Dave Evans

14/3/02

What is d0_mess?

- Monte Carlo Event Selector driven by an rcp
- Allows user to choose events containing particles within specifed ranges of a parameter (Eg pT)
- Runs in Pythia or as a standalone executable
- New Framework package

- Web page:
  - http://www−clued0.fnal.gov/d0_mess/
- Describes:
  - getting and setting up the package
  - building an excutable
  - running the executable
  - Controlling the executable

- Makes use of the d0cuts and cuts packages by Harry Melanson
- Not yet in release so may need to add the latest versions of these when you build the executable until it gets released. (~1 weeks time)

- Allows the user to specify cuts on:
  - PdgId
  - Pt
  - E
  - Eta
  - AbsEta
  - Phi
  - Parent Id
  - Hadron Flavor
- And logical combinations of these

- Get and build the package as per the setup instructions on the web page.
- Edit the rcp file.
  - d0_mess.rcp turns on selection and defines the cuts
    - bool d0_mess_on = true
    - string Cut1 = "PdgId == 13 && Pt >= 5.0"
  - You can specify as many cuts as you want
  - You can require multiple particles in an event pass a cut

- The rcp allows you to define as many cuts as you require, with whatever values you like using the NumberOfCuts switch
- int NumberOfCuts = 2 // means 2 cuts
  - means you specify Cut1 and Cut2
- int NumberOfCuts = 4 // means 4 cuts
  - means you specify Cut1,Cut2,Cut3 and Cut4

Repeated Cuts

- Default behaviour is that there is at least one particle in an event that passes a cut.
- This can be set to any integer value for any given cut. For Example:
  - string CutX = "PdgId == 13"
  - int RepeatCutX = 3
- Means that there must be at least 3 particles passing CutX in an event to keep it.

- To Specify a value range:
- Eg Muon in Pt range 10.0 − 50.0 GeV
  - string Cut1 = "PdgId == 13 && Pt >=10.0 && Pt<=50.0"
- Warning:
  - 10.0 <= Pt <= 50.0 will not work, you must and two seperate cuts

Logical Combinations

- Cuts can be defined using logical AND, OR and NOT operators:
  - && AND
  - || OR
  - ! NOT
  - ==, >=, >, <, <=, != Equality Operators
- Basic Cuts are defined as:
  - <Parameter> <Equality Operator> <Value>
- They can then be related using the logic operators

How Cuts Work

- Each Cut is run on every individual particle in the event. Those that pass are counted.
- A Particle can only have 1 Id, so you can only specify OR relationships for Ids:
  - PdgId == 13 || PdgId == −13 will work
  - PdgId == 13 && PdgId == −13 will not work
  - This is because a muon cant also be an antimuon (unless you are a weird theorist...)

- Muon or Antimuon with Pt > 10.0 GeV in central Eta region:
  - "AbsPdgId==13 && Pt >10.0"
- Electron within Pt range from a B0 parent
  - "PdgId == 11 && Pt>=2.0 && Pt <= 50.0 && ParentId == 521"
- Event with a b flavored hadron:
  - "FlavorId == 5"

Relating Cuts

- It is not yet possible to relate cuts to each other although hopefully it can be implemented soon to allow statements like (Cut1 && Cut2) || Cut3
- This means certain selections cant be done yet that require related cuts.
- Eg: (Muon with Pt >5 and Antimuon with Pt>3) or (Antimuon with Pt >5 and Muon with Pt>3)

- Implement Weight controls
- Set up to write Selection Ntuple
- Expand d0cuts/cuts packages to allow for sets of cuts (Eg Daugther Lists etc)
- Find someone to take it over....