TTree::Draw

What is it trying to be?

Where is it going?

What is it?

- Quick way of creating Histograms or TEventList from a TTree.
- Implement one or 2 simple straight-forward loops
  - Outer loop:
    - For each entries
  - Inner loop:
    - For each elements of the ‘implied’ 1D array

What is the ‘implied’ 1D array

- For each variables, all array dimensions are calculated

  With int fArr[6][4][7]

  “fArr[][3][]” has 2 dimensions: 6 and 7

- For each dimension calculates the minimum size

  With fFloat[4][9]

  “fArr[][3][]+fFloat[][]” has 2 dimensions: 4 and 7

- Calculates fArr[i][3][j]+fFloat[i][j] for (i,j) in:
  - (0,0),(0,1)...(0,6),(1,0)...(1,6)...(3,0)...(3,6)
  - Iteration$ with return the ‘index’ in the above mention list (I.e. 6*i+j)
  - Length$ will return the number of element in the list (I.e 4*7=28)

Other features

- The index of an array can be a “formula”
- If the classes of the objects in the tree is available, on can call their members functions

  `fH.GetXaxis().GetXmax()`
- If the objects are of different types, one can cast to a specific type. The objects which are NOT of that type (or an inherited type) are skipped:

  With `TShape *fShape`:

  `“((TBRIK*)fShape)->GetVisibility()>0”`

New: Custom Histogram Size

- Abilities to customize bin numbers (in rootrc and function call)
  - In rootrc file:
    - Hist.Binning.?D.[x,y,z,Prof,Profx,Profy]
  - In TTree::Draw
    - tree.Draw("sqrt(x)>>hsqrt(500,10,20)“
    - 1 - bins in x-direction
    - 2 - lower limit in x-direction
    - 3 - upper limit in x-direction
    - 4-6 same for y-direction
    - 7-9 same for z-direction

New: Specials Variables

- Entry$
  - Return the current entry number; TTree::GetReadEntry()
- Entries$
  - Return the total number of entries; TTree::GetEntries()
- Length$
  - Return the total number of elements of this formula for this entry (like TTreeFormula::GetNdata())
- Iteration$
  - Return the current iteration over this formula for this entry (i.e. Varies from 0 to length$).

Other New Features

- Array of objects
- Fixed (added?) Support for 1D and 2D graphical cuts
- TEventList now carries not only the list of entries but also the internal conditions.
  - I.e filters both the entry and the array elements inside.

Other New “Features”

- Proper treatments of strings and char arrays
- String inside arrays
- Variable length array inside a non-split TClonesArray
- Variable size arrays coordination
  - (Between main formula, selection and index formula)

Outstanding Bugs

- Check for pointers that are 0 in the method calling chain
- Handling of variable size array that happens to be of size 1
- A few odd problem with strings

Question

- fTracks a TClonesArray of TTrack
- fPoints an array of double inside TTrack
- What should the following do?

```
tree->Draw("fTracks.fPoints[][]-fTracks.fPoints[][fTracks.fAvgPoints]");
```

- Either

```
fTracks[ I ].fPoints[ J ] - fTracks[ I ].fPoints[fTracks[ I ].fAvgPoints]
```

- Or

```
fTracks[ I ].fPoints[ J ] - fTracks[ I ].fPoints[fTracks[ J ].fAvgPoints]
```

Future Plan

1. Calling CINT function (or macro file)
2. Allow arguments if function calls

In particular, using a notation to be defined, the CINT function will have direct access to the entry data.

Future Plan

3. Allow following (transparently) TRef and TRefArray
4. Implement Index$(variable,dimension)
5. Add an interface to pass the histogram by address rather than name