

# Histogram handling in CxAODReader

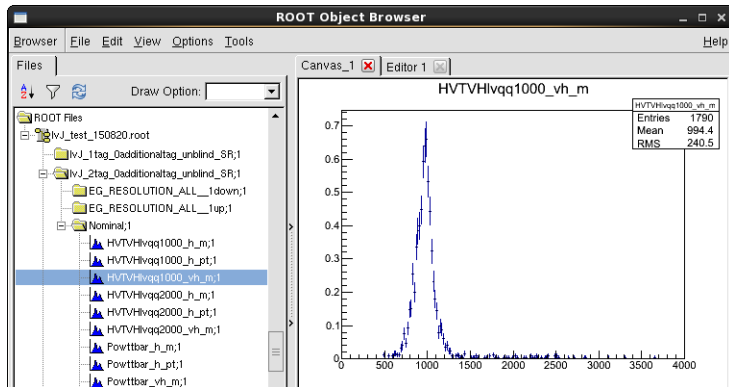
Daniel Büscher

Universität Freiburg

September 18, 2015

# Introduction

- Discussing histogram handling in CxAODReader using the example of the VHreso code
  - Naming convention: “channel\_tags\_description/variation/sample\_distribution”
  - Showing example code from SVN (links below), but simplified in many places
- ⇒ The SVN code has more checks and optimizations, which should be kept



<https://svnweb.cern.ch/trac/>

[atlasoff/browser/PhysicsAnalysis/HiggsPhys/Run2/Hbb/CxAODFramework/CxAODReader/](https://svnweb.cern.ch/trac/atlasoff/browser/PhysicsAnalysis/HiggsPhys/Run2/Hbb/CxAODFramework/CxAODReader/)

<https://svnweb.cern.ch/trac/>

[atlasphys-exa/browser/Physics/Exotic/Analysis/DibosonResonance/Data2015/Code/CxAODFramework\\_DB/CxAODReader\\_DB/](https://svnweb.cern.ch/trac/atlasphys-exa/browser/Physics/Exotic/Analysis/DibosonResonance/Data2015/Code/CxAODFramework_DB/CxAODReader_DB/)

# HistSvc and HistNameSvc initialization

- HistSvc: generic histogram handling
  - HistNameSvc: analysis specific histogram naming
- ⇒ The VHreso code uses a custom HistNameSvc\_VHreso
- Both are initialized in AnalysisReader\_DB::histInitialize()
  - The HistSvc object gets a pointer to the HistNameSvc object

```
1  EL::StatusCode AnalysisReader_DB :: histInitialize () {  
2      // ... do some stuff ...  
3      m_histSvc = new HistSvc();  
4      if (m_analysisType == VH1LepHist) {  
5          m_histNameSvc = new HistNameSvc_VHreso();  
6      } else {  
7          m_histNameSvc = new HistNameSvc();  
8      }  
9      m_histSvc -> SetNameSvc(m_histNameSvc);  
10 }
```

# Histogram booking and filling

- HistSvc provides 1-line booking and filling of histograms (lines 11-12)
  - Only the distribution name is provided when filling
  - The full histogram name is determined by HistNameSvc (info given beforehand)
- ⇒ E.g. number of tags (nTag, nAddTag) are set before filling (lines 8-9)
- Note that such information can be provided to HistNameSvc anywhere in the reader
- ⇒ E.g. in a separate b-tagging method

```
1  EL::StatusCode AnalysisReader_DB :: fill_VH1LepHist() {
2    // ... get containers from selection result ...
3    int nTag = 0;
4    for(const xAOD::Jet* trackJet : matchedJets){
5        if (Props::MV2c20.get(trackJet) > bTagCut) nTag++;
6    }
7    // ... calculate nAddTag ...
8    ((HistNameSvc_VHreso*) m_histNameSvc) -> set_nTag(nTag);
9    ((HistNameSvc_VHreso*) m_histNameSvc) -> set_nAddTag(nAddTag);
10   // ... calculate four vectors ...
11   m_histSvc -> BookFillHist("vh_m", 200, 0, 4000, VHVec.M()/1e3, m_weight);
12   m_histSvc -> BookFillHist("h_pt", 200, 0, 4000, HVec.Pt()/1e3, m_weight);
13 }
```

# Inner workings of HistSvc

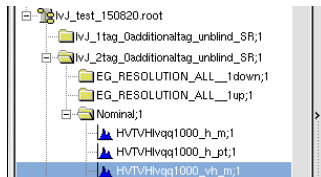
- BookFillHist is basically one line: call BookHist and fill the returned histogram
- BookHist retrieves the full histogram name from HistNameSvc (line 7)
- Then it tries to find the requested histogram in a map and return it
- If the histogram is not found (the first time it is requested) a new one is created

```
1 void HistSvc::BookFillHist(const string& name, /* ... */) {
2     BookHist(name, nbinsx, xlow, xup) -> Fill(value, weight);
3     // ... additionally fill weight systs ...
4 }
5
6 TH1* HistSvc::BookHist(const string& name, /* ... */) {
7     string fullname = m_nameSvc -> getFullHistName(name);
8     TH1* hist = FindHistInMap(fullname);
9     if(!hist) {
10         hist = new TH1F(fullname, fullname, nbinsx, xlow, xup);
11         m_hists[fullname] = hist;
12     }
13     return hist;
14 }
```

# Implementation of custom HistNameSvc

- The implementation of HistNameSvc\_VHreso determines the full histogram name
- ⇒ Slashes are interpreted by EventLoop/SampleHandler to create folder structure
- The name is built in getFullHistName() using given information
- ⇒ E.g. “\_2tag” is added if HistNameSvc::set\_nTag(2) was called beforehand (line 6)
- Note that getFullHistName() is called many times ⇒ needs efficient implementation

```
1  std::string HistNameSvc_VHreso::getFullHistName(const std::string& variable) {
2      m_name = "lvJ";
3      if (m_nTag < 0) m_name += "_0ptag";
4      else if (m_nTag == 0) m_name += "_0tag";
5      else if (m_nTag == 1) m_name += "_1tag";
6      else if (m_nTag == 2) m_name += "_2tag";
7      else m_name += "_3ptag";
8      // set nAddTag and description string ...
9      m_name += "/" + m_variation + "/";
10     m_name += m_sample + "_" + variable;
11     return m_name;
12 }
```



# Summary

## Summary

- HistSvc provides a convenient way for histogram booking and filling (1-liner)
- HistNameSvc separates the histogram naming from the reader, can easily be customized

## Not discussed

- Efficient filling of weight systematics with HistSvc, some info at:  
<https://its.cern.ch/jira/browse/CXAOD-24>
- Filling the same distribution multiple times in one event (e.g. for different regions)

⇒ Can mostly be avoided by a good definition of orthogonal regions

⇒ If needed one can do a simple loop over BookFillHist, example at:

[https://svnweb.cern.ch/trac/atlasphys-exa/browser/Physics/Exotic/Analysis/DibosonResonance/Data2015/Code/CxAODFramework\\_DB/CxAODReader\\_DB/trunk/Root/AnalysisReader\\_DB.cxx?rev=240346#L927](https://svnweb.cern.ch/trac/atlasphys-exa/browser/Physics/Exotic/Analysis/DibosonResonance/Data2015/Code/CxAODFramework_DB/CxAODReader_DB/trunk/Root/AnalysisReader_DB.cxx?rev=240346#L927)