



xAOD Object

(xAOD::EventInfo, xAOD::Muon, ...)

1. Something that inherits from SG::AuxElement
 - An AuxElement holds a pointer to an AuxStore
2. Without the AuxStore, the AuxElement has no information
3. It has functions/methods for manipulating the information that will come from the AuxStore:
 - e.g. a px() method for x-momentum, where the auxstore holds pt,eta,phi

xAOD Container

(xAOD::MuonContainer, ...)

1. A Collection of xAOD objects
2. Almost always simply a DataVector (think of as vector of pointers) of the xAOD objects
3. Container can be assigned an AuxStore, which holds vectors of numbers/information

AuxStore

(xAOD::EventAuxInfo, xAOD::MuonsAuxContainer, ...)

1. Usually inherit from xAOD::AuxInfoBase (holds numbers etc.) or xAOD::AuxContainerBase (holds vectors of numbers etc.)
 - An AuxContainerBase (or any derived class) is suitable for holding information for a vector of xAOD objects (e.g. the xAOD::MuonContainer: a DataVector of xAOD::Muons)
 - An AuxInfoBase (or any derived class) is suitable for things like xAOD::EventInfo
2. When written to a ROOT file, they appear in a TTree as simple branches (floats, ints, etc) or vector branches (vector<float>, etc)
 - Therefore you can browse/inspect this information using just ROOT
 - The actual xAOD Objects,Containers,AuxContainers are **not ROOT readable: you need the libraries**
3. A piece of information in the aux store is either **static** or **dynamic**
 - Static aux information is essential information predeclared at compile time. E.g. a xAOD::MuonsAuxContainer **must** have a vector<float> for the pt, eta, phi values of the muons
 - Dynamic aux information is any other information; AuxInfoBase and AuxContainerBase can hold only dynamic information