In EL::StatusCode AnalysisReaderZV2Lep :: fillEasyTree(const ResultsVV2Lep& sel, float scaleFactor)

```
//Always fill in these branches at least
if(mTreeStrategy>0){
    mETree->SetBranchAndValue<bool> ("isLep1MuonFlavour", (sel.mLep1.Flavor() != LepFlavor::EL), -99.);
    mETree->SetBranchAndValue<bool> ("isLep2MuonFlavour", (sel.mLep2.Flavor() != LepFlavor::EL), -99.);
    mETree->SetBranchAndValue<float> ("X_resolved_ZZ_m", X_resolved_ZZ.M()*1e-3, -99.);
    mETree->SetBranchAndValue<float> ("X_resolved_WZ_m", X_resolved_WZ.M()*1e-3, -99.);
    mETree->SetBranchAndValue<float> ("X_boosted_m", X_boosted.M()*1e-3, -99.);
    mETree->SetBranchAndValue<float> ("X_boosted_m", X_boosted.M()*1e-3, -99.);
    for (std::string const & cut : VV2LepCuts_ResolvedFinalRegion)
        easySetAndFillCut(sel, cut);
    for (std::string const & cut : VV2LepCuts_MergedFinalRegion)
        easySetAndFillCut(sel, cut);
}
```

1

WARNING: Can't find cut DFLeptons EJSEJSEJSEJSEJSEJSEJSEJS EL::StatusCode AnalysisReaderZV2Lep::mergedAnalysis(ResultsVV2Lep&) EJSEJSEJSEJSEJSEJSEJSEJSJES EL::StatusCode AnalysisReaderZV2Lep::resolvedAnalysis(ResultsVV2Lep&, std:: cxx11::string) EJSEJSEJSEJSEJSEJSJSEJSJES EL::StatusCode AnalysisReaderZV2Lep::fillEasyTree(const ResultsVV2Lep&, float) EJSEJSEJSEJSEJSEJSEJSJSEJS EL::StatusCode AnalysisReaderZV2Lep::fillCutFlowTables(const ResultsVV2Lep&) WARNING: Can't find cut DFLeptons EJSEJSEJSEJSEJSEJSEJSJES EL::StatusCode AnalysisReaderZV2Lep::resolvedAnalysis(ResultsVV2Lep&, std:: cxx11::string) EJSEJSEJSEJSEJSEJSEJSJES EL::StatusCode AnalysisReaderZV2Lep::eventCategorization(ResultsVV2Lep&) WARNING: Can't find cut DFLeptons

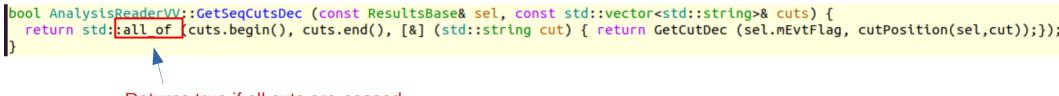
```
bool mFlushTree=false;
if(mTreeStrategy==1 || mTreeStrategy==2)
mFlushTree=inFinalRegion;
else
mFlushTree= GetSeqCutsDec(sel,{"Trigger", "LepPt", "MuonEtaLt2p5", "OSMuons", "Mll", "SFLeptons", "SigJetORFatJet"});
```

if (mFlushTree) EL_CHECK("AnalysisReaderZV2Lep::Run1LepAnalysis", EasyTreeFiller<ResultsVV2Lep>(sel,std::bind(&AnalysisReaderZV2Lep::fillEasyTree, this,_1,_2)));

In our case, treeStrategy is 3

```
# Output Control
#####################################
# If want histogram output
bool writeHistograms
                            = true
# If want tree output
bool writeEasvTree
                            = true
# If true, EasyTree will only be written for Nominal case.
bool nominalTreeOnly
                            = false
# If true, EasyTree will only have subset of variables needed for fit studies
int doTreeStrategy
                            = 3 #0=none, 1=RFonly, 2=more, 3=all
# If true, will only save histograms for the invariant mass distribution: lvjj and lvJ
int doHistStrategy
                            = 2 #0=none, 1=MVVonly, 2=more, 3=all
```

This means that in the if-else structure above, the GetSeqCutsDec(...) function is always called



Returns true if all cuts are passed

if(mFlushTree) EL_CHECK("AnalysisReaderZV2Lep::Run1LepAnalysis",EasyTreeFiller<ResultsVV2Lep>(sel,std::bind(&AnalysisReaderZV2Lep::fillEasyTree,this,_1,_2)));

Rerun the analysis of mc16a Zjets without that if The file size increased, not uniformly (some file grew a bit, some grew 3 times, ecc)