



# STAR Software Basics

---

- Compiling using 'cons'
- STAR\_LEVELS, starver and changing environment
- Coding Standards for C++
- File types in use

<http://www.star.bnl.gov/STAR/comp/>



# STAR Software Library

## Is \$STAR

BIN OBJ StRoot asps include lib obj

LIB StDb StarDb bin kumacs mgr pams

## Makers

– Compiled code to put in a “chain”,

2 main chains are `bfc.C` and `doEvents.C`

`bfc.C` is the chain for running reconstruction. Works on **raw** data.

`doEvents.C` is for analyzing the **reconstructed** dsts

## Macros

– Interpreted code - many examples

**\$STAR/StRoot**



# CVS - Concurrent Versions System

---

- Used to record the history of your source files.
- CVS is essential for a project is being worked on by multiple people.

## Quick Reference

***cv* *co name*** = check out a copy of ' name' into your working directory

***cv* *update*** = update all the files in the current directory

***cv* *-n update*** = tells what it would update, but the -n means it doesn't do it



# Cons

---

Replacement for “Make”

Cons is a system for constructing software

Cons ensures that complex software is easily and accurately reproducible

**To get help: *cons -x***

**Full build:** In top level directory type

***cons***

**Partial build:**

***cons +name***

**Remove products:**

***cons -r +name***



# Cons – example:

---

Obtain token

***klog*** (then type afs password at prompt)

In a fresh directory “check out”

***cvs co StRoot/StFtpcTrackMaker***

Compile and build using “cons”

***cons +StFtpcTrackMaker***



# STAR\_LEVEL

[http://www.star.bnl.gov/STAR/comp/ofl/software\\_releases.html](http://www.star.bnl.gov/STAR/comp/ofl/software_releases.html)

---

## The present release assignment:

**SL00m** (SL00m) ROOT\_LEVEL 2.25.03  
**SL01i** (SL01i) ROOT\_LEVEL 3.02.00 AuAu 130GeV production  
**SL01j** (SL01j) ROOT\_LEVEL 3.02.00 updated SL01i with SIMU tag  
**SL02c** (SL02c) ROOT\_LEVEL 3.02.07 AuAu200 GeV production  
**SL02d** (SL02d) ROOT\_LEVEL 3.02.07 AuAu200 GeV production  
**SL02e** (SL02e) ROOT\_LEVEL 3.02.07 AuAu and pp 200GeV production  
**SL02g** (SL02g) ROOT\_LEVEL 3.02.07 FTPC side AuAu200 production  
**SL02h** (SL02h) ROOT\_LEVEL 3.02.07 only tag exists  
**SL02i** (SL02i) ROOT\_LEVEL 3.02.07 daq100 test,dAu200 MC  
**SL03a** (SL03a) ROOT\_LEVEL 3.03.09 real dAu 200GeV production  
**SL03b** (SL03b) ROOT\_LEVEL 3.03.09 dau200 MC hijing  
**old->** SL03d (SL03d) ROOT\_LEVEL 3.03.09  
**pro->** SL03e (SL03e) ROOT\_LEVEL 3.05.04 ppMinBias, ppTrans-1,ppLong-1  
**new->** SL03f (SL03f) ROOT\_LEVEL 3.05.04 ppMinBias 2001/2002 SVT rerun  
**dev->** DEV ROOT\_LEVEL 3.10.01  
**.dev->** .DEV ROOT\_LEVEL 3.99.99

---



# Changing STAR\_LEVEL

- 3 main library versions:
  - **dev** (updated daily) for development
  - **new** (~few weeks) relatively stable, perform studies here
  - **pro** (when necessary) for official production
- switch levels by typing e.g. '*starnew*'
  - can check current level with 'STAR\_LEVELS'
- switch levels by '*starver tag*' e.g.  
*starver 02d* - to work in SL02gd

## Important:

*If the software you're using was written in a more recent library version than the one you're in – switch immediately*



# Coding standards

---

<http://www.star.bnl.gov/STAR/comp/train/standards.html>

## File Extensions

- **ROOT related code** .h .cxx
- **Plain C++ header files** .hh .cc

## C++ coding style guidelines

- **The *public*, *protected* and *private* keywords must be used explicitly**
- **Self-explanatory English names**
  - *theTotalEnergy*, *setEnergyTable()*, *getTrackPosition()*
- **Member functions should start with a small letter**
  - *StTrack::whatever()*
- **Begin each class data member with a lowercase *m*.**
  - *StDouble mDipAngle;*





# Coding standards

---

- Each **header file** should contain only one class declarations
- Each **class** implementation code should go into a single source file
- Each **header file** must be protected from multiple inclusions

```
#ifndef NAME_HH
#define NAME_HH
...
#endif
```

- Every class should have at least one **constructor** and a **destructor**
- Use of **private/protected**, inline functions
- Avoid **global** variables or functions
- No **hard-coded numbers**. Use of constants (const)
- Use **StType** types



# Data Organization Model

---

1. **DAQ** (Data Acquisition) *file created during event-taking*
3. **DAQ** file gets through **BFC** (Big Full Chain)
3. **Event.root** files created: no hits, instead: tracks fit using different methods and stored using **StEvent** model.
4. **StMuDST**: a **Root** Tree with selected information; tracks are organized and sorted (*StMuDST' s are smaller than the event.root files*)

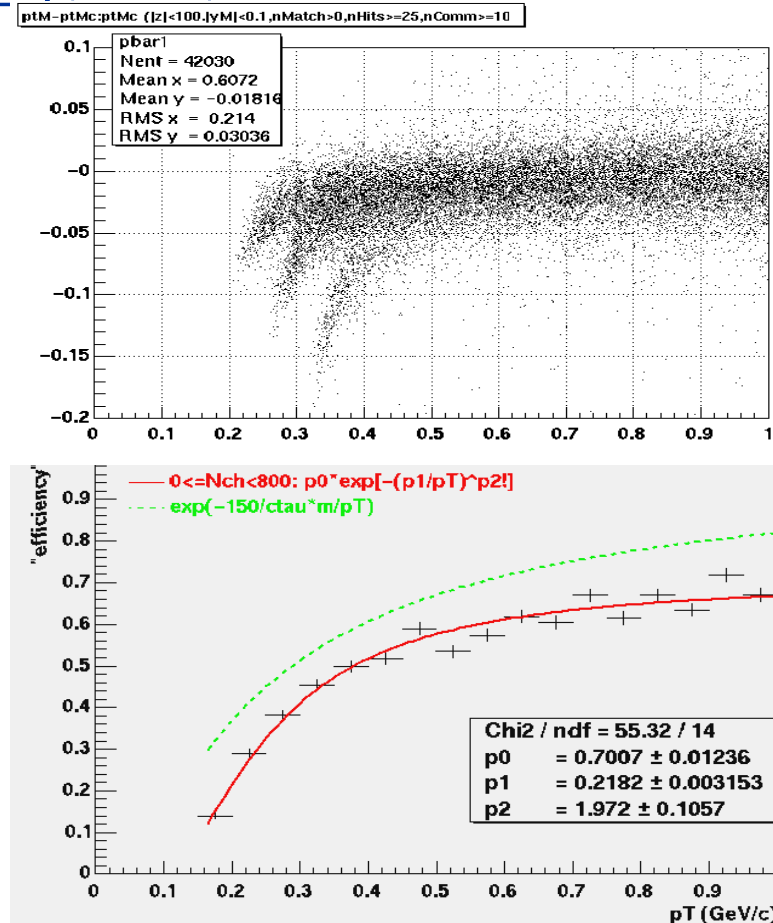


# StAssociationMaker

## A Class for **StMcEvent-StEvent** Object Association

<http://www.star.bnl.gov/STAR/comp/simu/stasso.html>

- Used on MC or Embedded data
- Studies of
  - Energy loss
  - Acceptance
  - Efficiency
  - Hadronic interaction effects
  - Momentum resolution
  - Background contribution
  - .....





# Multimap Usage

Map:
("Brian", 2034322043)
("Manuel", 6313448342)
("Thomas", 2034325829)

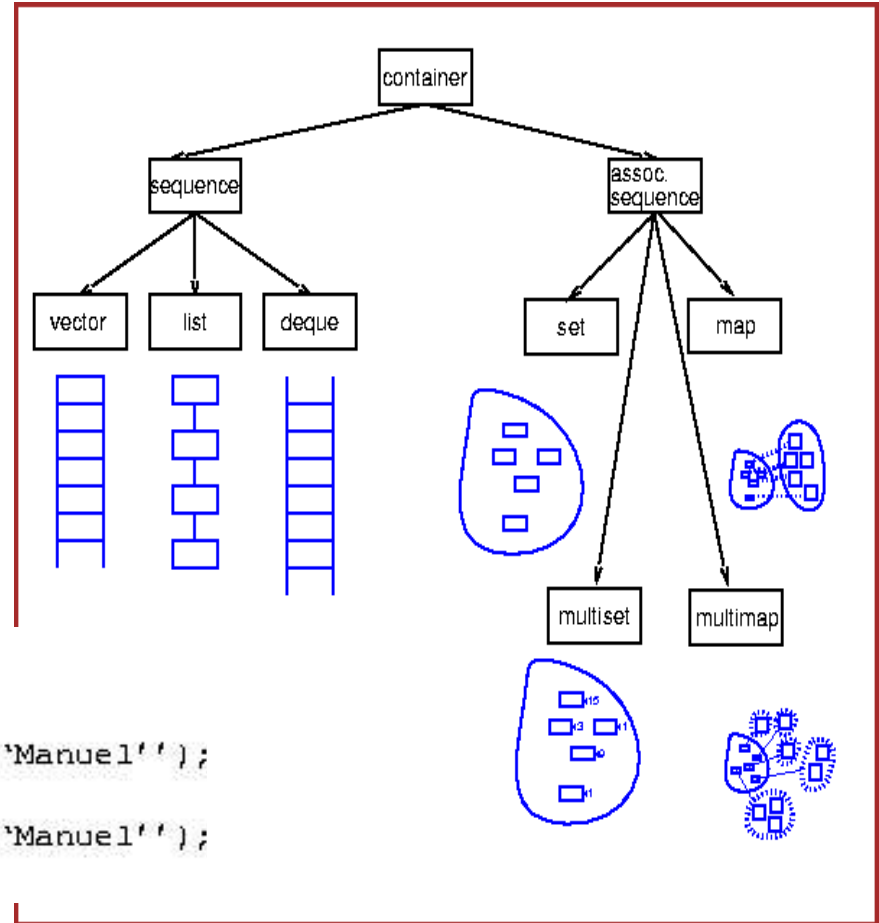
← `phonebook.begin()`

← `phonebook.end()`

Multimap:
("Brian", 2034322043)
("Manuel", 6313448342)
("Manuel", 2034325637)
("Thomas", 2034325829)

← `phonebook.lower_bound('Manuel');`

← `phonebook.upper_bound('Manuel');`



**Methods: Find, Count, Equal\_range**



# Maps

---

TPC Hits

SVT Hits

FTPC Hits

Tracks (using all the previous hit multimaps)

Kink Vertices

V0 Vertices

Xi Vertices

## The defaults are:

### TPC Cuts

X Cut : 5 mm Y Cut : 5 mm Z Cut : 2 mm

Required TPC Hits for Associating Tracks : 3

### SVT Cuts

X Cut : 1 mm Y Cut : 1 mm Z Cut : 1 mm

Required SVT Hits for Associating Tracks : 1

### FTPC Cuts

R Cut : 3 mm Phi Cut : 5 degrees

Required FTPC Hits for Associating Tracks : 3



# How to make a use of it:

---

*\$STAR/StRoot/macros/examples/StAssociator.C*

## 3. Run a chain

- **StEventManager**
- **StMcEventManager**
- **StAssociationMaker**
- **StMcAnalysisMaker**

## 4. Use standard McDst

- **MiniMcDst** (\*.minimc.root)      **Spectra PWG**
- **strangeMuDst** (\*.MuDst.root)      **Strangeness PWG**