CAF STANDARD TOOLS TUTORIAL

Viatcheslav Shary

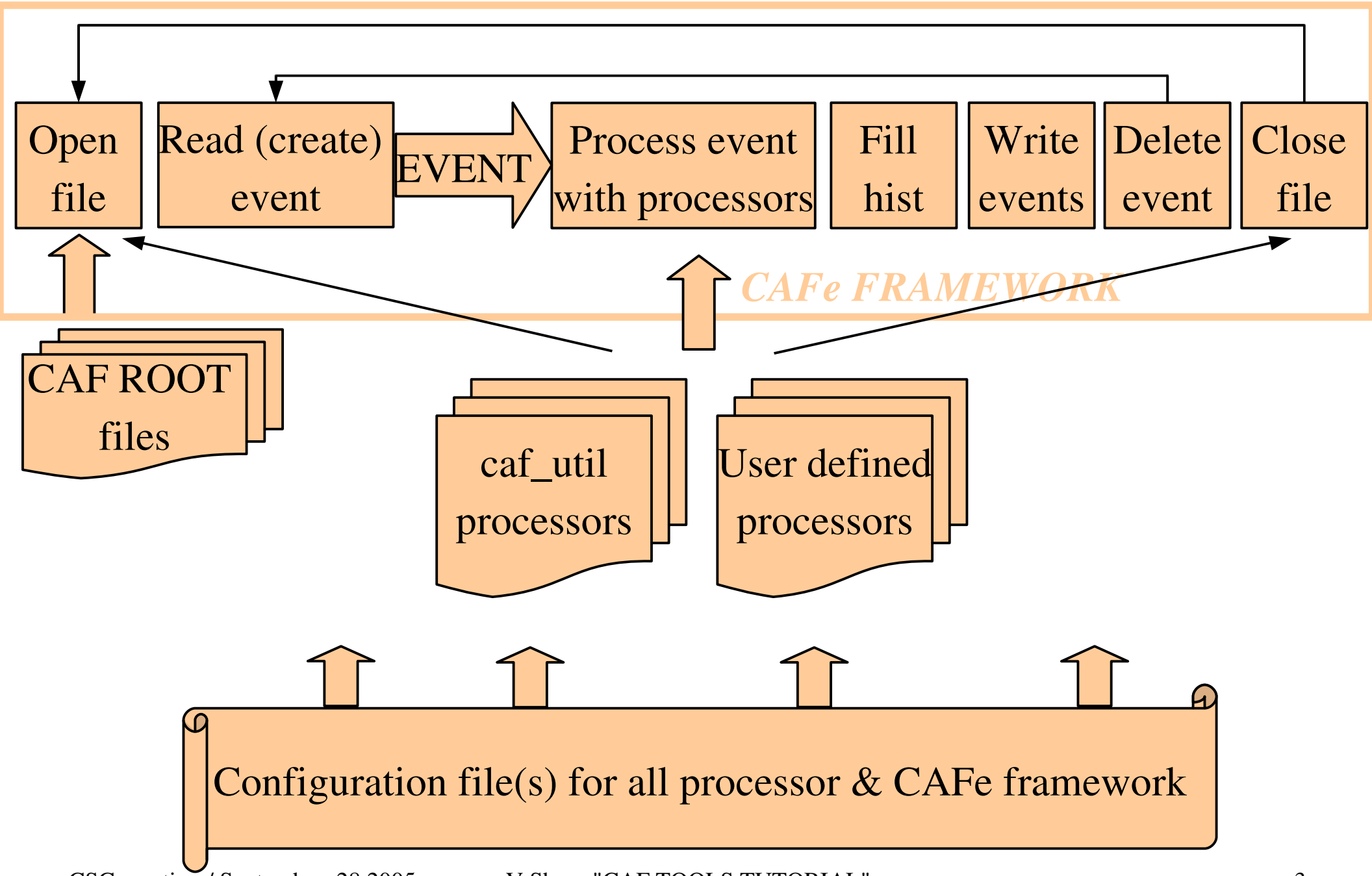
- Philosophy
- CAF tools status
- Quick start guide
- Writing your one selector
- Performance optimization

CSG meeting, September 28, 2005

GLOSSARY

- **CAF** is a common analysis format in the form of ROOT tree. CAF tree structure documented at
  http://www-d0.fnal.gov/Run2Physics/working_group/data_format/caf/
- **CAFe** is an analysis framework. It provides basic functionality to make analysis with CAF trees (packages cafe, cafe_sam). Tutorial:
  http://www-d0.fnal.gov/Run2Physics/cs/caf/docs/Tutorial-Vancouver.pdf
- **CAF UTILITIES** is a set of standard tools to make basic selections, apply MC efficiency corrections and study systematics using CAFe framework (packages caf_util). Documentation:
  *http://www-d0.fnal.gov/cgi-bin/cvsweb.cgi/*checkout*/caf_util/doc/html/index.html*
- **caf_tools** package contains tools to run cafe jobs in batch.

CAFE PHYLOSOPHY



CAF_UTIL STRUCTURE & STATUS

- Idea: collect the commonly used tools, “standardize” them, save manpower, reduce number of software bugs.
- Status & plans: (color legend: processors exist; need to be written)
  - Selectors: Electron, Jets, Muon, Vertex, MET, HT, B-tagging, single track.
  - MC corrections: EM, Muons, Jets, Vertex(?)
  - Systematics tools: JES, objects ID.
  - Tools: Efficiency tool, anything more you want
- Processors communications: save selected objects to a new Tree.

DOCUMENTATION

- Subscribe yourself to d0-caf-users@fnal.gov
- Look the package doc: *caf_util/doc/html/index.html* or *http://www-d0.fnal.gov/cgi-bin/cvsweb.cgi/*checkout*/caf_util/doc/html/index.html*

CAF TOOLS (caf_util package)

Authors:

This package include contribution fron

Introduction

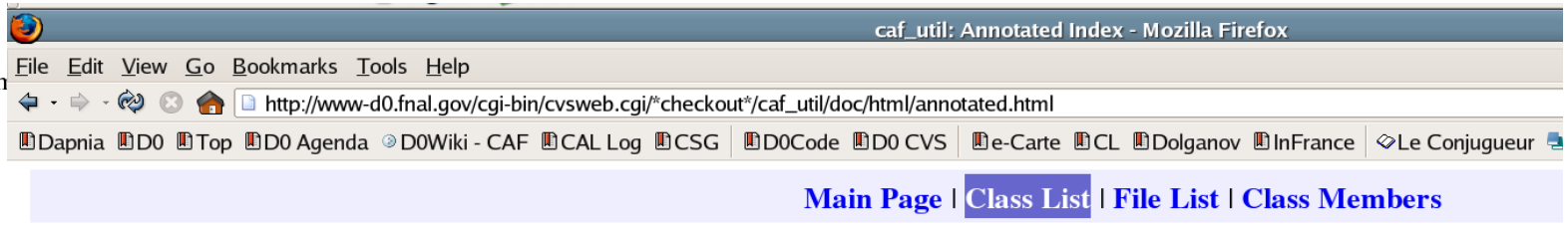
This package includes the various tools to

To start analysis in d0 framework only:

- setup D0RunII p18.02.00 -O SRT_QU
- setenv LINK_SHARED yes
- d0setwa
- set path = ($SRT_PRIVATE_CONTE

For standalone and D0 framework installat

- cvs checkout the recent version of the



caf_util Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

caf_util::ElectronSelector	*This selector will choose electrons satisfying selection criteria and write selected objects to*
caf_util::EventQuality	*This processor removes events with bad event quality*
caf_util::JetSelector	***JetSelector** selects Jets from any jet branch (JCCA, JCCB,...) using standard cuts. It is base*
caf_util::METSelector	*This processor make a MET selection (selection code provided by Patrice Verdier)*
caf_util::MuonSelector	*This processor make a muon selection and write selected objects to the new branch in new*

INSTALLATION (IN A RELEASE)

(updated 28 November 2005)

- setup D0RunII p18.03.00 -O SRT_QUAL=maxopt
- setenv LINK_SHARED yes
- newrel -t p18.03.00 work
- cd work
- d0setwa
- set path = ($SRT_PRIVATE_CONTEXT/shbin/$SRT_SUBDIR $path)
- addpkg cafe p18-br-56
- addpkg cafe_sam
- addpkg caf_util v01-08-02
- addpkg emid_cuts v01-01-01
- addpkg tmb_tree p18-br-42

INSTALLATION (STANDALONE)

(updated 28 November 2005)

- You need to have a kerberised cvs access ot extract packages at clued0 and copy them to your computer
- cvs checkout -r p18-br-56 cafe
- cvs checkout -r p18-br-42 tmb_tree
- cvs checkout -r p18-03-00 cafe_sam
- cvs checkout -r v01-08-02 caf_util
- cvs checkout -r v01-01-01 emid_cuts
- cd include
- ln -s ../caf_util/caf_util caf_util (and the same for other packages)
- cd ..
- source tmb_tree/tools/setup_standalone.csh

END OF INSTALLATION (BOTH TYPES)

- copy the file test.config from the caf_util/doc directory to your working dir
- create file badLBNs.txt with bad lbns.list in your working directory (e.g. empty file)
- copy the emid_cuts/support/emid_cuts.txt file to your working directory
- gmake
- ./test.config
- The following output files will be created:
  - test.html: with analysis efficiencies
  - test.root: with histograms
  - skim.root: with skimmed CAF tree ("TMBTree") and new "SelectedObjects" tree

CONFIGURATION FILE

emacs: test.config

File Edit View Cmds Tools Options Buffers Help

Open Dired Save Print Cut Copy Paste Undo Spell Replace Mail Info Compile Debug News

test.config

```
############# THIS INSTRUCTION RUNS PROCESSORS IN THE GIVEN ORDER ############

cafe.Run: Stat(eff) BadLBNs(lbns) EventQuality(eventQuality) ElectronSelector(electron)
 MuonSelector(muon)   VertexSelector(vertex) JetSelector(jet)  METSelector(met) Topolog
icalSelector(topo) Hist1D(muonpT)

############# GENERAL CAFE CONFIGURATION #####################################

# attach specified packages (emid_cuts packages is used by ElectronSelector)
cafe.Packages: emid_cuts caf_util

# print debug messages if value grater 0
cafe.Debug: 0

# input root file or list of files
#cafe.Input:  /prj_root/1402/caf_write/CAF-EMMU-p18.01.00/CAF-V1-CSskim-EMMU-20050507-1
92359-825502.raw_p17.03.03.root
cafe.Input: listfile:filelist.dat

# number of events to proceed
cafe.Events: 5000

#output file with histograms
cafe.Output: test.root
```

EFFICIENCY TOOL

```
cafe.Run: Stat(eff) BadLBNs(lbns) EventQuality(eventQuality) ElectronSelector(electron)
 MuonSelector(muon)   VertexSelector(vertex) JetSelector(jet)  METSelector(met) Topolog
icalSelector(topo) Hist1D(muonpT)
############# EFFICIENCY COLLECTOR #######################

# efficiency will be printed in the file test.html
eff.Output: test

# sample names
eff.Sample: DATA
```

Inside Processor

```
ElectronSelector(electron) StatPointer stat ;
    event.get("StatPointer") ;
    ... Selection 1 ...
  stat.EventSelected("Selection 1 name") ;
    ... Selection 2 ....
   stat.EventSelected("Selection 2 name") ;
```

tt -> e, mu ANALYSIS EFFICIENCIES -

File Edit View Go Bookmarks Tools Help

http://www-clued0.fnal.gov/~shary/caf_out/test.html

Dapnia D0 Top D0 Agenda D0Wiki - CAF CAL Log CSG D0Code D0 CVS

ANALYSIS EFFICI

SELECTION	DATA
Initial	5000
Event quality: "cal_empty_crate"	5000 (100.000% / 100.000 %)
Event quality: "cal_ring_of_fire"	5000 (100.000% / 100.000 %)
Event quality: "cal_noon_noise"	4998 (99.960% / 99.960 %)
Event quality: "cal_coherent_noise"	4922 (98.479% / 98.440 %)
Electron ID	3727 (75.721% / 74.540 %)
Muon detector eta < 2	3692 (99.061% / 73.840 %)
Muon quality is medium	2810 (76.111% / 56.200 %)
Number of layers >= 3	2657 (94.555% / 53.140 %)
Veto on cosmic muon	2552 (96.048% / 51.040 %)
Matched with central track	2412 (94.514% / 48.240 %)
Muon flobal fit chi square < 4	2156 (89.386% / 43.120 %)

OBJECTS SELECTORS OPTIONS (jet, muon, electron, vertex selectors)

- Read events from the specified branch
- Write objects to the new branch in the new tree
- Write rejected objects to the specified branch in the same tree
- Can sort objects in pT
- ResetBranch option allow to save objects to the tree, without saving it to disk and not waste program memory (will be implemented soon)

```
# input branch for jets
jet.From: JCCB

# output tree name
jet.Tree: SelectedObjects

# save selected ("good") jets to this branch
jet.To: selectedJets

# save rejected ("bad") jets to this branch
jet.RejectedBranch: rejectedJets
```

ELECTRON SELECTOR

- Use emid_cuts package to select electrons with predefined quality.
- You should add file emid_cuts/support/emid_cuts.txt to you path (e.g. copy to the working directory)
- 4 qualities available:
  - *Loose*: isolation < 0.15, EMfrac > 0.9, p_T > 8 GeV, HMx7< 50
  - *Medium*: Loose + Hmx7CC < 12, Hmx8EC < 20
  - *Tight*: Medium + space track match χ^2 > 0.01
  - *TightLhood*: Loose + space track match χ^2 > 0, likelihood > 0.85

```
# the electron selection
electron.Quality: Tight

# minimal number of selected electrons in events
electron.nElectrons: 1
```

MUON SELECTOR

```
# muon quality
muon.Quality: Medium

# number of segments per muon
muon.Nseg: 3

# pT cut in GeV
muon.pT: 5.0

# minimal number of muons in events
muon.nMuons: 1

# remove cosmic muons
muon.cosmicVeto: true

#cut on the DCA significance (DCA / sigma_DCA)
#muon.dcaSignif: 3.0

#isolation criteria

#muon.etHalo: 5.0
#muon.etHaloScaled: 0.1

#muon.etTrkCone: 5.0
#muon.etTrkConeScaled: 0.1
```

JET SELECTOR

- CAF keep separately the jet pT and JES correction. Before working with jets you should choose JES to use: JES or JESMU
- Use “standard” cuts, applied at d0correct level (isGood(), isEM(), isL1Conf()) or user specified one

```
# "normal" JES type (no muon correction)
jet.JES: JES

# standard "d0correct" ID will be used
jet.UseStandardID: true

# minimal number of jets in the events
jet.nJets: 1

# pT cut in GeV
jet.pT: 15.0

# sort output jets in pT
jet.Sort: true
```

VERTEX SELECTOR

- Could use branches with electron and muon from the previous selectors.
- Can sort vertices by min bias probability

```
# sort vertices by min bias probability (the first one has a minimal probability)
vertex.Sort: true

# maximal absolute valued of the vertex z (abs(z) < zmax)
vertex.zmax: 60

# minimal number of tracks associated with vertex
vertex.ntack: 3

# minimall distance in Z between muon and vertices
vertex.dz_muon: 1.0
# name of the muon input branch
vertex.muonBranch: selectedMuons

# minimall distance in Z between electron and vertices
vertex.dz_electron: 1.0
# name of the electron input branch
vertex.electronBranch: selectedElectrons
```

MET SELECTOR

- MET depends on:
  - Jet algorithm type: JCCB or JCC
  - JES type muon corrected or not
  - MET can be corrected with muon or not
- Do not use JES corrected with muon and muon correction to MET together. This combination is not valid because of the muons double counting!

```
#lower MET cut (0 means no cut)
met.minMET: 0.0

#upper MET cut (0 means no cut)
met.maxMET: 0

#jet type (JCCB/JCCA)
met.jetAlgoType: JCCB

# use JES corrected with muons for MET correction
met.jesMuCorrected: false

# use MET corrected with muons
met.MuCorrected: false
```

HT (Topological) SELECTOR

- 3 definition: only jets, jets and leptons, jets and leading lepton
- Must use branch with selected jets and leptons branches (for HT with leptons)

```
cafe.Run: Stat(eff) BadLBNs(lbns) EventQuality(eventQuality) ElectronSelector(electron)
 MuonSelector(muon)   VertexSelector(vertex) JetSelector(jet)  METSelector(met) Topolog
icalSelector(topo) Hist1D(muonpT)

########## TOPOLOGICAL SELECTION  ######################################

# branches names
topo.jetBranch: selectedJets
topo.electronBranch: selectedElectrons
topo.muonBranch: selectedMuons

#lower HT cut (0 means no cut)
topo.minHT: 50.0
topo.minHTleptons: 50.0
#topo.maxHT: 200.0

#number of jets used in HT calculation (negative value means all jets)
topo.njets: 2
topo.nmuons: 1
topo.electrons: 1
```

EVENT QUALITY SELECTOR

- Remove events if any calorimeter flag is fired
  - Missing crate
  - Ring of fire
  - Noon noise
  - Coherent noise

```
cafe.Run: Stat(eff) BadLBNs(lbns) EventQuality(eventQuality) ElectronSelector(electron)
 MuonSelector(muon)   VertexSelector(vertex) JetSelector(jet)  METSelector(met) Topolog
icalSelector(topo) Hist1D(muonpT)
```

WRITING YOUR ONE SELECTOR

- Read the CAF / CAFe tutorial before
  http://www-d0.fnal.gov/Run2Physics/cs/caf/docs/Tutorial-Vancouver.pdf
- Create you one package in the release directory.
- Use caf_util processors as an example:
  - CalEventQuality is a simplest processor which read one branch and reject or accept event
  - MuonSelector read the objects branch, selects objects, save them to a new branch and accept or reject event
  - TopologicalSelector is reading the branches with selected objects, calculate some parameters and accept or decline event.
- Choose a unique name for your processor. “Topological” is not good, “TopEMMUTopological” is a good name.

WRITING YOUR ONE SELECTOR (2)

- Copy files caf_util/SelectorName.hpp, src/SelectorName.cpp src/SelectorName_linkdef.h src/SelectorName_t.cpp
- Create file your_package/scripts/rootrc and add following line: ***+Plugin.Processor: Test Test package_name "Test(const char *)"*** where "Test" is a processor name. You should add such line for every processor you are creating.
- Add your selector to the test.html file. You can save the modified version of this file in your package directory.

```
cafe.Run:  Stat(eff) BadLBNs(lbns) EventQuality(eventQuality) ElectronSelector(electron) MuonSelect
or(muon)   VertexSelector(vertex) JetSelector(jet)  YourSelector(funny_name) METSelector(met) Topol
ogicalSelector(topo) Write(skim)

cafe.Packages: emid_cuts caf_util emu_caf your_package
```

SELECTOR EXAMPLE (*.hpp)

```
#ifndef CAFUTIL_EVENTQUALITY_HPP__
#define CAFUTIL_EVENTQUALITY_HPP__

#include "cafe/Event.hpp"
#include "cafe/Processor.hpp"

namespace caf_util {

  /*! \brief
    This processor removes events with bad event quality.

  */

  class EventQuality : public cafe::Processor
  {
  private:

    bool _cal_empty_crate ;
    bool _cal_coherent_noise ;
    bool _cal_ring_of_fire ;
    bool _cal_noon_noise ;

  public:

    EventQuality(const char *name);
    ~EventQuality() {};
    bool processEvent(cafe::Event &event);

    ClassDef(EventQuality, 0) ;
  };

}
#endif // CAFUTIL_EVENTQUALITY_HPP__
```

SELECTOR EXAMPLE (*.cpp)

```
namespace caf_util {

  EventQuality::EventQuality(const char *name) : cafe::Processor(name)
  {
    // get parameters from the configuration file
    cafe::Config config(name);

    _cal_empty_crate = config.get("cal_empty_crate", true) ;
    _cal_coherent_noise = config.get("cal_coherent_noise", true) ;
    _cal_ring_of_fire = config.get("cal_ring_of_fire", true) ;
    _cal_noon_noise = config.get("cal_noon_noise", true) ;
  };

  bool EventQuality::processEvent(cafe::Event &event)
  {
    //get pointer to statistics collector
    StatPointer stat ;
    event.get("StatPointer", stat) ;

    const TMBGlobal* global = event.getGlobal() ;

    if(_cal_empty_crate && global->cal_empty_crate()) return false;
    stat.EventSelected("Event quality: \"cal_empty_crate\"") ;
    if(_cal_ring_of_fire  && global->cal_ring_of_fire ()) return false;
    stat.EventSelected("Event quality: \"cal_ring_of_fire\"") ;
    if(_cal_noon_noise  && global->cal_noon_noise ()) return false;
    stat.EventSelected("Event quality: \"cal_noon_noise\"") ;
    if(_cal_coherent_noise  && global->cal_coherent_noise ()) return false;
    stat.EventSelected("Event quality: \"cal_coherent_noise\"") ;

    return true  ;
  };
}
ClassImp(caf_util::EventQuality) ;
```

SIZE OPTIMIZATION via SKIMMING

	3JET (39k/event)	EMMU (28k/event)
TrackCal	28,4%	26,0%
Track	18,0%	23,0%
EMcnn	9,0%	7,0%
EMscone	6,5%	5,5%
L3 SUM	4,1%	5,0%
L2 SUM	3,8%	3,5%
L1 SUM	2,7%	3,5%
Muon	2,1%	1,9%
JCCA	2,1%	2,2%
JCCB	2,1%	1,8%
Tau	1,8%	1,8%
BTag SUM	1,7%	4,5%
MetEx	1,7%	2,3%
Met	1,0%	1,2%
CaloCells	1,5%	0,8%
TrackCalJet_JCCB	1,0%	0,7%
TrackCalJet_JCCA	1,1%	1,0%
IsoTrack	0,7%	0,8%
OTHER	11,0%	8,0%

- Skimming a CAF tree:

cafe.Run: EventFlags(flag) Write(skim)

Flag.All: SKIM_EM1TRK

skim.Disable: TrackCal Track

or

skim.Branches: JCCB EMscone

- Note: usually writing limits the processing speed. Writing smaller files will increase the processing speed.

CPU MEASUREMENT & OPTIMIZATION

Measurement:

```
cafe.Run: Stat(eff) BadLBNs(lbns) Timer(t1) Timer(t2)  ... Timer(t9)

t1.Run: EventQuality(eventQuality)

t2.Run: ElectronSelector(electron)

...

t9.Run: Write(skim)
```

You can optimize the reading time, if you read only the leaf you need. Attention! All other leaves are in the undefined state.

```
Variables vars("_evtno");

const TMBGlobal *global = event.getGlobal(vars);
```

CONTRIBUTORS & REQUESTS

- The first version of the caf_util 1 week ago. Only the basic tools has been created. The package will be improved, speed-up and optimized. New tools will be added. This is the right time to make you request and contribution.
- Special thanks to people who are contributing to the development of common tools with new code or actively testing and optimizing the existing one: Venkat, Dag, Mikko, Jens.