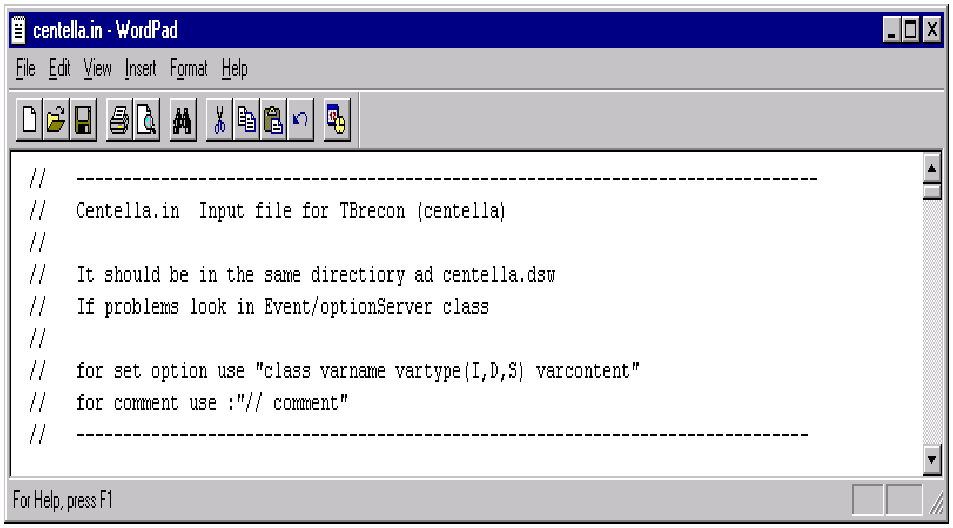
Centella - tb_recon

Software teleconf, July 13th, 2000

Runing algorihtms and applying cuts in tb_recon (centella)

Jose A. Hernando ***UCSC***

The possibilities of centella.in



```
//   ----------------------------------------------------------------------------
//   Centella.in  Input file for TBrecon (centella)
//
//   It should be in the same directory ad centella.dsw
//   If problems look in Event/optionServer class
//
//   for set option use "class varname vartype(I,D,S) varcontent"
//   for comment use :"// comment"
//   ----------------------------------------------------------------------------
```

I) Tb_recon **algorithms** and **filters** in

centella.in

centella.in and line syntax

define input and output files (ROOT trees)

how to scan, read, skip events?

What algorithms to run.

II) Algorithms and filters in **centella**

User implementation of algorithms and cuts

***Taks:** complex algorithms and cuts created at run time (with centella.in). Examples*

III) Histograms and ntuples in **centella**

How to create and add a folder with histograms/ntuples into tb_recon

Centella - tb_recon

I.How to run algorithms and filters in tb_recon

- tb_recon uses an input file:

console version: name of the file as the first argument

default: *centella.in*

- the input file uses the following line syntax:

Syntax

```
//   for set option use "class varname vartype(I,D,S) varcontent"
//   for comment use :"// comment"
//   ------------------------------------------------------------------------------
```

- how to indicate the input/output ROOT tree files:

```
//
// input root tree
//
```

Raw Tree

```
IOfileServer InFile S C:/TBData/rawV1.2/run57.root

//
// output histos/ntuples root
//
```

Recon histos and ntuples

```
IOfileServer histoFile S  C:/TBData/reconV1.1/histoRun57.root

//
// Recon tree file output
//
```

Recon Tree

```
IOfileServer OutFile S C:/TBData/reconV1.1/reconRun57.root
```

Centella - tb_recon

I.How to read and scan events

- Reading and scanning events:

Skip events

Read Events

Scan events

```
//
// There are to filters in tb_recon that can be declared to selRead
//    cutEvtRead   to define how many events to read or skip
//    cutEvtScan   to define the events id to scan or process
// You can add one of the filter in selRead using one of the fillowing lines:
// selRead addCut S cutEvtRead
// selRead addCut S cutEvtScan
//
// examples:
// If you want to skip the 1st 100 events use:
// selRead addCut S cutEvtRead
// cutEvtRead nEvtSkip I 100
// If you want to read only 100 events use:
// selRead addCut S cutEvtScan
// cutEvtRead nEvtRead I 100
// If you want to scan the event 99 and 102 use
// selRead addCut S cutEvtRead
// cutEvtRead evtID I 99
// cutEvtRead evtID I 102

// The program is set to read the 1st 2000 events

selRead addCut S cutEvtRead
cutEvtRead nEvtRead I 2000
```

Centella - tb_recon

I.Algorithms to run in tb_recon

Running Algorithms:

- Before the run starts

  1.- Add algorithms into "*iniOfRun*"

- algorithms by event

  1.- eventTask "*runEvent*"

  2.- Create eventTasks

  3.- Add eventTasks to "*runEvent*"

- After the run ends

  1.- Add algorithms into "*endOfRun*"

```
//
// Section to define the algorithms to run (evtProcess or algorithmTask)
//
// 1.-The algorithms to run at the initial part of the RUN should be included
//    into the algorithmTask named: "iniOfRun"
// example: iniOfRun addAlgorithm S writeListOptions
//
// 2.-The algorithms to run at the end of the RUN should be included into
//    the algorithmTask named: "endOfRun"
// example: endOfRun addAlgorithm S writeByeBye
//
// 3.-The algorithms to be run for every event should be included:
//    into the generation, reconstruction or analysis of one eventTask.
//
//    2.1 There is a main eventTask named "runEvent". This is an example of
//        how to put algorithms into the main eventTask:
//        runEvent generation S loadSiLayers
//        runEvent reconstruction S makeSiClusters
//        runEvent analysis S saveSiClusters
//    2.2 The user can create an eventTask. For example to create
//        ann eventTask for the tracker named "evtTracker" do:
//        processManager newEventTask S evtTracker
//        if now we want to add algorithms to this eventTask, we can do as before:
//        evtTracker generation S loadSiLayers
//        evtTracker reconstruction S makeSiClusters
//        evtTracker analysis S saveSiClusters
//    2.3 To run the eventTask created by the user, it should be included into the
//        main eventTask named "runEvent". Do so with this line:
//        runEvent addEventTask S evtTracker
//    Note that runing an eventTask first they will be executed all the algorithms
//    in the generation, then the ones in the reconstruction, and finally the algorithms
//    in the analysis.
//
//  The user can create: algorithmTask, eventTask and algConditionalTask, and
// add to them algorithms, and cuts. See documentation.
//
```

Centella - tb_recon

I.Creating EventTasks and adding them into “runEvent”

- Creating EventTaks

Create eventTask

generation

Reconstruction

Analysis

```
//  create an eventTask for the tracker
processManager newEventTask S evtTracker
evtTracker generation S loadSiLayers
evtTracker reconstruction S makeSiClusters
evtTracker reconstruction S makeSiTracks
evtTracker analysis S fillSiHistoFolder
evtTracker analysis S saveSiClusters
evtTracker analysis S saveSiRecObjs
```

Load from persistency

Fill histograms

Save to persistency

```
// create an eventTask for the calorimeter
processManager newEventTask S evtCalorimeter
evtCalorimeter generation S loadCalRawLogs
evtCalorimeter reconstruction S makeCalRecLogs
evtCalorimeter reconstruction S makeCalClusters
evtCalorimeter analysis S fillCalHistoFolder
evtCalorimeter analysis S saveCalRecLogs
evtCalorimeter analysis S saveCsIClusterList
```

- And adding them into “runEvent”

Add to “runEvent”

```
//
//   Add now these eventTask(es) into the main eventTask named runEvent
//
runEvent addEventTask S evtTracker
runEvent addEventTask S evtCalorimeter
runEvent addEventTask S evtTower
```

Centella - tb_recon

II.Tb_recon - structure

Application: **CentellaAlg**

Dependent Managers:

- Data Manager

  transient and persistent data

- Process Manager

  Algorithms to run

- Selection Manager

  Cuts to apply

Independent Managers:

- Option Server

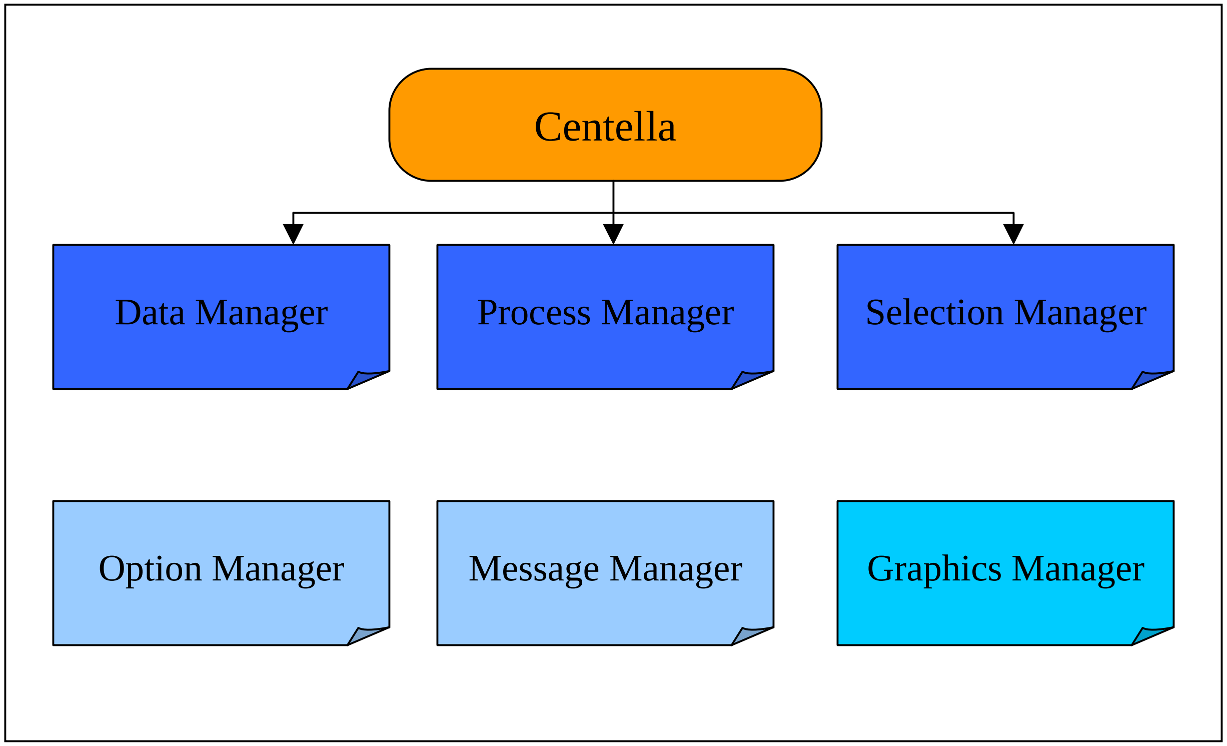
  (set option by the user)

- Message Server

  (print debug, info,etc)

- Graphics Manager

  (event Display)



Centella - tb_recon

II. Algorithms and Cuts in centella

Algorithms and Cuts in centella:

The Users can create algorithsm and cuts

static - coding

dynamic - via the input file

Static algorithms and cuts

1.- implementation:

algorithms inherits from algorihtmVI

cuts inherits from cutVI

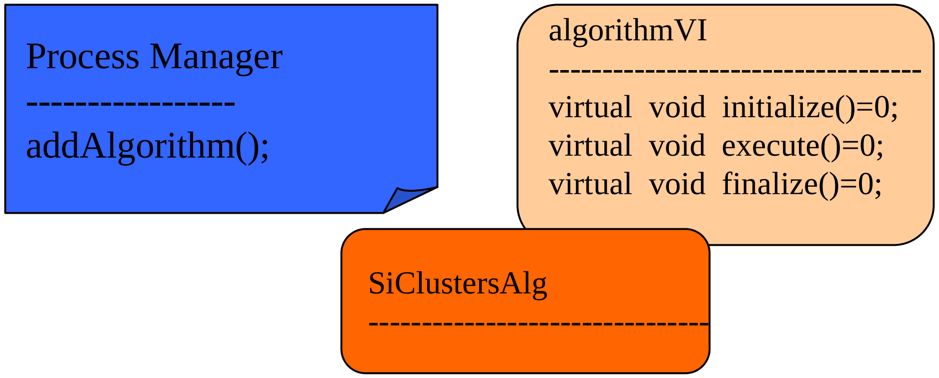
2.- Declare to the Managers

add algorithms to ProcessManager

class : userAlgorithms

add cuts to SelectionManager

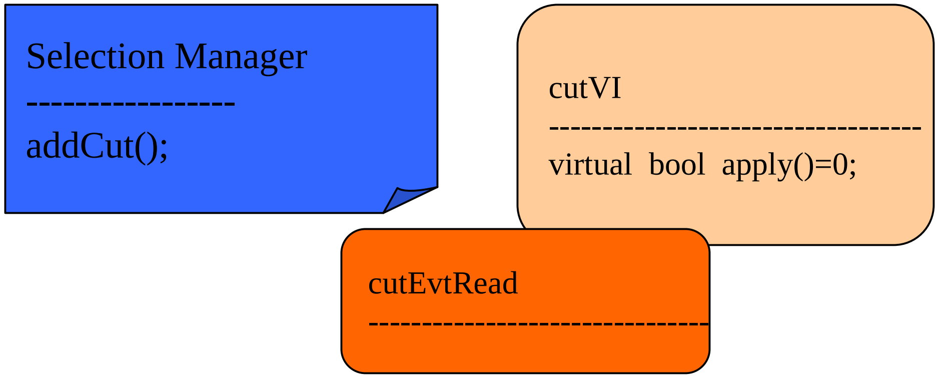
class: userCuts



Process Manager
----
addAlgorithm();

algorithmVI
----
virtual void initialize()=0;
virtual void execute()=0;
virtual void finalize()=0;

SiClustersAlg



Selection Manager
----
addCut();

cutVI
----
virtual bool apply()=0;

cutEvtRead

Centella - tb_recon

II. Algorithms in tb_recon

Algorithms in tb_recon

- System algorihtms:

  examples:

  - “iniOfRun”
  - “runEvent”
  - “endOfRun”

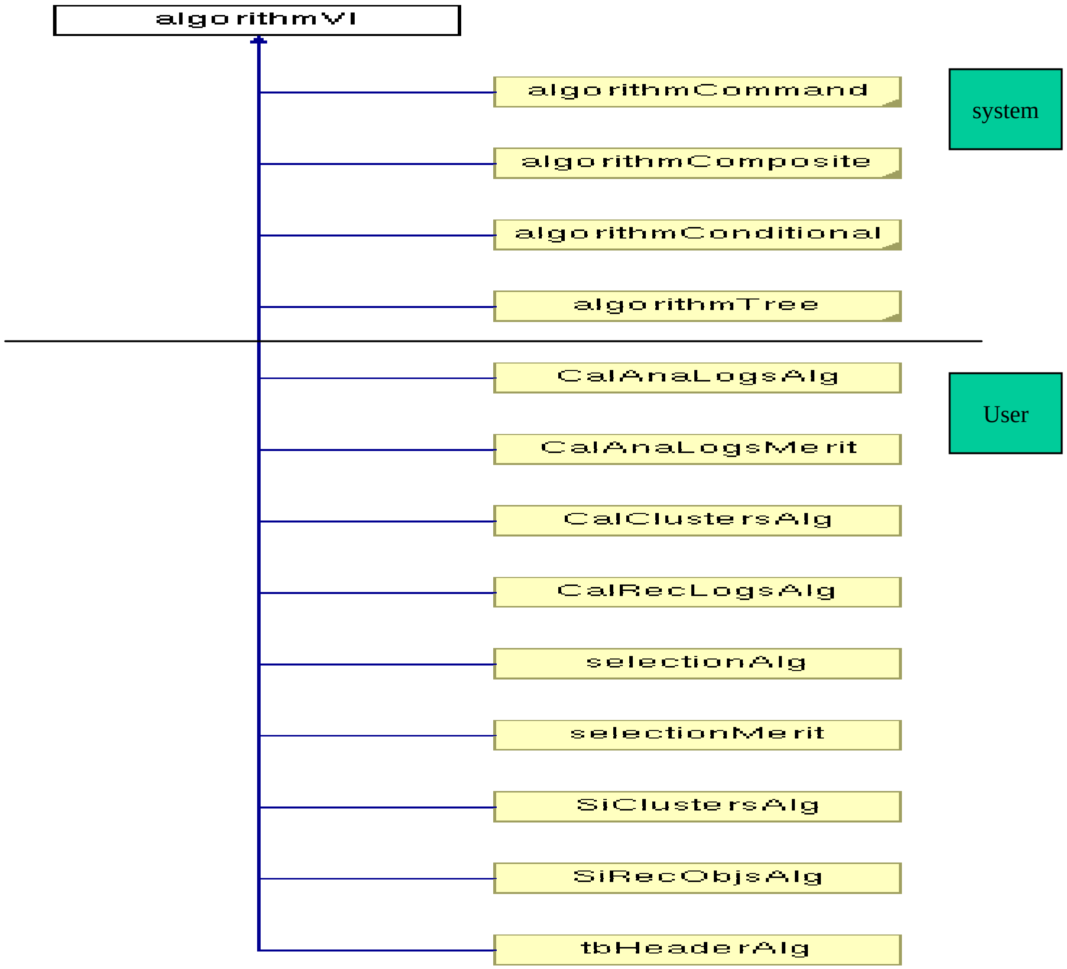
- User defined algorithms
- from histograms/ntuples

  fill histos/ntuples: *fillSiHistos*

- from converters:

  load algorithms: *loadSiLayers*

  save algoritms: *saveSiClusters*



algorithmVI

system

algorithmCommand

algorithmComposite

algorithmConditional

algorithmTree

User

CalAnaLogsAlg

CalAnaLogsMerit

CalClustersAlg

CalRecLogsAlg

selectionAlg

selectionMerit

SiClustersAlg

SiRecObjsAlg

tbHeaderAlg

Centella - tb_recon

II. Algorithms and Cuts in centella

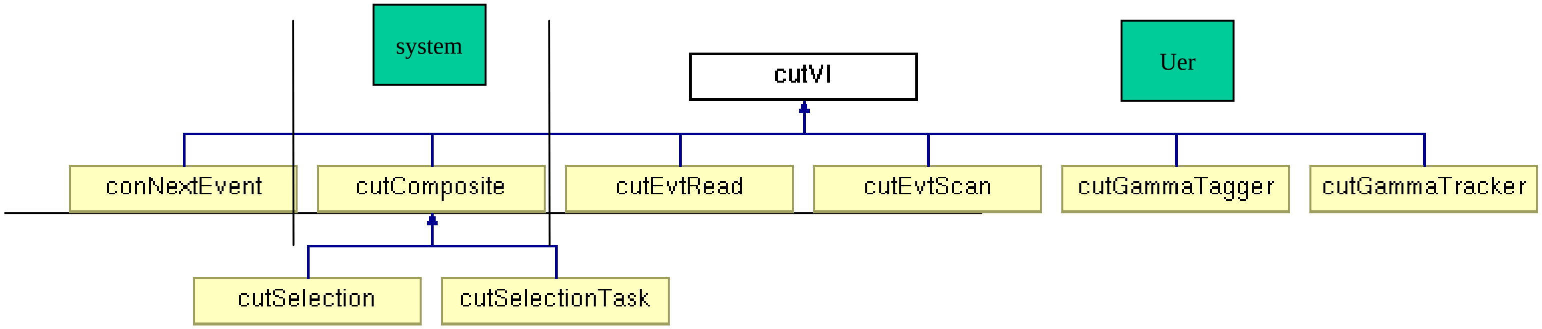
Cuts in tb_recon

- System Cuts:

  examples:

  - “selRead”
  - “selWrite”

- User defined Cuts



Centella - tb_recon

II Dynamic Algorithms and Cuts in centella

Dynamic algorithms and cuts - Tasks

created in the input file (centella.in)

construct complex algorithms and cuts

room for more additions

Example of creating a composite of algorithms
and another one of cuts (filtersr)

Create Algorithm

Add algorithms

```
processManager newAlgorithmTask  S fillHistos
fillHistos addAlgorithm S fillSiHistos
fillHistos addAlgorithm S fillCalHistos
```

Create Cut

Add Cuts

```
selectionManager newCutSelectionTask S selGoodEvent
selGoodEvent addCut S cutCalOK
selGoodEvent addCut S cutTrackerOK
```

Dynamic algorithms:

- **algorithmTask**

  a collection of algorithms

- **eventTask**

  a collection of algorithms

  organized into:

  generation

  reconstruction

  analysis

- **algConditionalTask**

  an algorithm executed only if

  a cut is fulfilled

Dynamic cuts

- **cutSelectionTask**

  a collection of cuts

  it performs the AND of all the cuts

Centella - tb_recon

II Dynamic Algorithms and Cuts in centella

Dynamic algorithms and cuts - Tasks

A more complex example

Create Algorithm

Add algorithms

```
// create an unique algorithm to fill all the histograms
processManager newAlgorithmTask  S fillHistos
fillHistos addAlgorithm S fillSiHistos
fillHistos addAlgorithm S fillCalHistos
```

Create Cut

Add Cuts

```
// create a selection to select good events
selectionManager newCutSelectionTask S selGoodEvent
selGoodEvent addCut S cutCalOK
selGoodEvent addCut S cutTrackerOK
```

Create Algorithm

Add Cut

Add Algorithm

```
// create an algorith to fill the histograms
// nly if the event is good
processManager newAlgConditionalTask S fillOKHistos
fillOKHistos addCut S selGoodEvent
fillOKHistos addAlgorithm  S fillHistos
```

Add Algorithm

```
// add the fillOKHistos algoriithm into
// eventTask into "runEvent"
runEvent analysis S fillOKHistos
```

Centella - tb_recon

III histogram and ntuples in centella

Histograms and ntuples in centella

1.- implementation:

Implement a “myHistoFolder” class

It should inherit from rHistoFolder

define() - book histograms and ntuples

fill() - fill histos and ntuples

*Data** maybe you want to keep a pointer to the data that you use to fill the histograms.

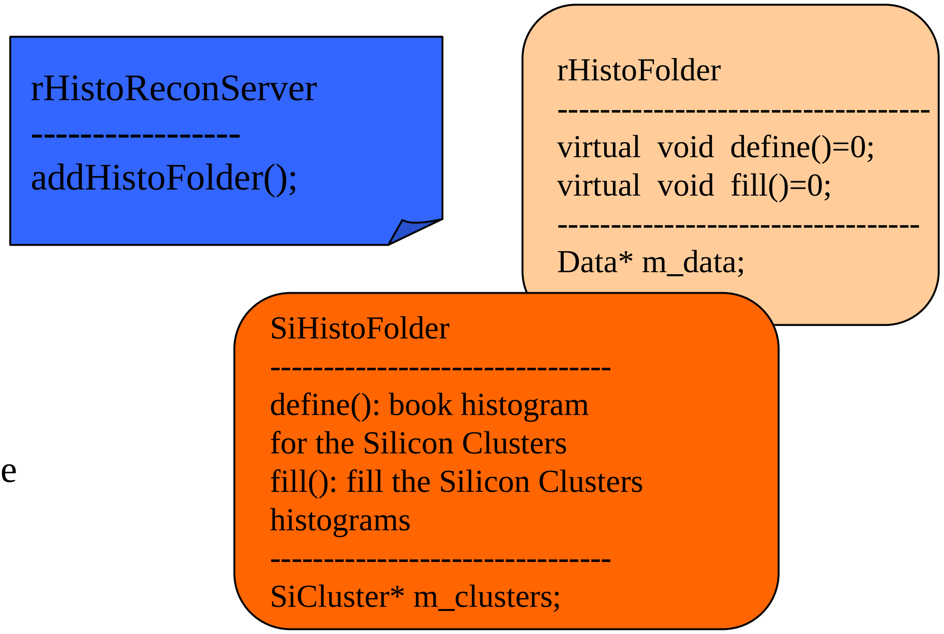
Ask for the data to the dataManager

2.- Declare “myHistoFolder” into the server of rHistoFolder named rHistoReconServer

use *addHistoFolder()*

It will automatically create fillmyHistoFolder algorithm.

3.-Follow examples: the other folder of histograms implemented in tb_recon, (I.e: SiHistoFolder)



```
rHistoReconServer
-----------------
addHistoFolder();
```

```
rHistoFolder
-----------------------------------
virtual  void  define()=0;
virtual  void  fill()=0;
----------------------------------
Data* m_data;
```

```
SiHistoFolder
-------------------------------
define(): book histogram
for the Silicon Clusters
fill(): fill the Silicon Clusters
histograms
-------------------------------
SiCluster* m_clusters;
```

Fill histograms

```
//  create an eventTask for the tracker
processManager newEventTask S evtTracker
evtTracker generation S loadSiLayers
evtTracker reconstruction S makeSiClusters
evtTracker reconstruction S makeSiTracks
evtTracker analysis S fillSiHistoFolder
evtTracker analysis S saveSiClusters
evtTracker analysis S saveSiRecObjs
```