





## Schedule day 1

10/14.00h	Essel (45m)	Go4 v4	Overview Analysis design GUI control
10/14.45h	Essel (15m)	Simple Analysis	First look into analysis code
11/15.00h	Adamczewski -Musch (60m)	Analysis control	Analysis server with controller/observer GUIs MBS/DABC monitoring
12/16.00h	Break (60m)		
13/17.00h	Linev (30m)	The Go4 browser	Analysis, Files, Workspace folders Monitoring, I/O, Treeviewer Update/frozen modes
13/17.30h	Linev (30m)	GUI Editors	Condition usage and editor Markers and conditions General purpose parameters and editor Dynamic list editor
14/18.00h	End		



### Schedule day 2

10/14.00h	Adamczewski -Musch (45m)	Analysis design with Go4	The Go4 analysis steps Modular analysis (analysis classes)
10/14.45h	Adamczewski -Musch (15m)	Go4 Trees in CINT	Go4 libraries, trees, make class
11/15.00h	Essel (60m)	Using macros in Go4	Analysis setup, macro execution in programs, analysis, or in GUI environment API to Go4 environment
12/16.00h	Break (60m)		
13/17.00h	Linev (30m)	User written GUI	Qt designer API to Go4 environment Macro execution
13/17.30h	Linev (30m)	Fitting with Go4	Interactively, save fitter batch using fitter
14/18.00h	End		

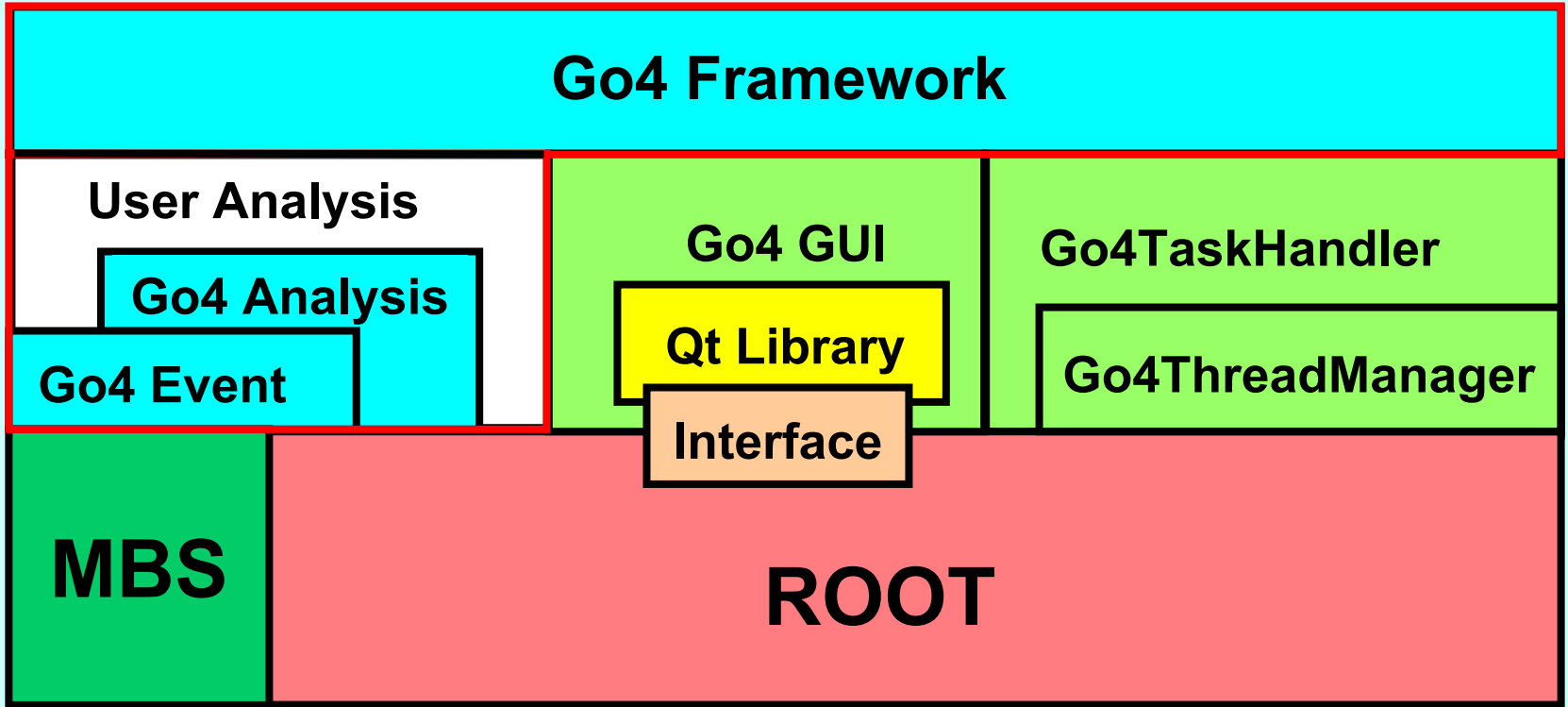




- Go4 features
- Status
- New in v.4
- Analysis framework
- Interactive analysis



- Go4 is a framework for many kinds of experiments (Atomic & Nuclear Physics)
- The analysis is written by the user (C++, unlimited ROOT)
- Go4 provides services and interfaces for analysis
- It runs in batch mode (CINT or compiled, on/off-line)
- or interactive mode (on/off-line):
  - A non blocking GUI controls and steers the analysis
  - The analysis runs independently and can update graphics asynchronously
  - ROOT objects are transported between analysis and GUI task
  - ROOT and Qt graphics are interfaced
  - User may create specific GUIs (Qt designer)





- **Development start:**           **April 1999**
- Go4 v.1.0                            May 2002
- Go4 v.2.0                            November 2002
- Go4 v.2.10                           June 2005
- Go4 v.3.0                            December 2005
- **Go4 v.4.4                            November 2009**
- **Users:**
  - **At GSI:** FRS, SHIP, AP, ESR, Rising, HypHi, HADES online, TASCA...
  - **Outside:** TU Darmstadt, Uni Mainz, Uni Giessen, INFN Milano, Weizman institute, IMPCAS,...





- **Inter-task communication** redesign:  
multiple viewers at one analysis server
- **New object manager** for GUI organization:  
decoupling of functionality and view  
other viewers possible
- Redesign of **GUI elements**:  
browser, viewpanel, editors, new MBS monitor, DIM client for DABC...
- Macro execution in GUI
- **V4.4** can be built with native Qt3 or Qt4
- **V4.4**: standard executable (replaces MainUserAnalysis)
- **V4.4**: Full Go4 distribution for **Linux, Solaris, Windows XP, W7**

**Go4 well established as GSI “standard” analysis framework**  
**Go4 v4.4 released and available at <http://go4.gsi.de>**





# Screenshot of first Go4 v1.0



Go4

File Panel Tools Analysis View Windows Help

Divide Pad into 1 x 1

hist ---None--- Cartesian X: Linear Y: Linear Z: Linear 10 %

Go4 Preview Panel 2

SuperImpose

Go4 Preview Panel 3

SuperImpose

Disk Memory Monitor Analysis

Name	Type
Conditions	Folder
DynamicLists	Folder
EventObjects	Folder
<b>Histograms</b>	<b>Folder</b>
Trees	Folder
UserObjects	Folder

Selection Mode

Date	Time	Description	Type
23.05.02	18.46.46	New analysis status was set.	Info
23.05.02	18.46.57	Analysis TEbAnalysis event classes were initialized.	Info
23.05.02	18.46.57	Analysis nameslist was requested from client..	Info

Tree Viewer

x:      y:      z:      Clear Histogram

Current Ev/s: 1303.479    Average Ev/s: 480.53 s    626463 Event



Browser

- Workspace
  - histo1
    - Panel1
      - 93TC0244\_ASF.root
        - Histograms
          - Calib
            - Pos0
              - Histo0\_C\_P0
              - Histo1\_C\_P0
              - Histo2\_C\_P0
              - Histo3\_C\_P0
              - Histo4\_C\_P0
              - Histo5\_C\_P0
              - Histo6\_C\_P0
              - Histo7\_C\_P0
            - Pos1
              - Histo0\_C\_P1
              - Histo1\_C\_P1
              - Histo2\_C\_P1
              - Histo3\_C\_P1
              - Histo4\_C\_P1
              - Histo5\_C\_P1
              - Histo6\_C\_P1
              - Histo7\_C\_P1
            - Pos2
              - Histo0\_C\_P2
              - Histo1\_C\_P2
              - Histo2\_C\_P2
              - Histo3\_C\_P2
              - Histo4\_C\_P2

File Edit Select Options

Style Binning

Name  
Histo0\_C\_P1::TH1D

Line  
1

Fill  
1

Title  
Calibrated Channel 1 Pos 2

Histogram

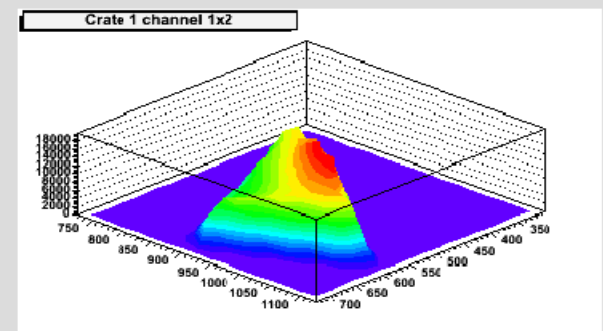
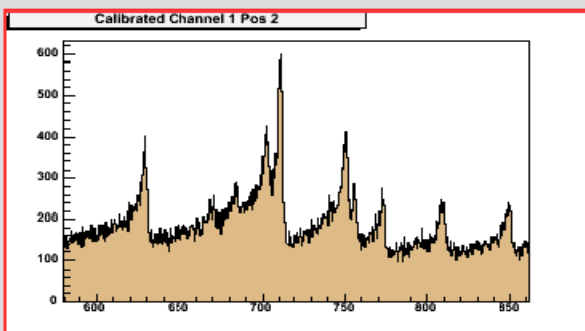
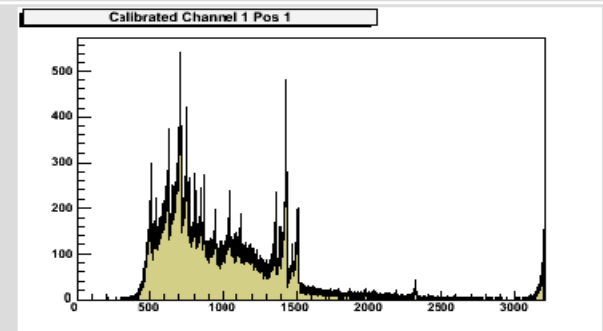
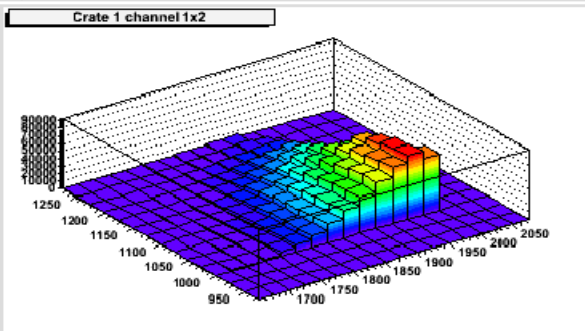
Plot  
 2-D  3-D

Error: No Errors

Style: No Line

Simple Drawing  
 Show markers  
 Draw bar chart  
 Bar option

Marker  
1.0



Marker Modes

loop new

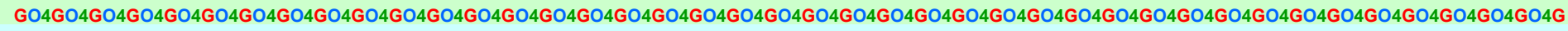
MBS r2-d2 Ev/s Ev kB/s MB

NO SERVER % - file closed - MB file Status Setup SetupML SetupMO 5 s 200 bins trend

Log window MBS monitor

X: Y: Z:

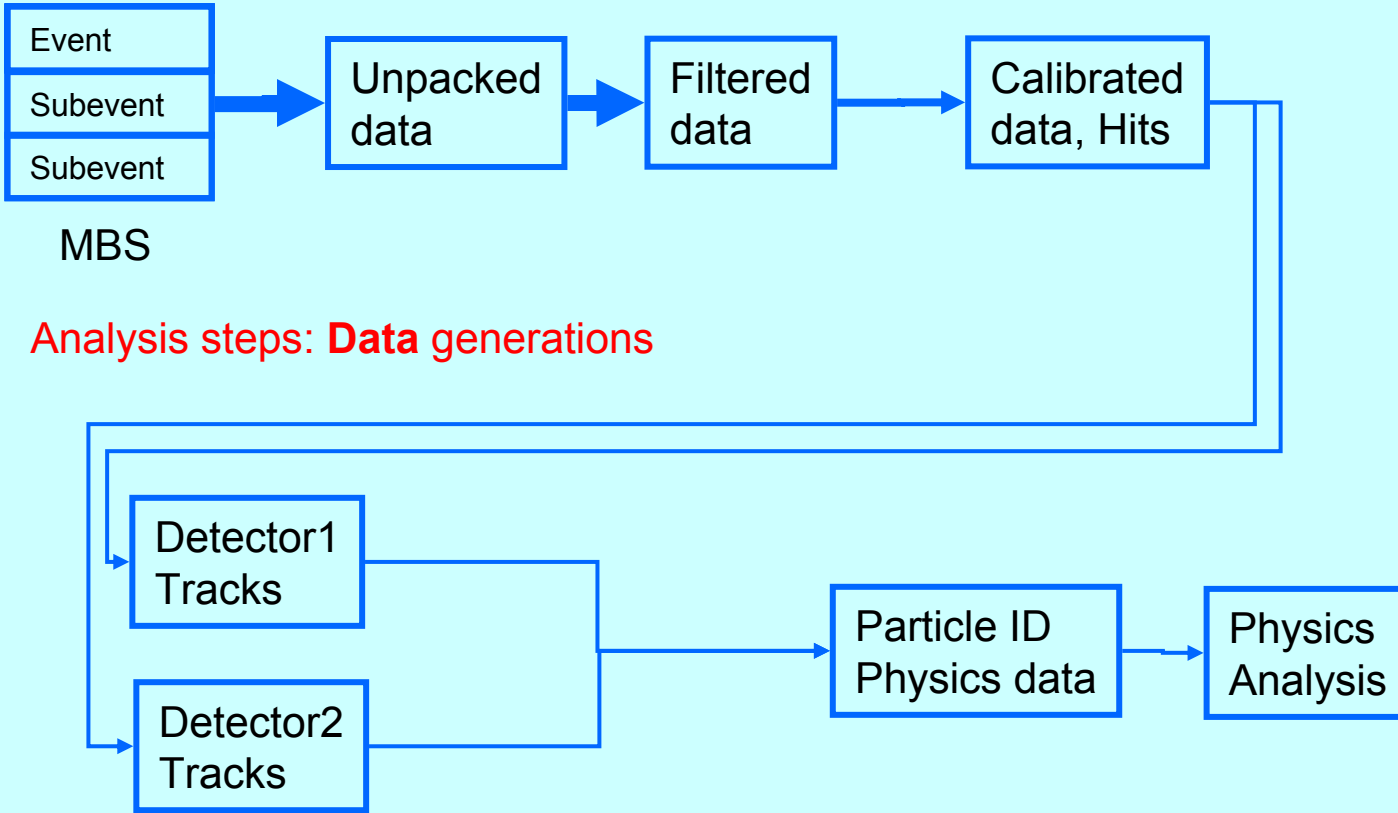
GUI command: rebin("", 2, kTRUE); Divide Pad: 2 x 2 SetPalette 1



**Analysis of event data:**  
**The event loop is executed by framework**  
**User code is plugged in**



# Analysis process: Data flow











# Analysis data flow and event generations



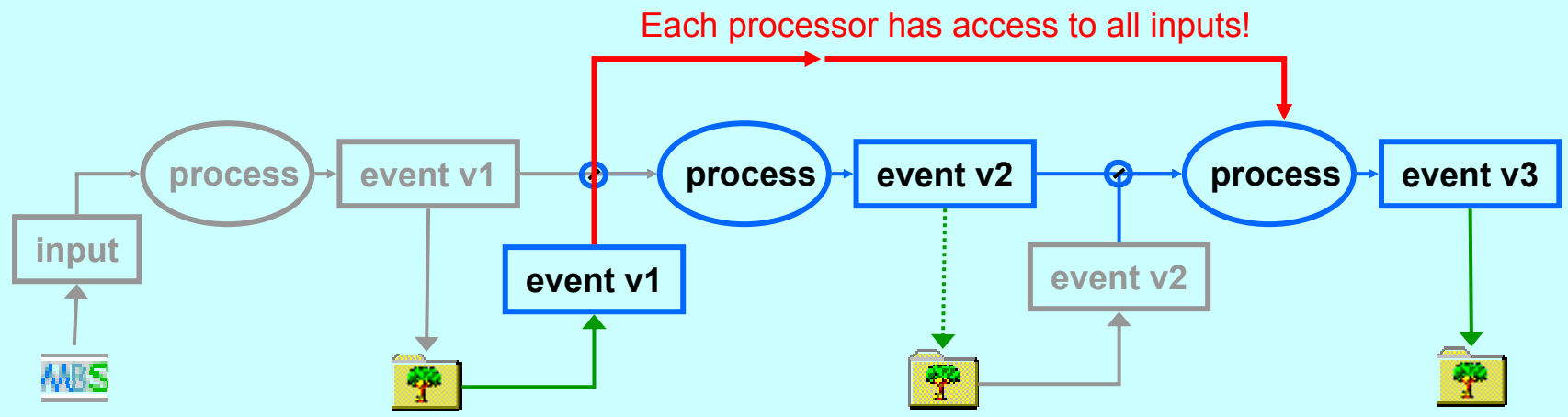
code

data



ROOT tree file

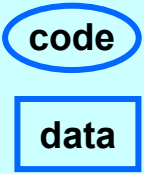
Analysis steps!





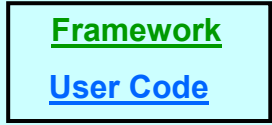
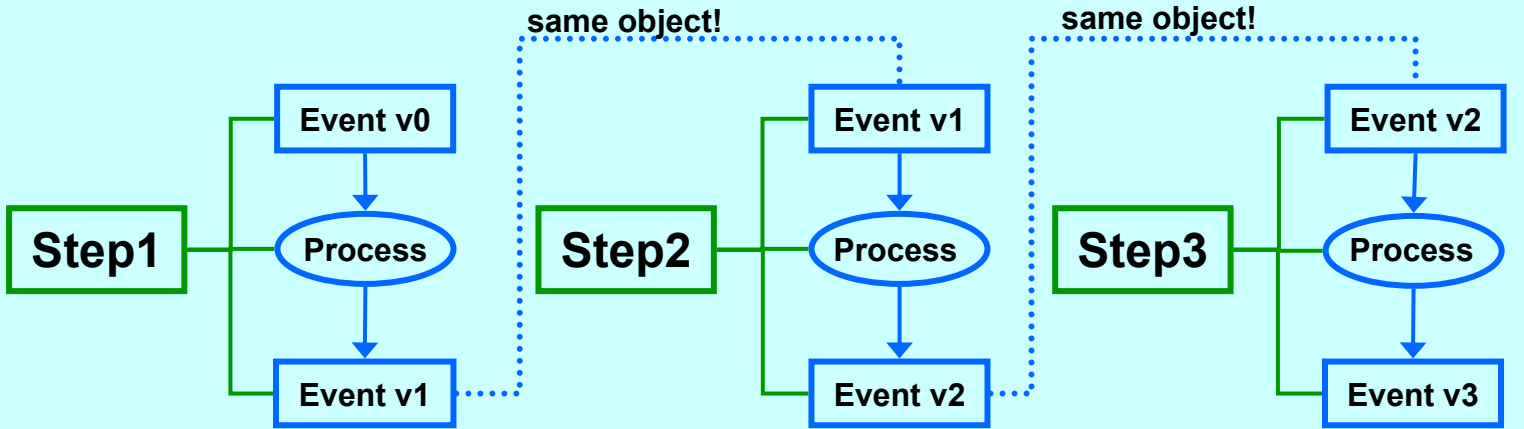


# Analysis steps structure



Chain of analysis steps processed **sequentially**  
 Each step can be **en/disabled (framework)**  
**File input/output can be switched (framework)**

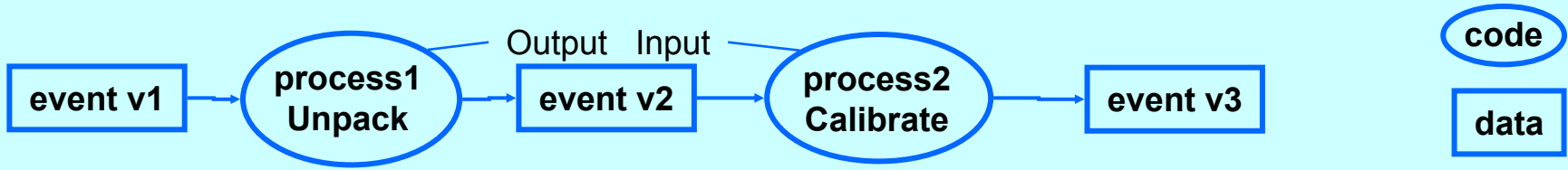
Analysis steps!





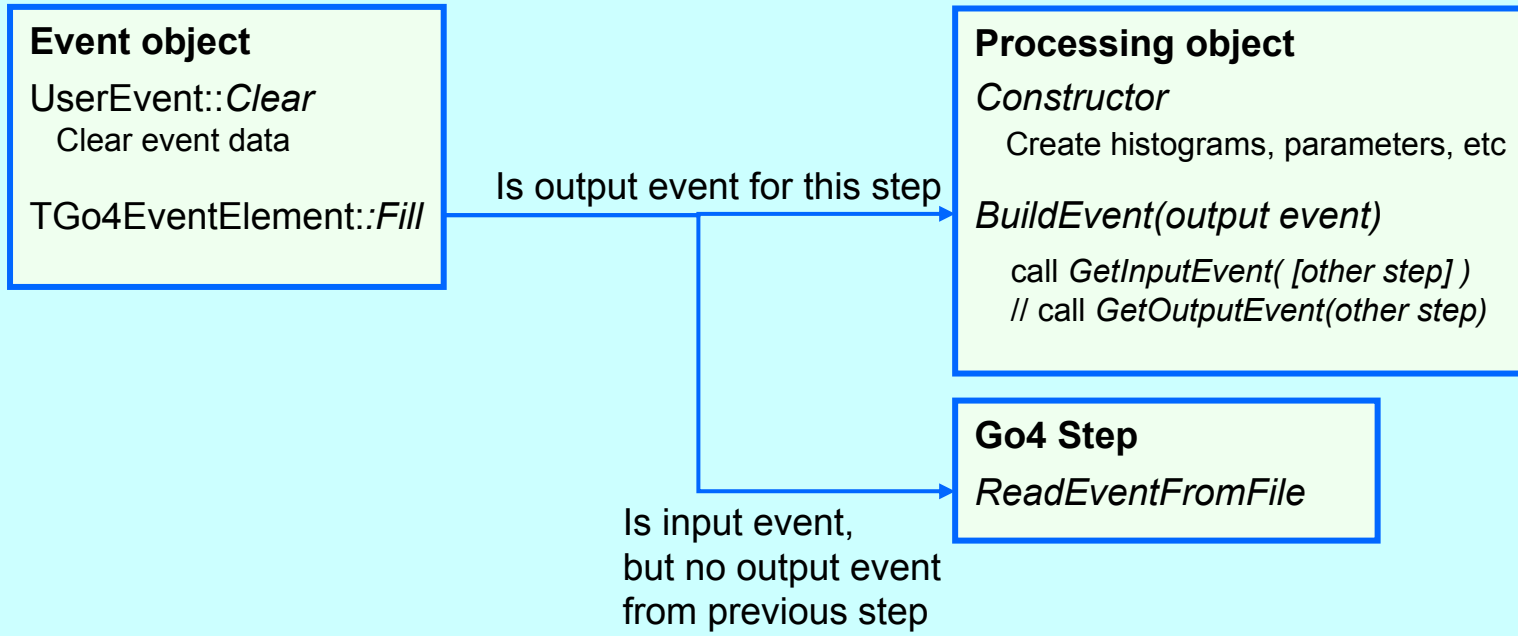


# Go4 analysis, processing chain



User **output event v2** object **calls function** of user **process1** object to get filled  
 User **input event v2** object is either filled **from previous step** or filled **from file** by Go4  
 Each event object like **event v2** is possible both, **input** and **output**

## Controlled by step





**UserAnalysis constructor:** create event, processor, and step objects

```
TGo4StepFactory* cali = new TGo4StepFactory("CaliFact"); // any name
cali->DefEventProcessor("Calibrator","CaliProc"); // object name, class name
cali->DefInputEvent("Unpacked","UnpackEvent"); // object name, class name
cali->DefOutputEvent("Calibrated","CaliEvent"); // object name, class name
TGo4AnalysisStep* calistep = new TGo4AnalysisStep("Calibrator",cali,0,0,0);
AddAnalysisStep(calistep);
```

Factory is a utility class





**UserAnalysis constructor:** create event, processor, and step objects

```
TGo4StepFactory* cali = new TGo4StepFactory("CaliFact"); // any name
cali->DefEventProcessor("Calibrator","CaliProc"); // object name, class name
cali->DefInputEvent("Unpacked","UnpackEvent"); // object name, class name
cali->DefOutputEvent("Calibrated","CaliEvent"); // object name, class name
TGo4AnalysisStep* calistep = new TGo4AnalysisStep("Calibrator",cali,0,0,0);
AddAnalysisStep(calistep);
```

Factory is a utility class

```
TGo4StepFactory* filter = new TGo4StepFactory("FilterFact"); // any name
filter->DefEventProcessor("Filter","FilterProc"); // object name, class name
filter->DefInputEvent(" Calibrated ", "CaliEvent"); // object name, class name
filter->DefOutputEvent("Filtered","FilteredEvent"); // object name, class name
TGo4AnalysisStep* filterstep = new TGo4AnalysisStep("Filter",filter,0,0,0);
AddAnalysisStep(filterstep);
```





# Go4 analysis, create step, configure step



**UserAnalysis constructor:** create event, processor, and step objects

```

TGo4StepFactory* cali = new TGo4StepFactory("CaliFact"); // any name
cali->DefEventProcessor("Calibrator","CaliProc"); // object name, class name
cali->DefInputEvent("Unpacked","UnpackEvent"); // object name, class name
cali->DefOutputEvent("Calibrated","CaliEvent"); // object name, class name
TGo4AnalysisStep* calistep = new TGo4AnalysisStep("Calibrator",cali,0,0,0);
AddAnalysisStep(calistep);

```

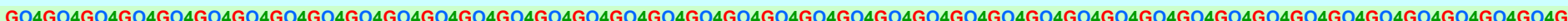
Factory is a utility class

**User setup macro:** configure step

```

step = go4->GetAnalysisStep("Calibrator");
step->SetProcessEnabled(true/false);
step->SetEventStore(new TGo4FileStoreParameter(filename,...));
step->SetStoreEnabled(true/false);
// default event source is output event from previous step
if(<previous step disabled>) // event must be read from file by Go4
    step->SetEventSource(new TGo4FileSourceParameter(filename));
step->SetSourceEnabled(true);

```





Tabs for steps  
2Step example

Analysis Configuration

Unpack xxx Analysis xox

Step Control

Enable Step  Source  Store

Event source

MBS Stream Server

Name: r3g-2

0 all 1 1 s

Event store

Go4FileStore (1 tree/step) (\*.root)

Name: workshop\_events.root

99 100 kB 3  Overwrite

Auto Save File

workshop\_auto.root

Enabled 300 s 5  Overwrite

Analysis Configuration File

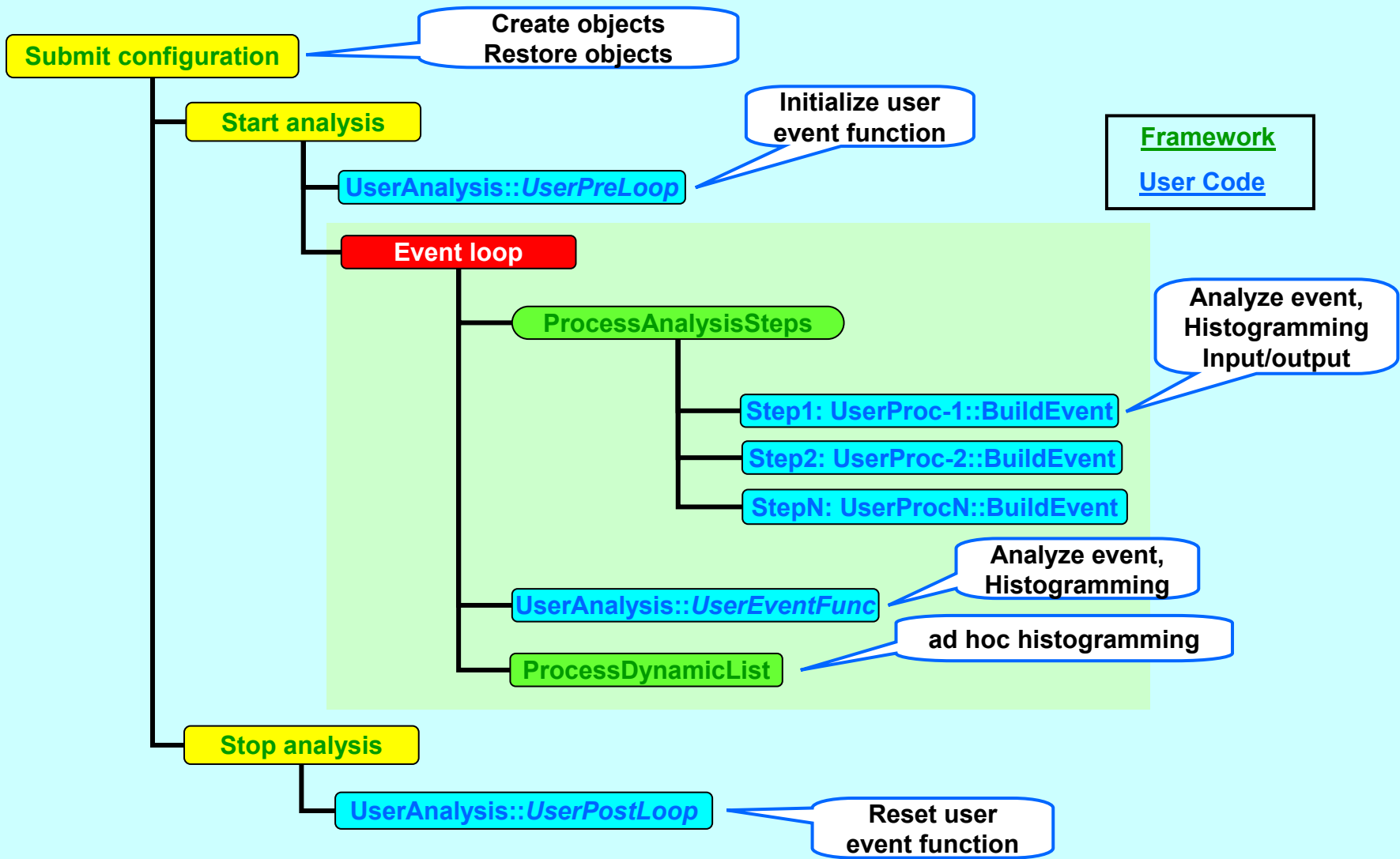
Go4AnalysisPrefs.root

Submit Submit+Start Close



<i>Go4 class</i>	<i>Derived user class</i>	<i>implements</i>	<i>calls</i>
<i>TGo4Analysis</i>	<b>MyAnalysis</b> (optional)	<b>PreLoop</b> <b>UserEventFunc</b> <b>PostLoop</b> (optional)	<b>Create steps (factories)</b> <b>Configure analysis</b> <b>Central management</b>
<i>TGo4EventElement</i>	<b>MyEvent_N</b> (optional)	<b>Clear</b> (optional)	<b>Calls the event function of processor (below):</b> <b>BuildEvent (*MyEvent_N)</b> <b>Argument is event object self</b>
<i>TGo4EventProcessor</i>	<b>MyProcessor_N</b> (required)	<b>BuildEvent</b> (required)	<b>Calls <i>GetInputEvent</i></b> <b>Gets output event as argument</b> <b>Analysis code</b>
<i>TGo4Parameter</i>	<b>MyParameter</b> (optional)	<b>UpdateFrom</b> (optional)	<b>Update all data members, or</b> <b>only a subset</b>







- **Conditions**

- **window condition**: check 1 (2) value(s) against 2 limits (pairs of limits)
- **polygon condition**: check if point (x,y) is inside/outside polygon
- **indexable arrays of conditions**
- **allows for analysis flow control**
- **statistics** (true/false counters)
- **interactive control** (GUI editor) (freeze)

- **Parameters**

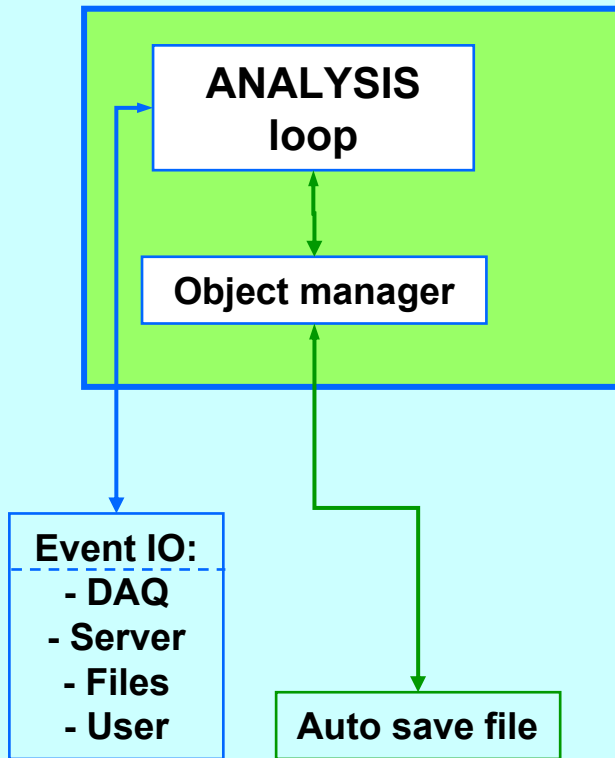
- **User classes** keeping parameter variables
- **interactive control** (generic GUI editor)
- **value protection** (update can be controlled by user function)
- **allows for specific analysis control**
- **"cheap" commands** (executed through editor) easy to implement
- **supports besides atomic data types also fit objects**

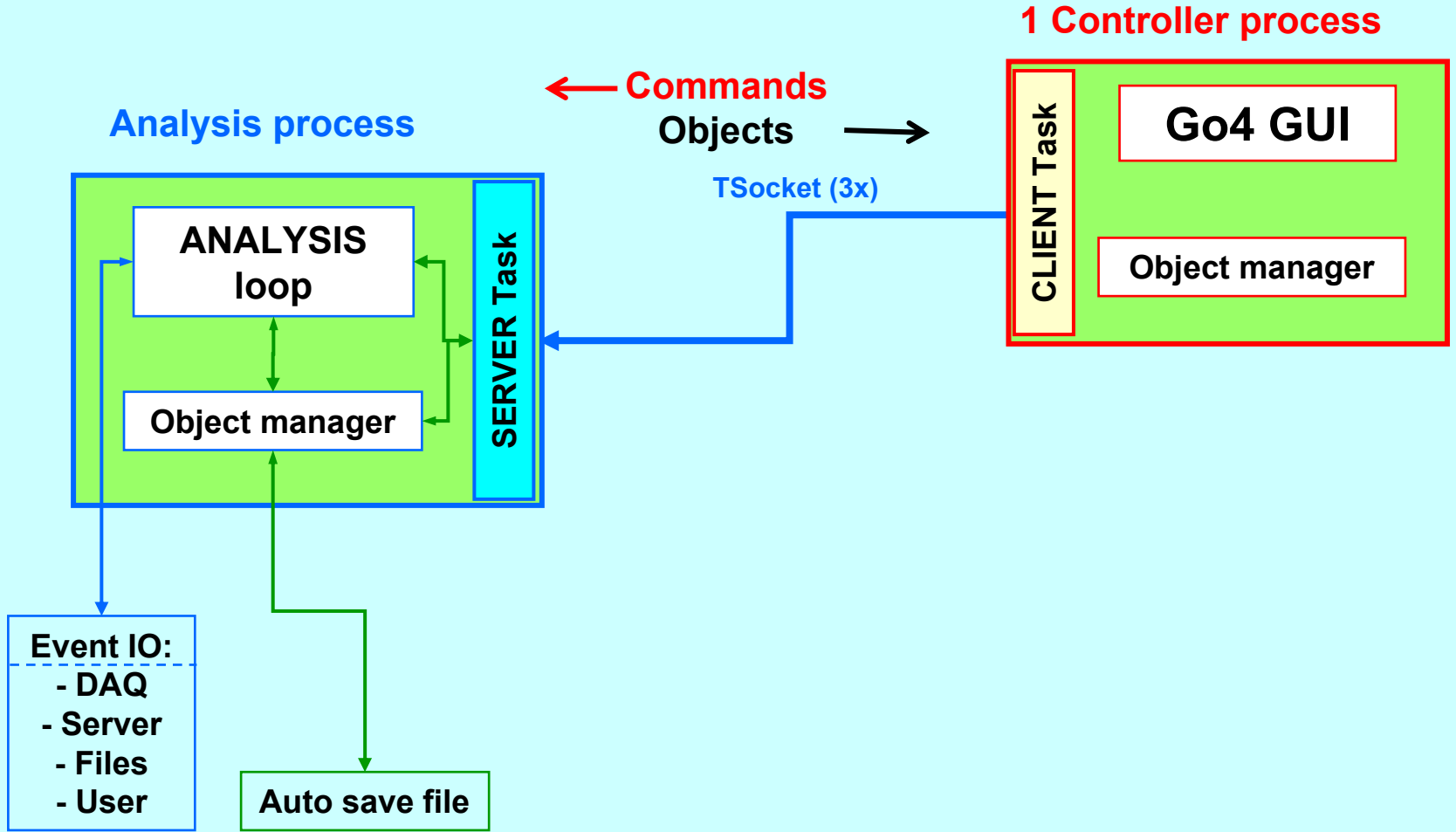
**Stored/restored in/from auto-save file.**

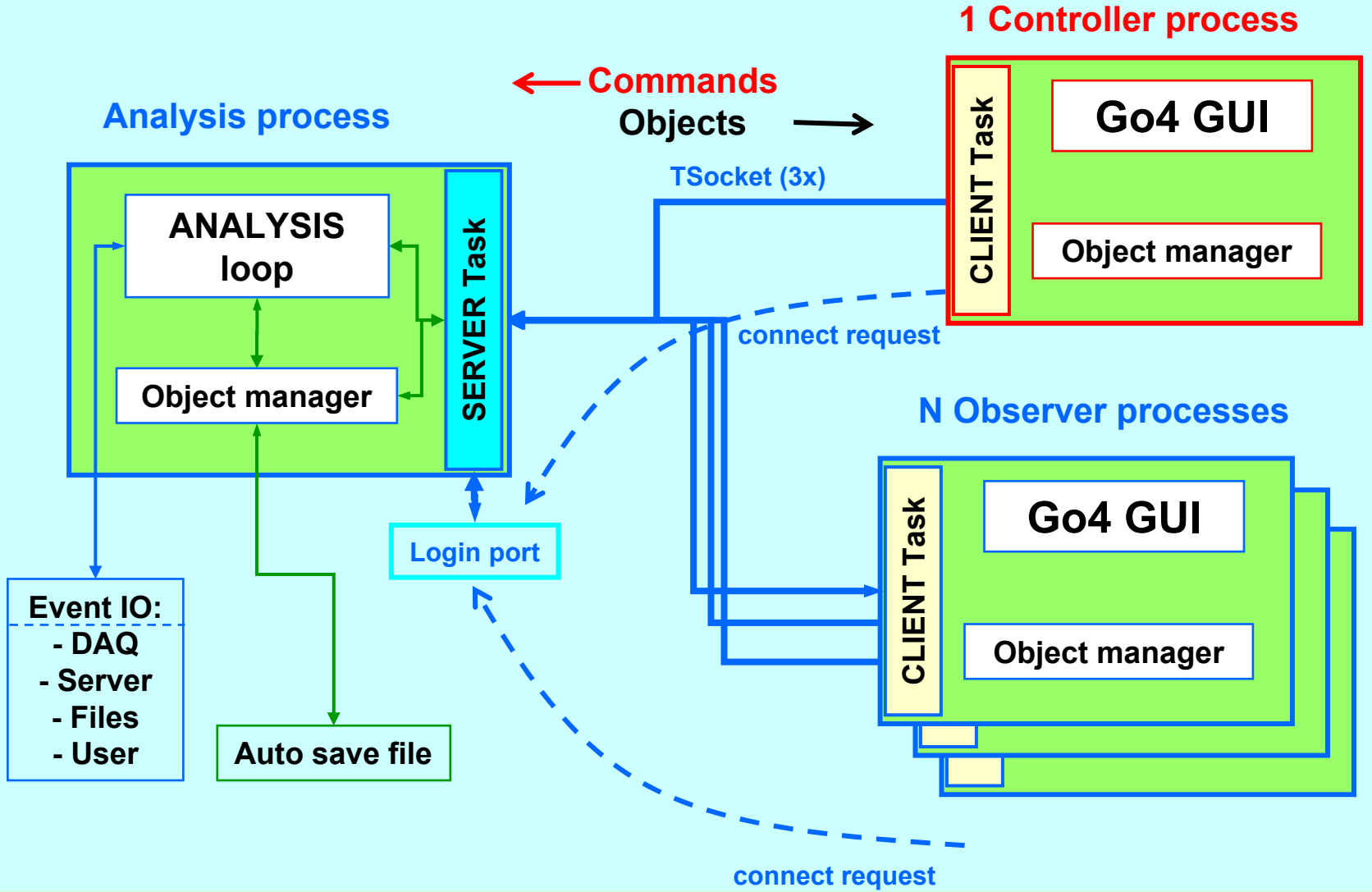


Set up in code or macros

## Analysis process









- Browser
- Graphics
- Interactive tools
- Editors





Browser

Name	Info	RMB popup
Workspace	folder	<input checked="" type="checkbox"/> Flags <input checked="" type="checkbox"/> Info <input checked="" type="checkbox"/> Date <input type="checkbox"/> Time <input checked="" type="checkbox"/> Class <input checked="" type="checkbox"/> Size
└─ histo1	histo title	
└─ d0016.root		
└─ decay-times.root		
└─ Decay_1	frequencies	
└─ Decay_2	frequencies	
└─ Analysis	Controller	
└─ Histograms	All Histogram objects	
└─ Crate1	UserFolder	
└─ Crate2	UserFolder	
└─ Cr1Ch1x2	Crate 1 channel 1x2	
└─ His1	Condition histogram	
└─ His2	Condition histogram	
└─ His1g	Gated histogram	
└─ His2g	Gated histogram	
└─ Conditions	All Condition objects	
└─ Parameters	All Parameter objects	
└─ 123 Par1	This is a Go4 Parameter Object	
└─ DynamicLists	Dynamic List Instances	
└─ Trees	References to trees	
└─ Pictures	Picture objects	
└─ condSet	Set conditions	

- Folders for**
- Workspace (memory of GUI)**
  - Files**
  - Analysis (memory of analysis)**
  - Servers**





Name	Flags	Info	Date	Time	Class	Size
Analysis		Controller			TGo4Analys...	= 692068
Histograms		All Histogram objects	2005-10-04	14:24:51	TFolder	= 686280
Conditions		All Condition objects			TFolder	= 1456
Subfolder		UserFolder			TFolder	= 252
wincon1	spw	Go4 window condition	2005-10-04	14:24:51	TGo4WinCond	164
wincon2	spw	Go4 window condition	2005-10-04	14:24:51	TGo4WinCond	164
polycon	spw	Go4 polygon condition	2005-10-04	14:24:51	TGo4PolyCond	120
winconar	spw	TGo4WinCond	2005-10-04	14:24:51	TGo4CondArra	132
polyconar	spw	TGo4PolyCond	2005-10-04	14:24:51	TGo4CondArra	132
chis1	spw	Go4 window condition	2005-10-04	14:24:51	TGo4WinCond	164
chis2	spw	Go4 window condition	2005-10-04	14:24:51	TGo4WinCond	164
myConny	sdw	1-D window condition	2005-10-04	14:24:51	TGo4WinCond	164
Parameters		All Parameter objects			TFolder	= 2328
123 XXXPar1		This is a Go4 Paramete...			TXXXParameter	920
123 XXXPar2		This is a Go4 Paramete...			TXXXParameter	920
123 sizefitter		This is a Go4 Paramete...			TGo4Fitter...	32
123 specfitter		This is a Go4 Paramete...			TGo4Fitter...	32
123 CaliPar		This is a Go4 Paramete...			TXXXCalibPar	424
DynamicLists		Dynamic List Instances			TFolder	= 884
Pictures		Picture objects			TFolder	= 184
condSet	spw	Set conditions	2005-10-04	14:24:51	TGo4Picture	92
Picture1	spw	Picture example	2005-10-04	14:24:51	TGo4Picture	92
Canvases		All TCanvases			TFolder	
UserObjects		For User Objects			TFolder	= 156
Calibration	spw		2005-10-04	14:24:51	TGraph	100
MultiTest	spw	This is a test multigraph	2005-10-04	14:24:51	TMultiGraph	56
Trees		References to trees			TFolder	
AnalysisxTree		This is a Go4 Status 0...			TTree	
XXXAn1Event.		XXXAn1Event.			TFolder	
XXXAn1Event.TGo4Event...		XXXAn1Event.TGo4EventE...			TFolder	
XXXAn1Event.TGo4Ev...		XXXAn1Event.TGo4EventE...			TFolder	
XXXAn1Event.TGo4Ev...		XXXAn1Event.TGo4EventE...			Bool_t	428
XXXAn1Event.TGo4Ev...		XXXAn1Event.TGo4EventE...			Short_t	428
XXXAn1Event.frData[16]		XXXAn1Event.frData[16]			Float_t	428
EventObjects		Event objects of curre...			TFolder	= 780
EventStores		References to event st...			TFolder	= 52
EventSources		References to event so...			TFolder	= 440















Remote editing of object (data structure) contents

Go4 v4.4.0 @lxxg0523 <Controller name:MyAnalysis>

File Tools Analysis Settings Windows Help

Browser

- Name
- cHis2
- polycon
- winconar
- polyconar
- Parameters
  - XXXParameter**
- DynamicLists
- Trees
- Pictures
- Canvases
- EventObjects

Parameter Editor

Parameter

Analysis/Parameters/XXXParameter - TXXXParameter

Object Members

Name	Type	Value	Comments
frP1	Float_t	100.000000	Offset for calibration
frP2	Float_t	200.000000	Factor for Calibration
fbHisto	Bool_t	1	Enable Histogramming

gauss [redacted] Current Ev/s 69843 Average Ev/s 764658 2010-01-21



### Interactive peak finding and fitting. Save fitter for use in macros

Go4 v4.4.0 @lxg0523 <2>

File Tools Analysis Settings Windows Help

Fit panel

Fitter Tools Settings

Name: Fitter  
Minimizer: Peak finder

Model: Gauss9 of class: TGo4FitModeGaus

background  use buffers

	Fixed	Value	Error	Epsilon
Ampl	<input type="checkbox"/> fix	92.8146	3.29964	
Pos	<input type="checkbox"/> fix	2717.64	0.787184	
Width	<input type="checkbox"/> fix	11.6812	0.668406	

Models:

- Gauss4
- Gauss5
- Gauss6
- Gauss7
- Gauss8
- Gauss9
- Gauss10
- Gauss11

Rebuild + - \*

Use pad Find Fit Draw Pars Active: Panel2. Fitter: Fitter

Panel2: [hDeg120\_CND], ::DataModel

File Edit Select Options  Apply to all  AutoScale

hDeg120\_CND 13:35:09 2009-12-08 histograms.root/hDeg120\_CND





### Histogramming adhoc

Go4 v4.4.0 @lxg0523 <Controller name:MyAnalysis>

File Tools Analysis Settings Windows Help

Browser

- adHocHisto
  - Conditions
  - Parameters
    - XXXParameter
  - DynamicLists
  - Irees
  - Pictures
  - Canvases
  - EventObjects
    - EventStores
    - EventSources
    - EventProcessors
    - Events
      - MbsEvent-10
      - UnpackEvent
        - fiCrate1[16]
        - fiCrate2[16]

Dynamic List Editor

Entry: TGo4HistogramEntry

enable Analysis/DynamicLists/AdHoc

Histogram

Analysis/Histograms/adHocHisto

Event data Condition TreeDraw

X UnpackEvent/fiCrate1[0]

Y

Z

Panel1: [adHocHisto]

File Edit Select >>  Apply to all  AutoScale

histogram title

12000

10000

8000

6000

4000

2000

0

0 100 200 300 400 500 600 700 800 900 1000

gauss 3442 Current Ev/s 34853 Average Ev/s 199 s 6968000 Events 2010-01-21 13:58

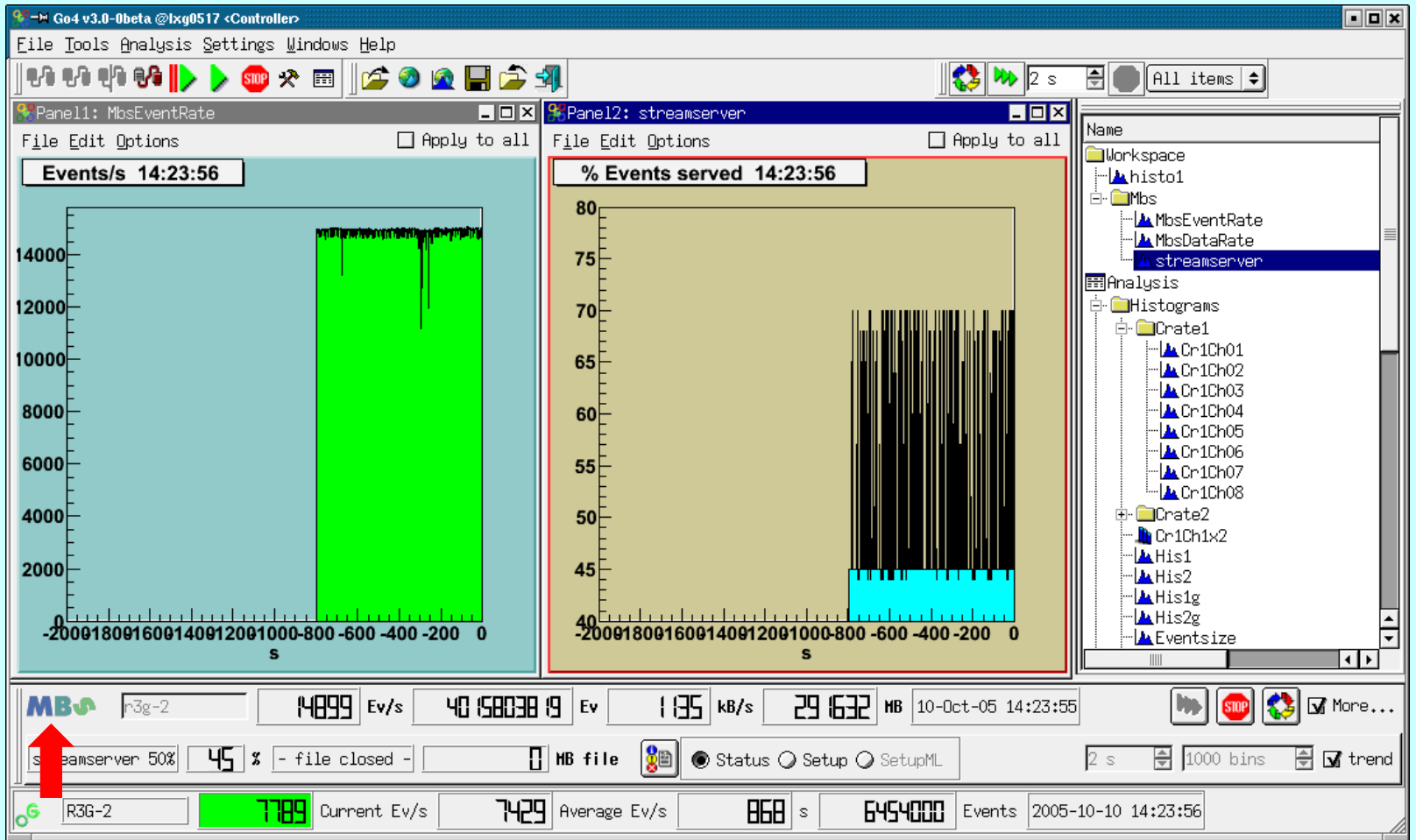
Drag







## Online event ratemeters, trending, remotely inspect status and setup



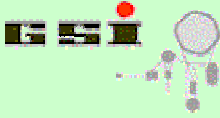




## Have a look to a very simple example

- **MBS data input**
- **No data output**
- **Histogramming (autosave file)**
- **One step (one processor class)**
- **One parameter (one parameter class)**
- **One condition**





## Our *TGo4EventProcessor* class *MyProc*

### MyProc.h

```
#include "TGo4EventProcessor.h"
#include "MyParam.h"
class MyProc : public TGo4EventProcessor {
public:
    MyProc() ;
    MyProc(const char* name);
    virtual ~MyProc() ;
    Bool_t BuildEvent(TGo4EventElement* target);
private:
    TH1                *fHisto;
    TGo4WinCond        *fCond;
    MyParam             *fParam;
    ClassDef(MyProc,1)
};
```



## Our TGo4EventProcessor class MyProc

### MyProc.h

```

#include "TGo4EventProcessor.h"
#include "MyParam.h"
class MyProc : public TGo4EventProcessor {
public:
    MyProc();
    MyProc(const char* name);
    virtual ~MyProc();
    Bool_t BuildEvent(TGo4EventElement* target);
private:
    TH1                *fHisto;
    TGo4WinCond        *fCond;
    MyParam             *fParam;
    ClassDef(MyProc,1)
};

```

### MyProc.cxx

```

#include "MyProc.h"
#include "TGo4WinCond.h"
#include "TGo4MbsEvent.h"
#include "TH1.h"
MyProc::~MyProc(){}
MyProc::MyProc() : TGo4EventProcessor(){}

MyProc::MyProc(const char* name) : TGo4EventProcessor(name){
    fParam = (MyParam*) MakeParameter("Par1", "MyParam");
    fHisto = MakeTH1("I", "His", "Test", 5000, 1., 5001.);
    fCond = MakeWinCond("cond", 100,2000);
}

Bool_t MyProc::BuildEvent(TGo4EventElement* target)
{ // Default empty event. We dont fill any output event here at all
    TGo4MbsEvent* event = (TGo4MbsEvent* ) GetInputEvent();
    event->ResetIterator();
    TGo4MbsSubEvent *psubevt = event->NextSubEvent();
    if(psubevt != 0) {
        Int_t * pdata = psubevt->GetDataField();
        if(fCond->Test(*pdata)&fParam->fillRaw)fHisto->Fill(*pdata);
    }
    return kFALSE;
}

```



## Our *TGo4Parameter* class *MyParam*

### MyParam.h

```
#include "TGo4Parameter.h"
class MyParam : public TGo4Parameter {
public:
    MyParam(const char* name = 0);
    virtual ~MyParam();
    Bool_t fillRaw; // control filling of raw histograms
    ClassDef(MyParam,1)
};
```



## Our *TGo4Parameter* class *MyParam*

### MyParam.h

```
#include "TGo4Parameter.h"
class MyParam : public TGo4Parameter {
public:
    MyParam(const char* name = 0);
    virtual ~MyParam();

    Bool_t fillRaw; // control filling of raw histograms

    ClassDef(MyParam,1)
};
```

### MyParam.cxx

```
#include "MyParam.h"
#include "Riostream.h"
MyParam::MyParam(const char* name) : TGo4Parameter(name),
                                     fillRaw(kTRUE){}
MyParam::~~MyParam(){}
```

### setPar.C

```
void setPar(){
    MyParam *mp;
    mp = (MyParam *) go4->GetParameter("Par1");
    mp->fillRaw=kTRUE;
}
```

```
gROOT->ProcessLine(".x setPar.C()");
```