

Seminar

# Offline and Online Data Analysis with Go4

Andreas Müller

- Abstract:
1. Overview
  2. MBS Data Structure
  3. Source Code
  4. Handling
  5. Analysis



Forschungszentrum  
Dresden Rossendorf

## Go4 - GSI Object Oriented On-line Off-line system

- Based on ROOT
  - complete ROOT functionality
- GUI for online/offline analysis
- Event classes and analysis framework
  - syntax to organize analysis steps and storage
- Browser and tree viewer
  - process standard root files

# Technical Information

- Current Go4 version: Go4 v4.00.01 (Mar08)
- Requirements:
  - Qt 4
  - ROOT  $\geq 5.19/02$
- Works with Debian 4.1 + gcc 4.1.2
- Currently works on fwk28:

user: ntofdaq

pw: \*\*\*\*\*

# Support

- ROOT: [root.cern.ch](http://root.cern.ch)
- Go4: <http://go4.gsi.de>
  
- Go4 documentation and tutorials:  
<http://www-linux.gsi.de/~go4/tutorial/go4.php>
  
- Go4 manual and class index  
<http://www-win.gsi.de/go4/documentation.htm>

# MBS – Multi Branch System

- Data acquisition system – standard at GSI
- Written in C
- Easy data interface to online-analysis like Lea, Go4
- Further information  
<http://www-win.gsi.de/daq/>

# Data Structure

- 32 bit words
- MBS readout event:
  - 1<sup>st</sup> subevent (ProclD=1) : several trigger events
  - 2<sup>nd</sup> subevent (ProclD=2) : several trigger events
- Example: nELBE → time of flight experiment
  - 1<sup>st</sup> subevent: plastic scintillator → trigger events
  - 2<sup>nd</sup> subevent: BaF array → trigger events

# Data Structure - Example

GEO																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	1	1	channel										measurement / counter / data																

First Sub Event (273 words):

```

0x08001388 =>      time flag:    time/ms = 5000
0x100003f0 =>      plastic scaler: channel = 0, Reol. = 0, Measurement = 1008
...
0x17e2c930 =>      plastic scaler: channel = 31, Reol. = 8, Measurement = 182576
0x3400207c =>      TDC Data:    channel = 0, Measurement = 8316
0x30000003 =>      TDC Trailer: status = 0, word counter = 3
0x34002087 =>      TDC Data:    channel = 0, Measurement = 8327
0x30000003 =>      TDC Trailer: status = 0, word counter = 3
...
0x4000606b =>      QDC 1 Data:  channel = 0, UN/OV= 2, Measurement = 107
...
0x40046058 =>      QDC 1 Data:  channel = 4, UN/OV= 2, Measurement = 88
0x44001c64 =>      QDC 1 Trailer: event counter = 7268
...

```

Second Sub Event (452 words):

```

0x5a000300 =>      BaF 1 Header: crate = 0, memorized channels = 3
0x580a069a =>      BaF 1 Data:  channel = 10, UN/OV= 0, Measurement = 132762
0x580b06c3 =>      BaF 1 Data:  channel = 11, UN/OV= 0, Measurement = 198339
0x581002b1 =>      BaF 1 Data:  channel = 16, UN/OV= 0, Measurement = 689
0x5c001c64 =>      BaF 1 Trailer: event counter = 7268
...

```

## Overview - files

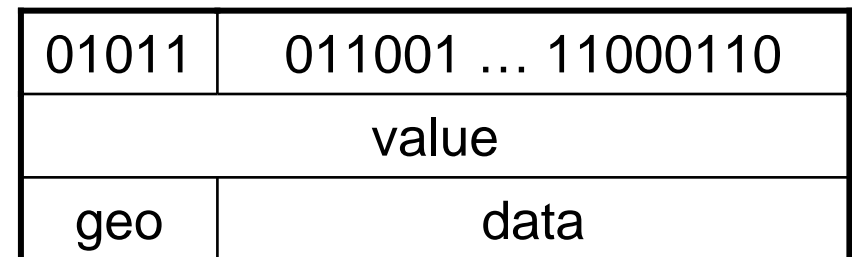
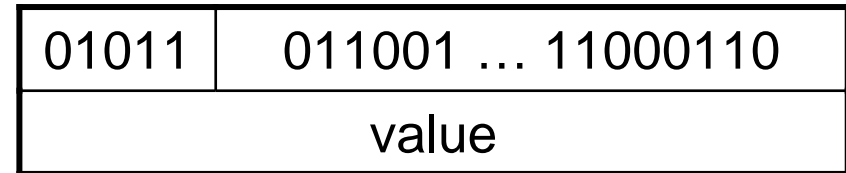
- **Main configuration:**
  - MainUserAnalysis.cxx, TXXXProc.cxx, TXXXParam.cxx  
→ Change syntax for customized analysis
  - TXXXProc.h, TXXXParam.h  
→ associated header files
- **Go4 analysis steps:**
  - setup.C, file.C  
→ provided; Go4 macros
  - AnalysisStart.sh  
→ provided; Analysis startup script
- **Compile:**
  - Makefile, Module.mk, Go4UserAnalysisLinkDef.h  
→ provided; required for compiling, need no change



# Data Handling

```
typedef union {
    unsigned int value;
    struct {
        unsigned int data : 27;
        unsigned int geo : 5;
    } common;
} nELBE_evt;

nELBE_evt myevt;
```



Example:  
myevt.common.geo = 11

# Principle

- **MainUserAnalysis.cxx:**
  - Executable, calls classes TXXXProc, TXXXParam
- **TXXXProc.cxx:**
  - Defines histograms
  - Fills histograms, manages conditions
  - Delivers pointer to 1<sup>st</sup> 32bit word
  - Loops over all words

## Look into the code (TXXXProc.cxx)

```
TGo4MbsSubEvent* psubevt;  
...  
fInput = (TGo4MbsEvent*) GetInputEvent();  
...  
while((psubevt = fInput->NextSubEvent()) != 0) {           // loop over subevents  
    ...  
    pdata = psubevt->GetDataField();                       // pointer to data  
    lwords = psubevt->GetIntLen();  
    ...  
    for(k = 0; k<lwords; k++) {                             // loop over lwords  
        data = *pdata++;  
        ...  
    }  
    ...  
}
```

# Starting Go4

Starting from bash:

- set environment variables (always first step)  
→ “go4log”
- Now, one can:
  1. Edit and compile your source code  
→ “make”
  2. Start Go4 and use the GUI  
→ “go4”

## Analysis via Batch mode (no GUI)

- Make sure of the following:
  - Be in your working directory
  - Your program compiled without errors
- Command to execute offline analysis:

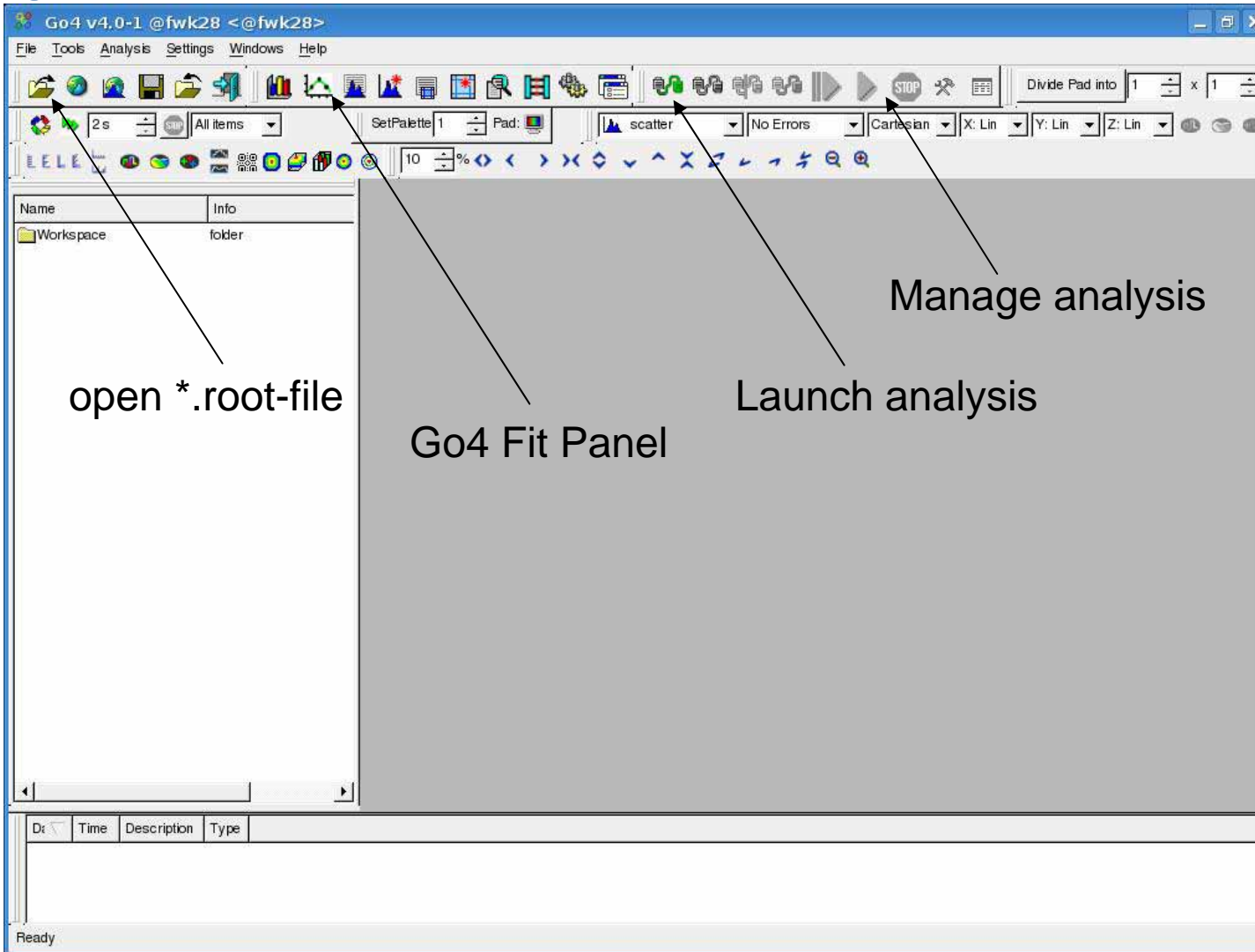
```
./MainUserAnalysis -f [path] [n]
```

[path] ... path of \*.lmd file

[n] ... nb. of events to be processed

→ \*.root file will be added to your directory

# Starting the GUI

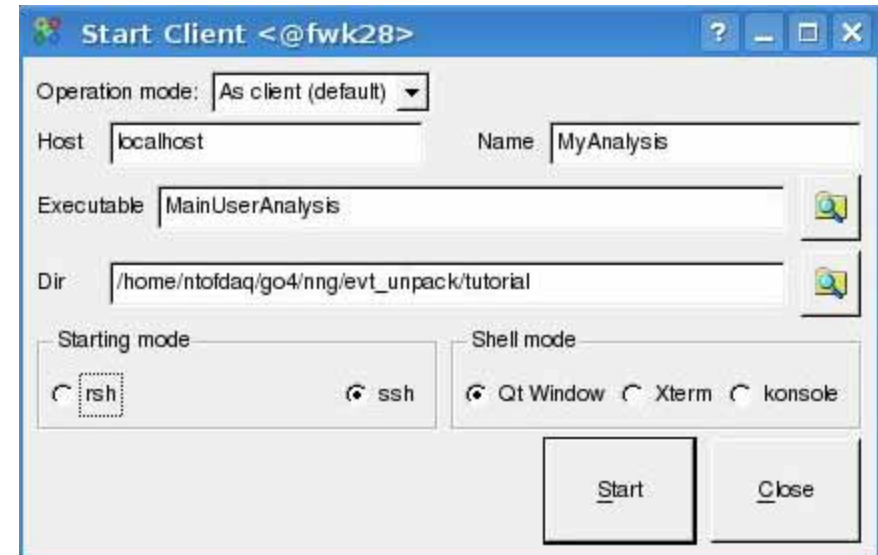


# Evaluating root-files

- Start the GUI
- Open → choose root file (e.g. created via batch analysis)
- Folder structure defined in the source code
- Advantages:
  - Fast and easy method
  - Full functionality for evaluating filled histograms  
→ use fitter, produce pictures, etc.
- Disadvantages:
  - Switch between command line and GUI
  - Wait for batch mode to be finished

# Analysis via GUI

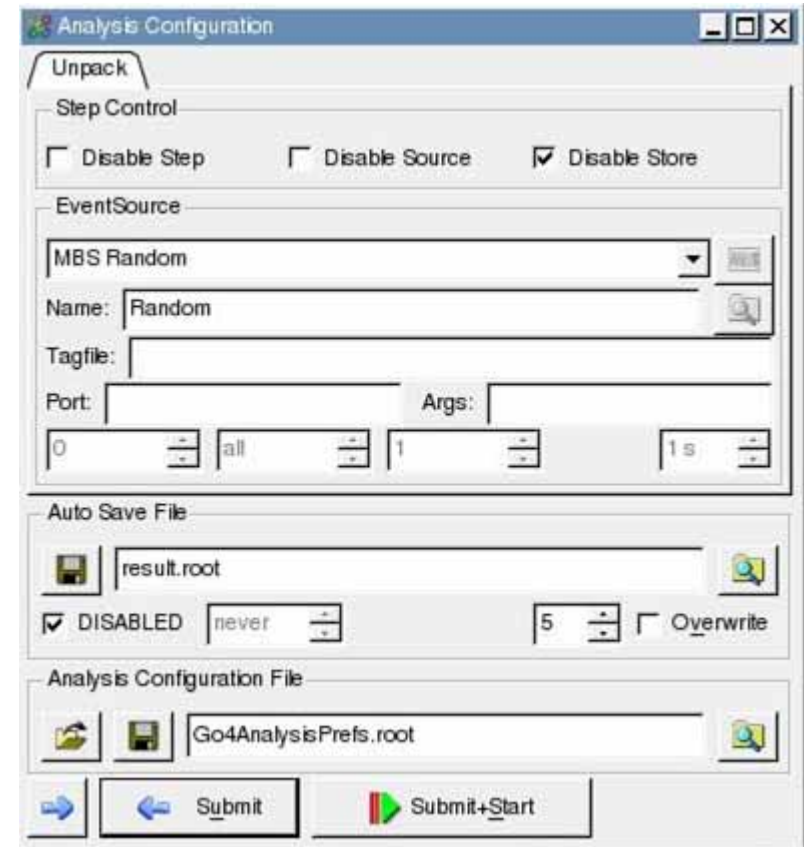
- **Setting up the ssh:**
  - > `cd ~/.ssh`
  - > `ssh-keygen -d`
  - > `cat id_dsa.pub >> authorized_keys`
  - > `ssh user@hostname`
- Program has to be compiled
- Executable → `MainUserAnalysis`
- Click Start and wait





# Analysis Configuration

- Event Source:
  - Offline Analysis: MBS File
  - Online Analysis: MBS Stream Server
- Name:
  - Online Analysis: e.g. rio7
- Submit and close window

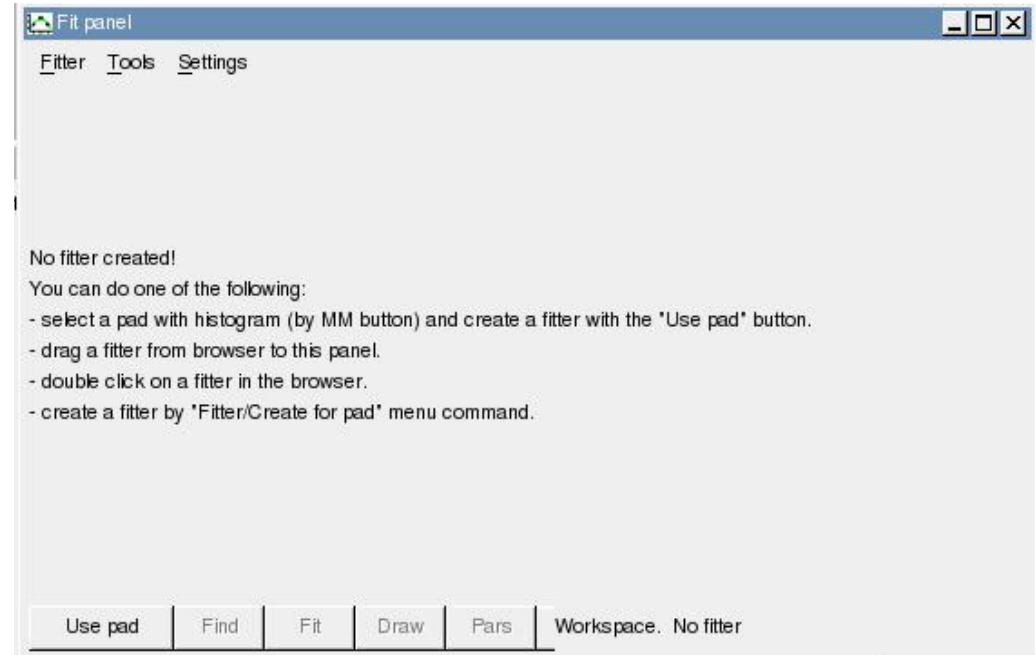


# Running the analysis

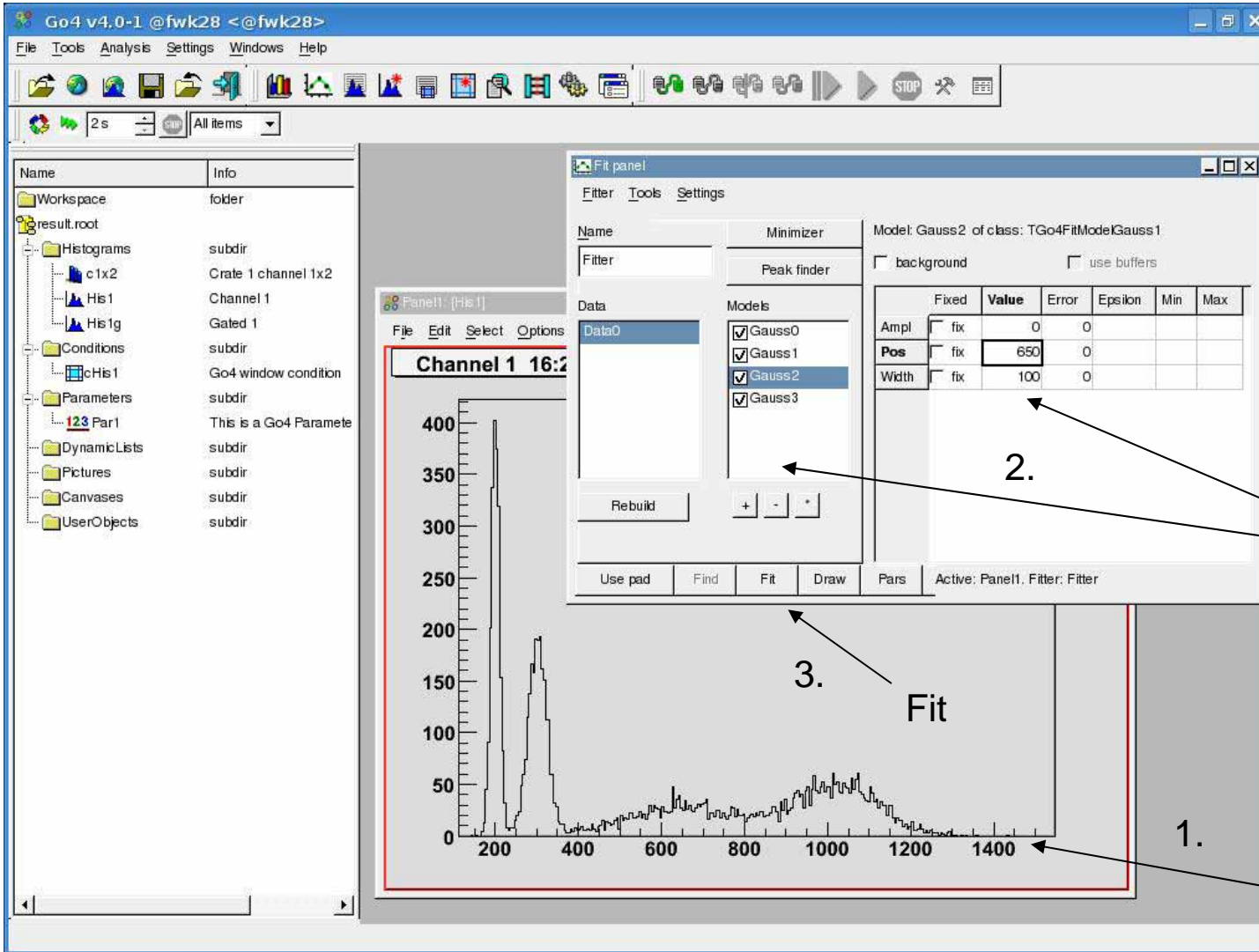
- Start analysis → status bar at the bottom
- Folder structure similar to batch analysis  
→ defined in the source code (TXXXProc)
- Advantages:
  - Look at the data while analysis is running
  - Stopping and continuing every time possible
  - Adjust parameter and conditions
- Disadvantages:
  - Slower
  - GUI might crash (rarely)

## Fitting example – step 1

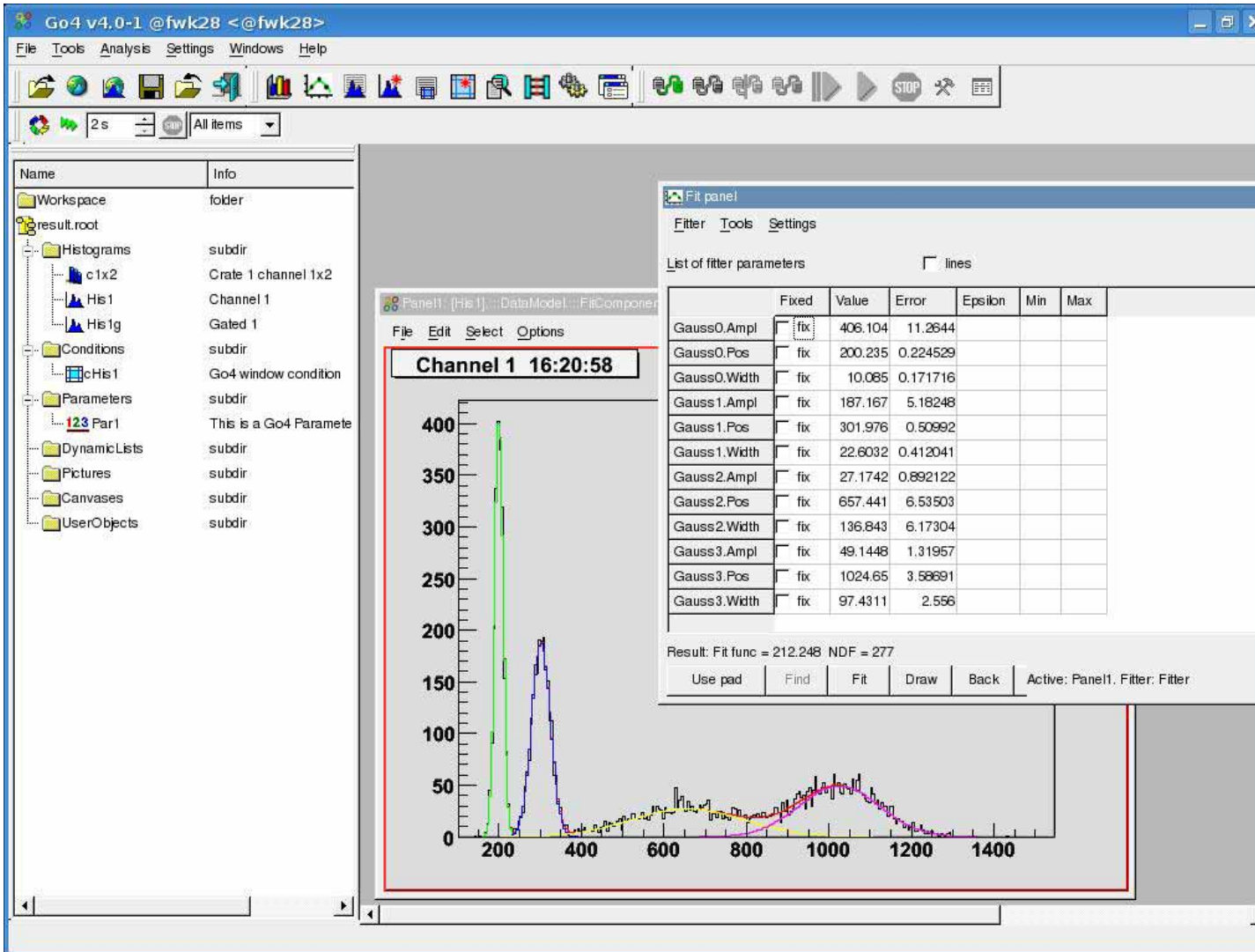
- Open canvas with histogram to fit
- Click on “Show Fit Panel”  
Tools → Wizard  
→ Use Pad
- Add and remove functions with “+/-”
- Initialize parameters
- Click “Fit”



# Fitting example – step 2



# Fitting example – step 3



# Condition example – step 1

The screenshot shows the Go4 v4.0-1 software interface. On the left is a tree view of the workspace containing folders like 'Analysis', 'Conditions', 'Parameters', etc. The main window displays a histogram titled 'Channel 1 17:01:11' with a red hatched region indicating a selected condition. A 'Condition editor' window is open, showing parameters for 'cHis1:His1' and 'Returns Result'. A 'Stats' window is also visible, providing statistical data for the histogram.

cHis1:His1		His1	
X1	= 3.8603E+02	Entries	483498
X2	= 1.3695E+03	Mean	1036
Int	= 1.9129E+05	RMS	967.9
Xmean	= 8.5763E+02	Underflow	1
Xrms	= 2.1597E+02	Overflow	0
Xmax	= 1.0033E+03	Integral	4.835e+05
Cmax	= 2.0450E+03	Skewness	1.069

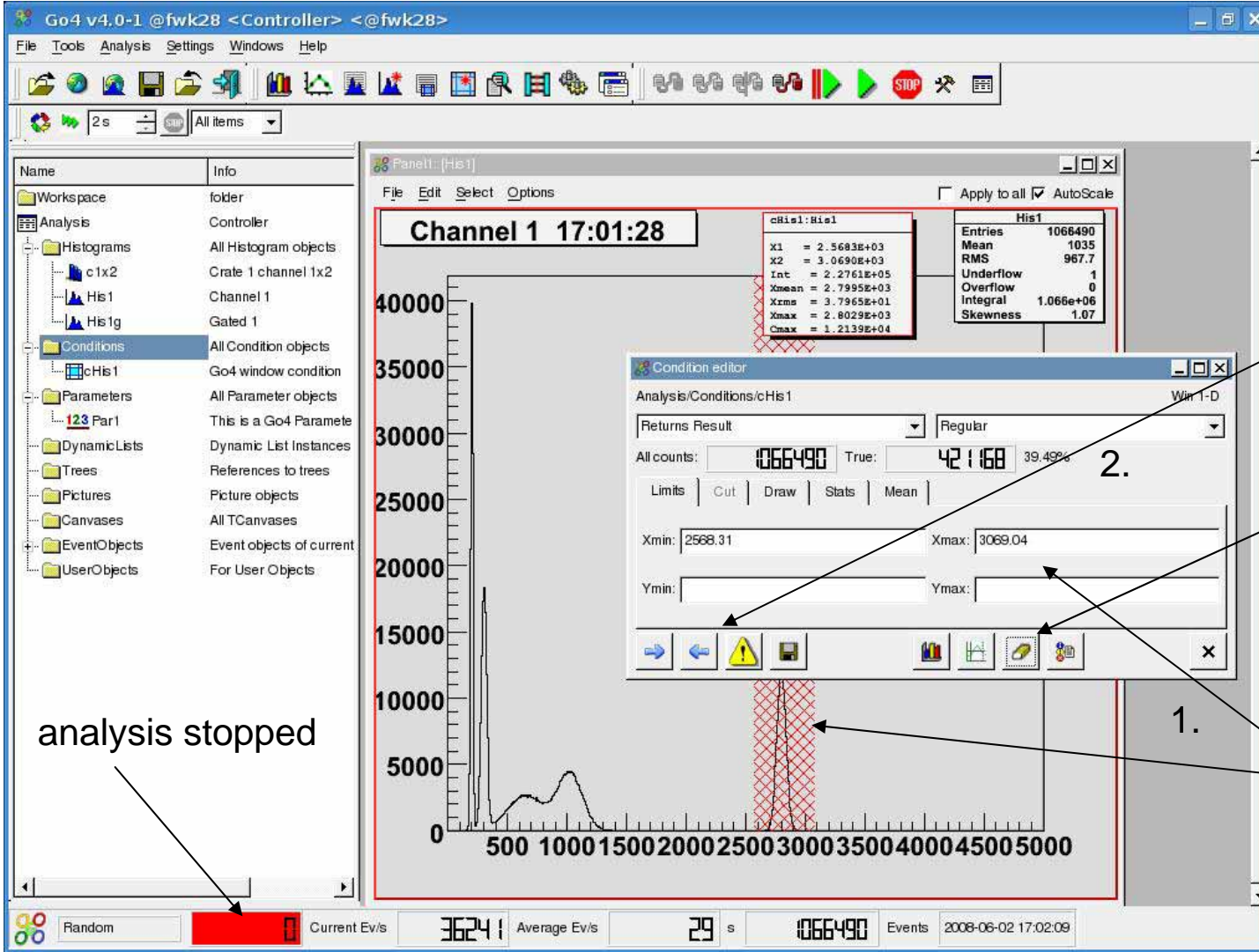
Condition editor			
Analysis/Conditions/cHis1		Win 1-D	
Returns Result	Regular		
All counts:	418263	True:	165400 39.54%
Limits	Cut	Draw	Stats
Xmin:	368.026	Xmax:	1369.5
Ymin:		Ymax:	

autoupdate  
browser content

conditions

analysis is  
running

# Condition example – step 2



analysis stopped

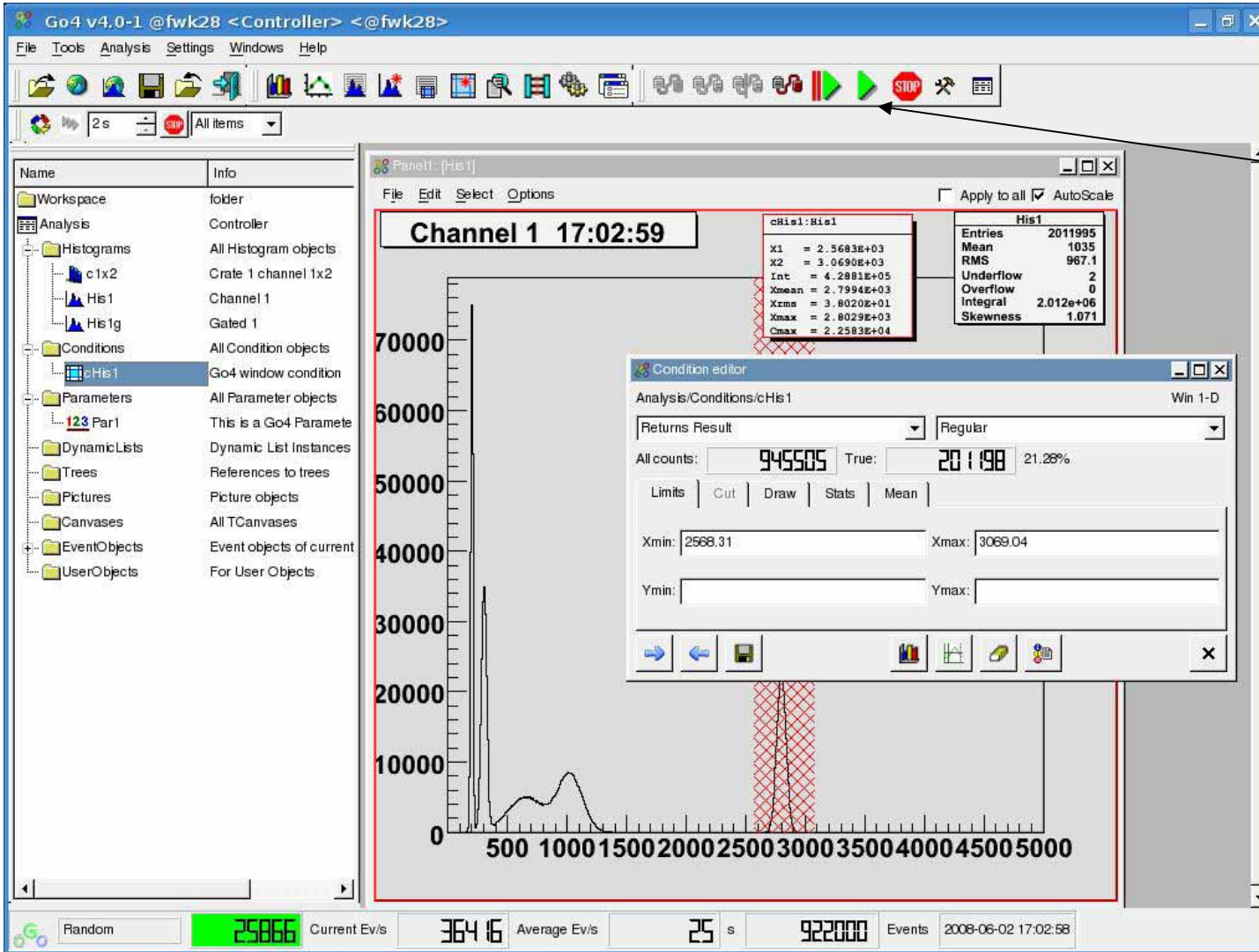
send values to analysis

reset counter

3.

change conditions  
 • drag & drop or  
 • enter values

# Condition example – step 1




restart analysis



# Tutorial

taken from:

<http://www-linux.gsi.de/~go4/tutorial/go4.php>



**Go4 Tutorials**

You need Go4 version 3.2 for the tutorials.

---

<a href="#">Entrance</a>	<a href="#">Go4</a>	Go4 web site
<a href="#">Login</a>	<a href="#">Lecture</a>	Overview Go4
<a href="#">Intern</a>	<a href="#">Analysis</a>	Overview Go4 analysis concepts
<a href="#">Documents</a>	<a href="#">Demos</a>	Some demos of workshop
<a href="#">Downloads</a>		

---

<a href="#">Tutorial 1</a>	<a href="#">Tutorial 1</a>	How to install Qt, ROOT and Go4
<a href="#">Tutorial 2</a>		<ol style="list-style-type: none"> <li>1. Install Qt</li> <li>2. Install ROOT</li> <li>3. Install Go4</li> </ol>
<a href="#">Tutorial 3</a>	<a href="#">Tutorial 2</a>	Running first very simple example <ul style="list-style-type: none"> <li>• Get tarball of example and build it</li> <li>• Run analysis from shell and look at the results with GUI</li> <li>• Run analysis by GUI</li> <li>• First usage of histogram monitoring, condition and parameter editors</li> </ul>
<a href="#">Tutorial 4</a>	<a href="#">Tutorial 3</a>	Looking into the code <ul style="list-style-type: none"> <li>• Look into the analysis setup file</li> <li>• Look into the analysis code</li> </ul>
<a href="#">Tutorial 5</a>	<a href="#">Tutorial 4</a>	Using an event object for output
<a href="#">Tutorial 6</a>		
<a href="#">Tutorial 7</a>		
<a href="#">Tutorial 8</a>		
<a href="#">Tutorial 9</a>		
<a href="#">Tutorial 10</a>		

# The End

## Aknowledgements:

- Dr. J. Adamczewski
- R. Beyer
- R. Hannaske
- Dr. M. Schlett
- Dr. A. Wagner