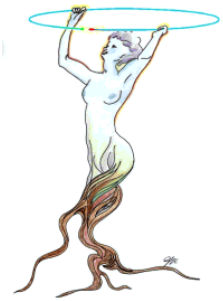
CDF Online Monitoring

T. Arisawa, K. Ikado, K. Maeshima, H. Stadie,
G. Veramendi, W. Wagner, H. Wenzel

- Goals
- Design
- Status

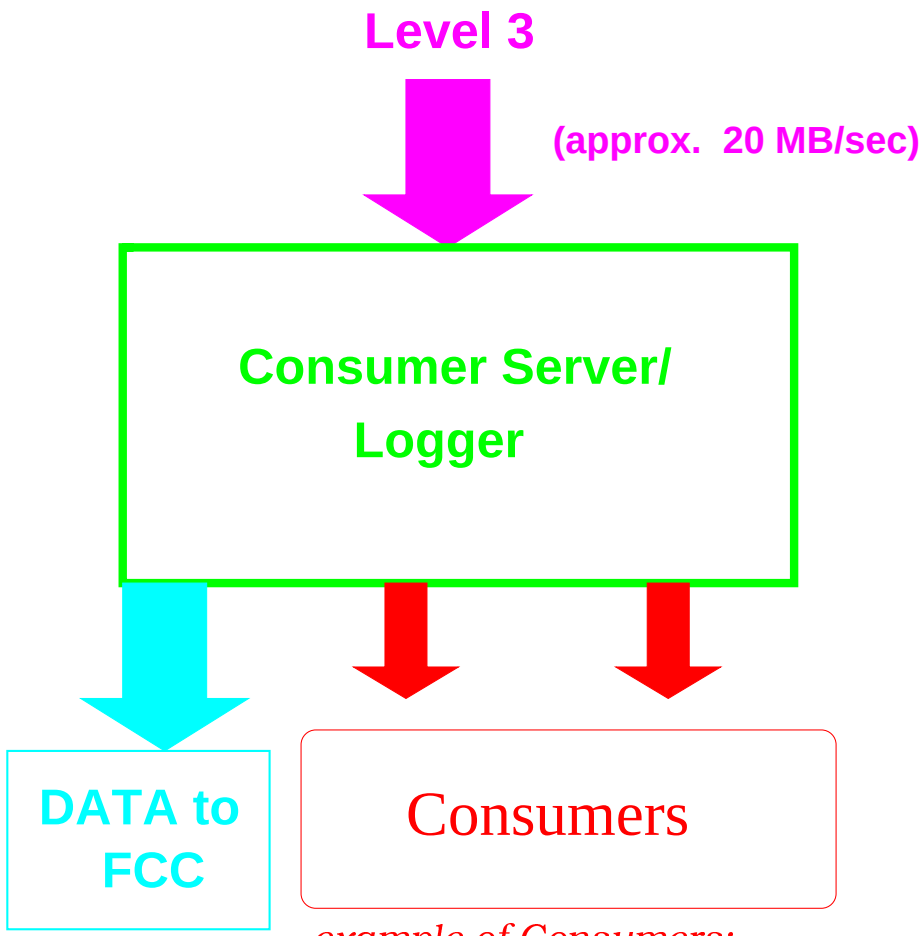


http://www-b0.fnal.gov:8000/consumer/home/consumer_home.html

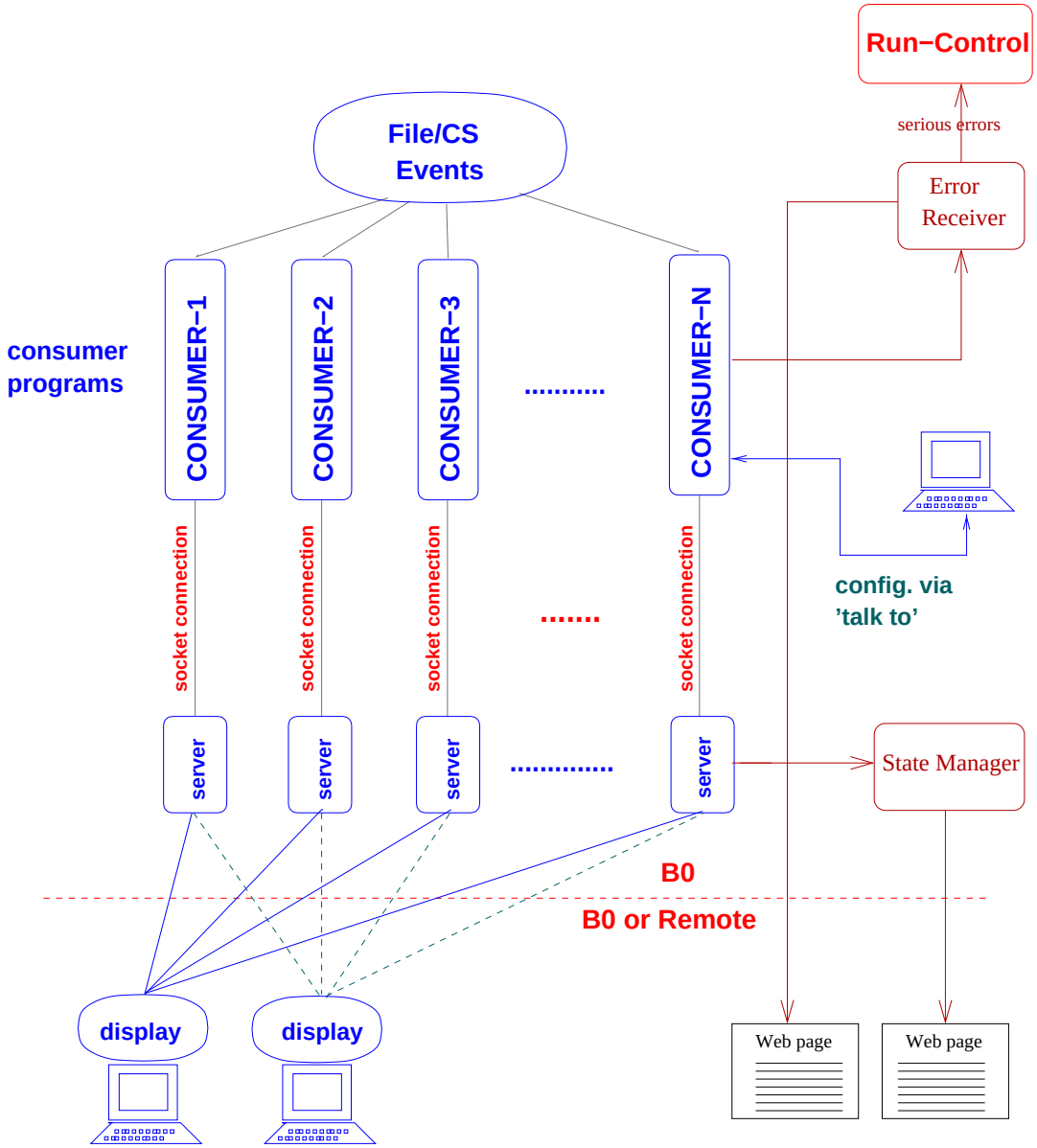
Desired features of Consumer-Monitoring Framework

- monitor the detector without interfering with the data taking
  make occupancy plots, watch trigger (rates), check reconstruction objects, etc.
- different consumer processes can run on different machines
- each consumer receives only the data it needs
- the monitoring and the display processes are separated
  The number of displays is only limited by network traffic and bandwidth
- different consumers can be combined to one executable
- common interface and maintainability

How do we get the data?



example of Consumers:
EventDisplay, YMon, Stage0, TrigMon, LumMon, etc..


Run−Control
serious errors
Error Receiver
File/CS Events
consumer programs
CONSUMER−1
CONSUMER−2
CONSUMER−3
.....
CONSUMER−N
socket connection
socket connection
socket connection
.....
socket connection
config. via 'talk to'
server
server
server
.....
server
State Manager
B0
B0 or Remote
display
display
Web page
Web page

I. The Consumers

- analyze and monitor the event data
- use CDF Run II offline framework to look at data
- consists of different AC++ modules:
  - APPConsumerInputModule
  - ConsumerErrorModule
    adds special destination to the ZOOM error logger to send errors to the ConsumerErrorReceiver
  - module inheriting from ConsumerFrameworkModule
    consists of different monitors that are written by the experts
    all these monitors inherit from BaseMonitor2
    starts server process at the beginning of a job

BaseMonitor2

- base class for all monitors
- functions to be overwritten by the monitor writer:
  initialize(), reset(), addEvent(EventRecord* event),
  checkMonitor(), finalFit(), summary(std::ostream &os=std::cout),
  store(TFile* f)
- "framework" functions:
  addObject(TObject *obj, const char *const path),
  addSlideshowObject(TObject *obj),
  addErrorObject(TObject *obj, const char *const text),
  createCanvas(const char *const name, const char *const title, int nx, int ny),
  removeObject(TObject *obj)

TConsumerInfo

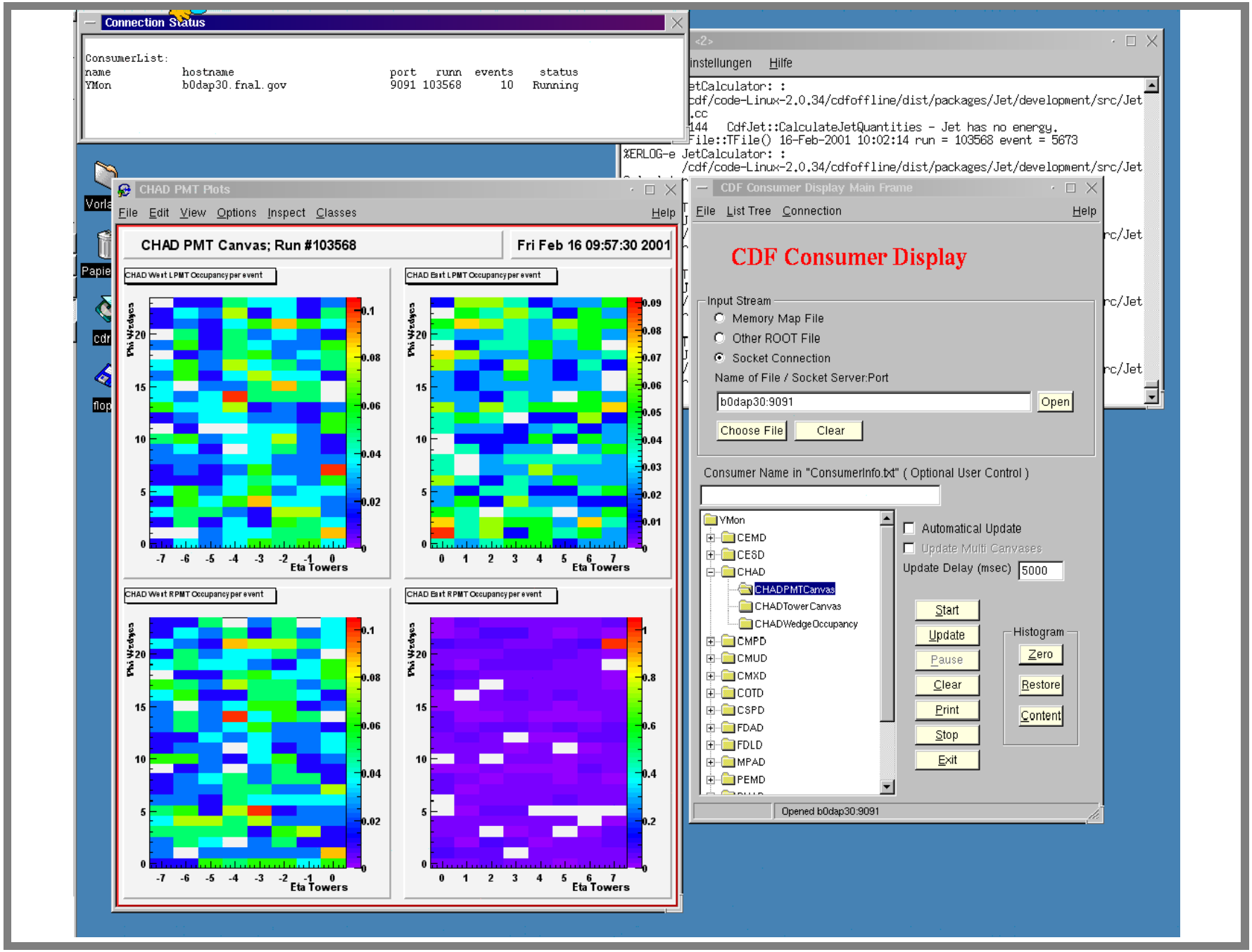
- is sent from the consumer to server
- is sent from the server to the display
- contains information about the consumer
  - name
  - run number
  - number of events processed
  - list of all ROOT objects that are available (at the moment)

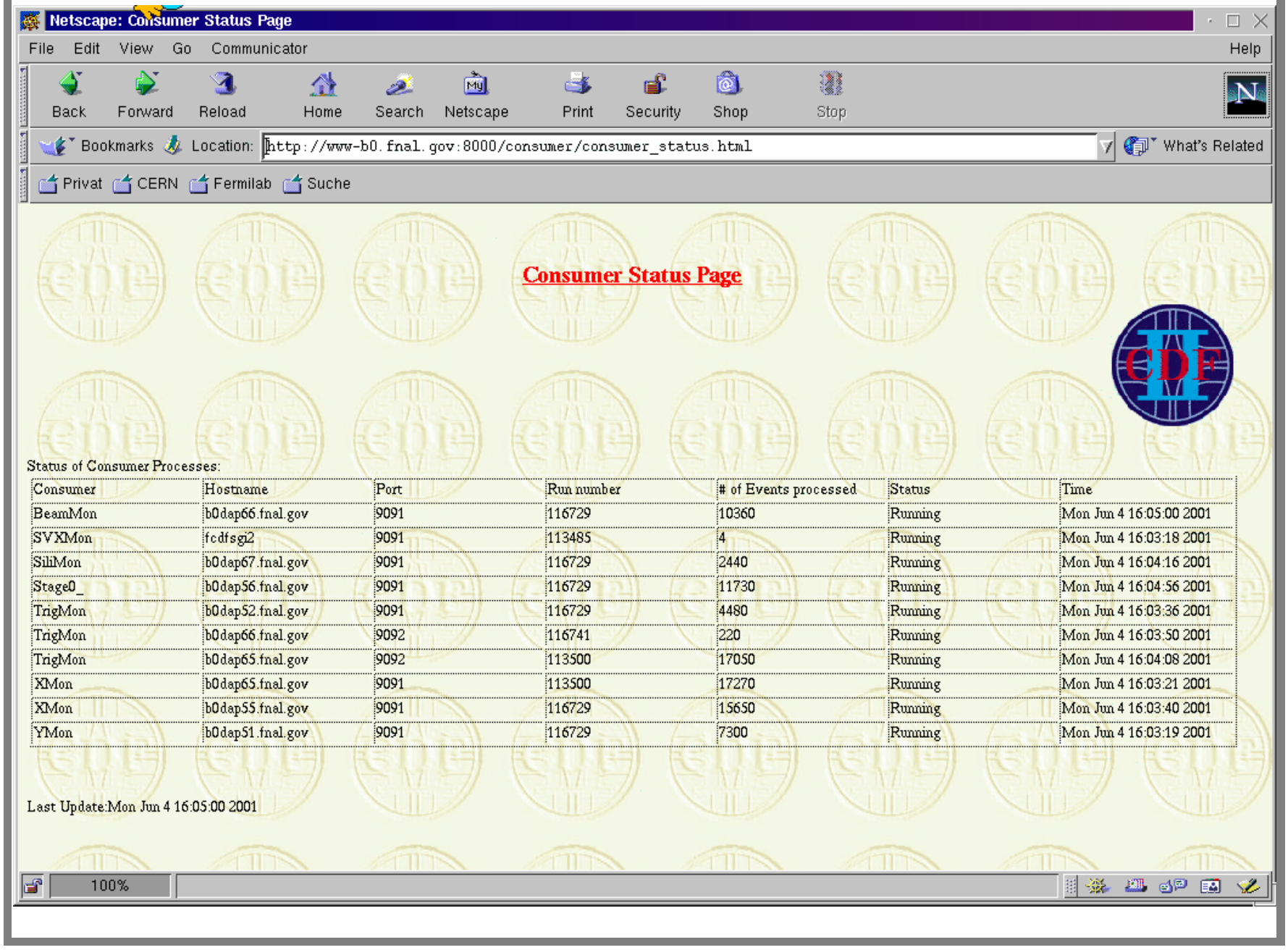
II. The Server

- receives ROOT objects from the consumer via socket
- deals with requests from the displays
- reports the status of the consumer to the state manager
- TCanvas: stores message, not the object

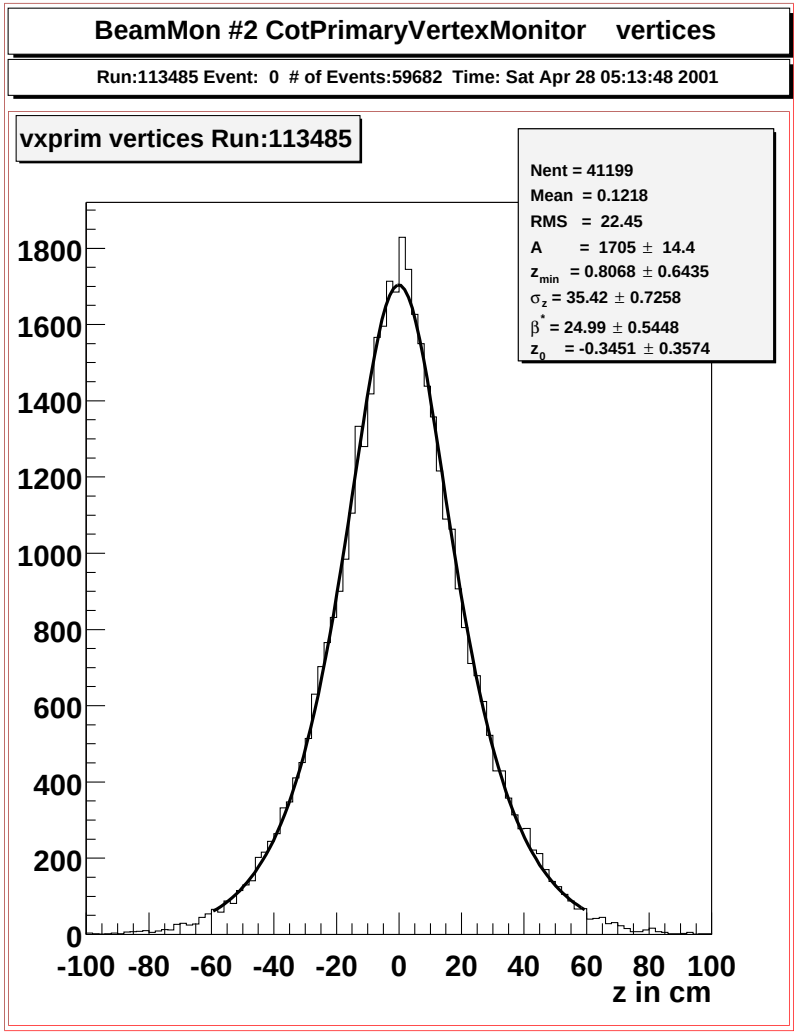
III. The Display

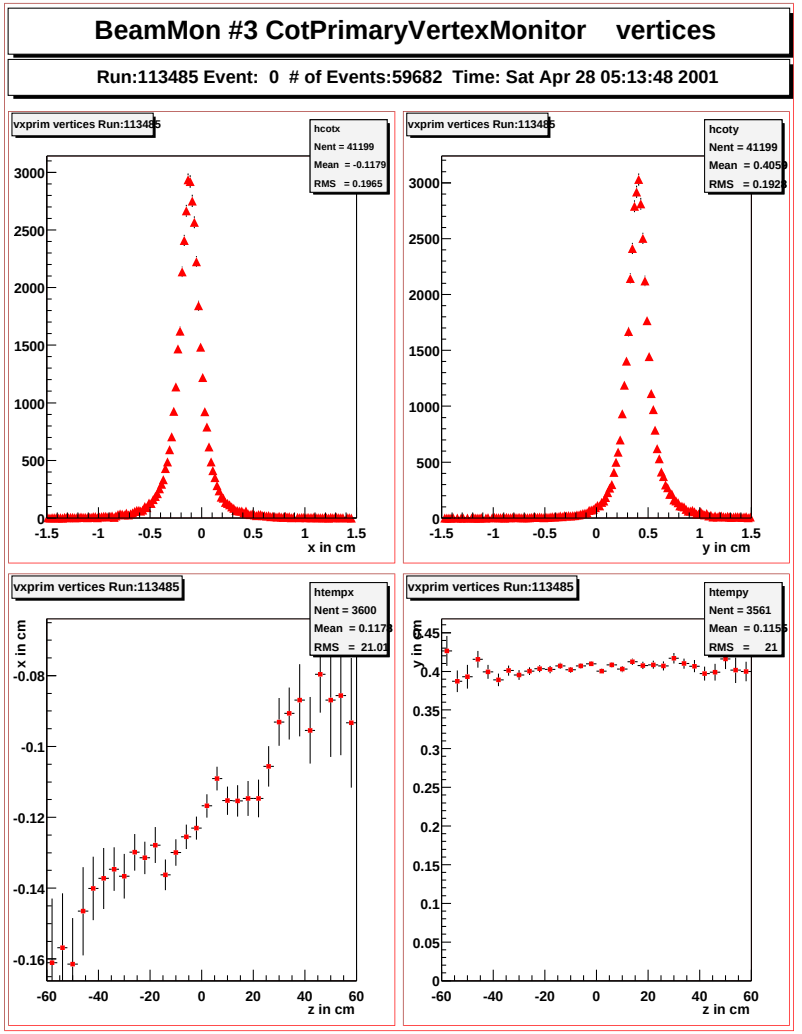
- ROOT-based GUI
- can connect to the server via socket, can browse TFile or TMapFile
- at first requests TConsumerInfo and creates a list tree
- only updates objects on canvas, does not redraw whole canvas


Connection Status
ConsumerList:
name hostname port runn events status
YMon b0dap30.fnal.gov 9091 103568 10 Running
CHAD PMT Plots
File Edit View Options Inspect Classes Help
CHAD PMT Canvas; Run #103568
Fri Feb 16 09:57:30 2001
CHAD West LPMT Occupancy per event
CHAD East LPMT Occupancy per event
CHAD West RPMT Occupancy per event
CHAD East RPMT Occupancy per event
Phi Wedges
Eta Towers
CDF Consumer Display Main Frame
File List Tree Connection Help
CDF Consumer Display
Input Stream
Memory Map File
Other ROOT File
Socket Connection
Name of File / Socket Server:Port
b0dap30:9091
Open
Choose File
Clear
Consumer Name in "ConsumerInfo.txt" (Optional User Control)
YMon
CEMD
CESD
CHAD
CHADPMTCanvas
CHADTowerCanvas
CHADWedgeOccupancy
CMPD
CMUD
CMXD
COTD
CSPD
FDAD
FDLD
MPAD
PEMD
Automatical Update
Update Multi Canvases
Update Delay (msec)
5000
Start
Update
Pause
Clear
Print
Stop
Exit
Histogram
Zero
Restore
Content
Opened b0dap30:9091
JetCalculator: :
/cdf/code-Linux-2.0.34/cdfoffline/dist/packages/Jet/development/src/Jet.cc
144 CdfJet::CalculateJetQuantities - Jet has no energy.
TFile::TFile() 16-Feb-2001 10:02:14 run = 103568 event = 5673


Netscape: Consumer Status Page
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Location: http://www-b0.fnal.gov:8000/consumer/consumer_status.html
What's Related
Privat CERN Fermilab Suche
Consumer Status Page
CDF
Status of Consumer Processes:
Consumer | Hostname | Port | Run number | # of Events processed | Status | Time
BeamMon | b0dap66.fnal.gov | 9091 | 116729 | 10360 | Running | Mon Jun 4 16:05:00 2001
SVXMon | fcdfsgi2 | 9091 | 113485 | 4 | Running | Mon Jun 4 16:03:18 2001
SiliMon | b0dap67.fnal.gov | 9091 | 116729 | 2440 | Running | Mon Jun 4 16:04:16 2001
Stage0_ | b0dap56.fnal.gov | 9091 | 116729 | 11730 | Running | Mon Jun 4 16:04:56 2001
TrigMon | b0dap52.fnal.gov | 9091 | 116729 | 4480 | Running | Mon Jun 4 16:03:36 2001
TrigMon | b0dap66.fnal.gov | 9092 | 116741 | 220 | Running | Mon Jun 4 16:03:50 2001
TrigMon | b0dap65.fnal.gov | 9092 | 113500 | 17050 | Running | Mon Jun 4 16:04:08 2001
XMon | b0dap65.fnal.gov | 9091 | 113500 | 17270 | Running | Mon Jun 4 16:03:21 2001
XMon | b0dap55.fnal.gov | 9091 | 116729 | 15650 | Running | Mon Jun 4 16:03:40 2001
YMon | b0dap51.fnal.gov | 9091 | 116729 | 7300 | Running | Mon Jun 4 16:03:19 2001
Last Update:Mon Jun 4 16:05:00 2001
100%

Example: BeamMon





Status

- first full scale test in CDF commissioning runs last fall with four consumers
- framework implementation restructured during winter/spring
- Run II started ; framework in place ; eight consumers ; $\sim$ 20 monitor writers
- still adding more consumers and enhancing others during the commissioning of the detector
- cdfsoft-independent version of consumer framework available

Comments & Problems with ROOT 2.26

- nice and easy to send objects via socket
- a lot of problems with TMapFile → do not use it any more
- scope rules with files , canvas
- problems with overdrawing contents on a canvas ; canvas design
- batch mode , gif-files (TCanvas::Print)

Acknowledgements

We thank all the people who are contributing to this work. Special thanks to the ROOT team, the CDF online group, and the Fermilab Computing Division. We also thank the community of ROOT users who via the roottalk mailing list provide answers and solutions to many problems.