Beam Line Monitor DAQ
Manual ver. 1.3

M.Kohama

September 8, 1999

Contents

1 Component

1.1 Stations

Monitor	modules	data	N-HL	AWA.	T-St.	Muon	total	size
SPILL	CAM COINS.	1 W	1	1	3	1	6	288B
GPS	VME moudle	20 W	1	0	0	0	1	168B
CT	CAM CH-ADC	1 W	5	3	6	0	14	448B
SEC	CAM CH-ADC	1 W	6	0	0	0	6	288B
SPIC	F-BUS ADC	64 W	12	8	8	0	28	8.7KB
SLOW	CAM SC-ADC	32 W	0	0	2	0	2	354B
HORN	VME Module	400 W	0	0	8	0	8	13KB
P/PI	F-BUS ADC	352 W	0	0	1	0	1	1.5kB
PI	CAM CH-ADC	32 W	0	0	1	0	1	172B
IO	CAM CH-ADC	60 W	0	0	1	0	1	284B
MUON	TKO CH-ADC	96 W	0	0	0	1	1	412B
Total			25	12	30	2	69	25.6kB

Table 1: Location of Monitors

There are 4 daq stations on the beam line to assemble beam monitor data. At North Counter Hall, G.P.S. time measuring system and Proton beam monitors (CTs,SECs,SPICs) are in place. The beam timing (-1.1msec) comes from accelerator, and is distributed to each station(-120 micro sec). Another beam timing (-20msec) go through to Target station. For tagging each station's data, spill number is constructed and distributed to each station by VME TRG module. The DAQ is a combination of CAMAC, fastbus and VME systems.

At AWABAKO station, some beam monitors(CTs,SPICs) are placed. The trigger timing(-120 micro sec) comes from North Counter Hall. The DAQ is a combination of CAMAC and fastbus systems.

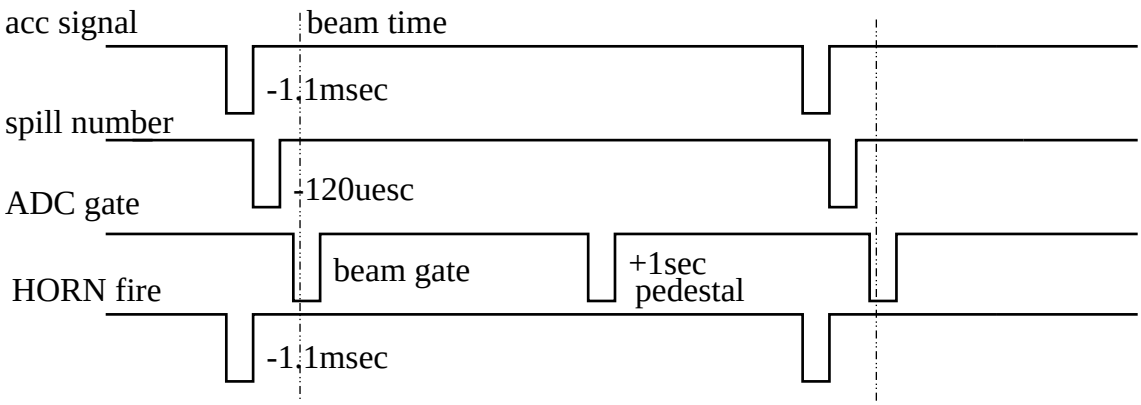
At the Target Station, Proton beam monitors(CT,SEC,SPIC),HORN current monitor, PI cherenkov monitor, Ionocopter and P/PI Profile monitor are in place.Two trigger timing signal (-1.1 msec, -20 msec) come from North Counter Hall, the -1.1 msec signal triggers beam monitors and HORN fire, and the -20 msec signal stops the HORN capacitor charging. The DAQ is a combinatin of CAMAC,fastbus and VME systems. PI cherenkov moni-

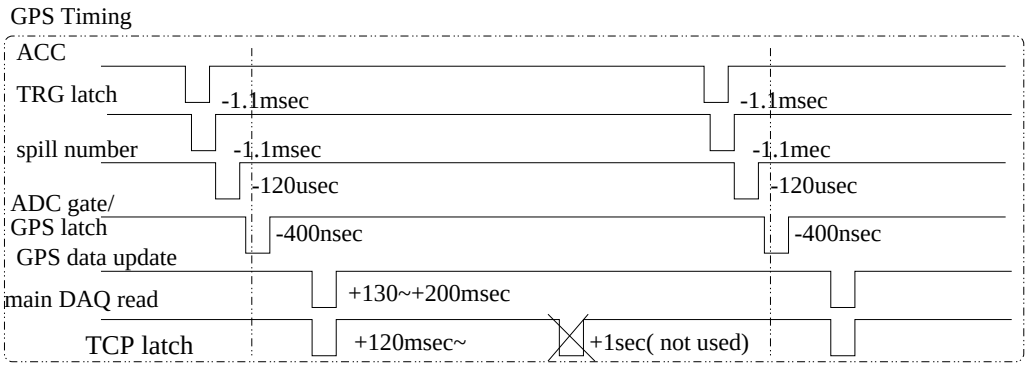
tor and Ionocopter is independent DAQ system, so there are three DAQ in Target Station. DAQ monitor system is also placed, this is different DAQ system with main DAQ. This system read GPS file which made by North Counter Hall DAQ, and Target monitors(CT,SEC).

At the Muon monitor room, Muon monitor is placed The trigger timing(-120 micro sec) comes from North Counter Hall.
DAQ is TKO system.

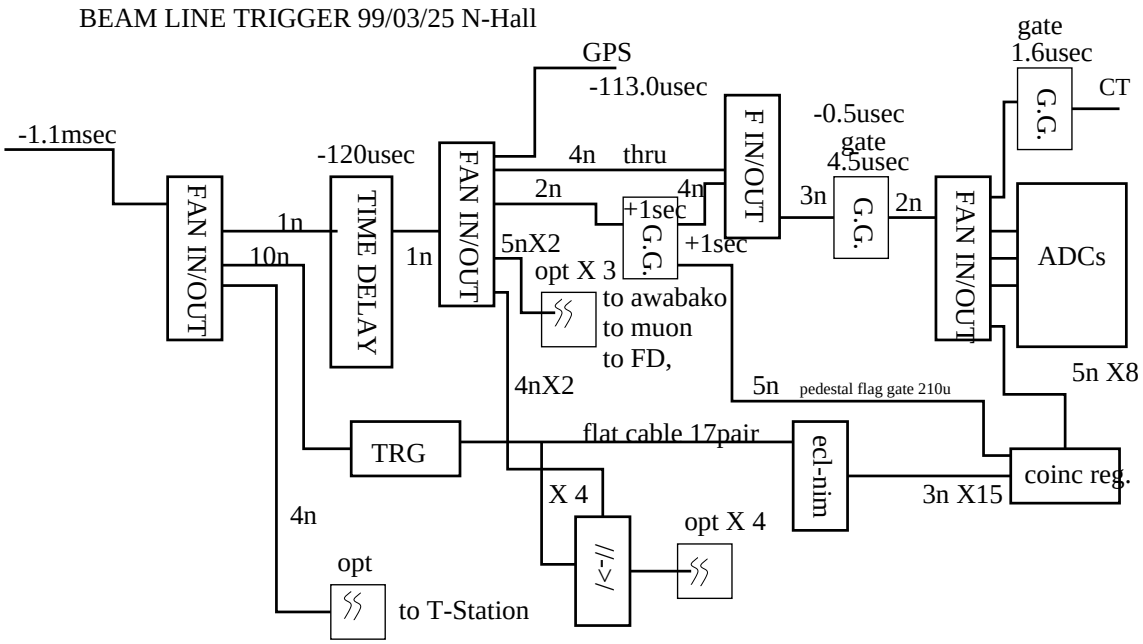
2 trigger logic

2.1 timing

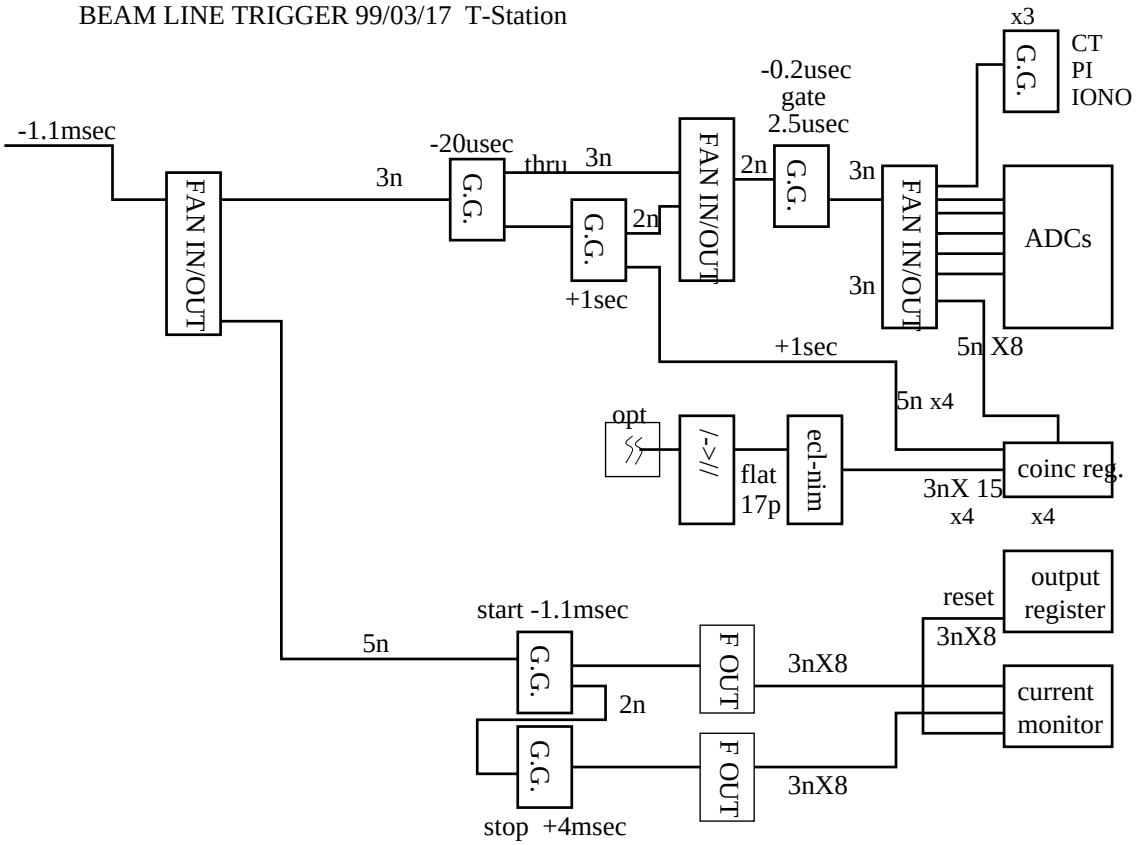




2.2 North Counter Hall trigger

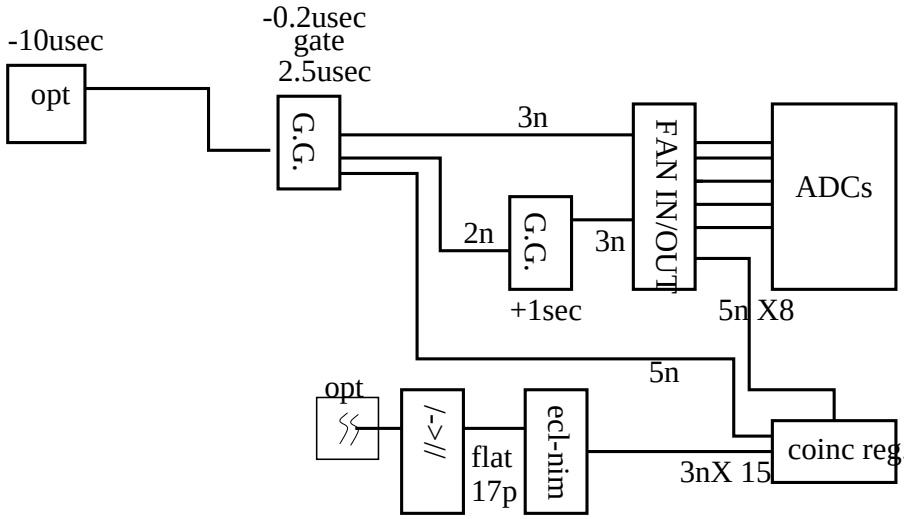


2.3 Target Station trigger



2.4 AWABAKO/Muon station trigger

BEAM LINE TRIGGER 99/01/14 MUON/AWABAKO



3 SoftWare

3.1 System environment

All machine are PC-linux.This is typical spec.

CPU	Pentium class	more than 133Mhz
Memory	64MB	HOST,EV server need 128MB
HD	2GB	HOST need more than 4GB
OS	Linux 2.0.35	Slackware 3.4 or later
JAVA	jdk 1.1.7	
HORB	horb 1.3	http://ring.etl.go.jp/openlab/horb/
lib	libc5, libc6	libc5.4.38 or later

Table 2: Spec

Each machine current spec is shown.

Machine	CPU	Memory	HDD	proceses
nubl01	P-133Mhz	96MB	6GB	gpstrg
nubl02	P-233Mhz	64MB	4GB	Client2(SPIC,CT)
nubl03	P-233Mhz	64MB	4GB	Client3(SPIC,CT,SEC,HORN)
nubl04	P2-266Mhz	64MB	4GB	Client4(PI)
nubl05	?	?	?	Client5(IONO)
nubl06	P-133Mhz	64MB	2GB	Client6(MUON)
nubl07	P2-400Mhz	128MB	10GB	Reform,Server(HOST machine)
nubl08	C-333Mhz	256MB	4GB	EvBank(event display sever)
nubl09	P-200Mhz	128MB	6GB	DAQ monitor
nubl10	P-166Mhz	32MB	2GB	Event display
nubl11	P-233Mhz	128MB	4GB	Client1(SPIC,CT,SEC)
nubl12	P3-500Mhz	256MB	40GB	Data server

Table 3: Each machine Spec

3.2 DAQ system

Station	Machine	monitors
N-Hall	nubl01	GPS,TRG
N-Hall	nubl11	SPILL,CT,SEC,SPIC
AWABAKO	nubl02	SPILL,CT,SEC,SPIC
T-station	nubl03	SPILL,CT,SEC,SPIC,SLOW,HORN,P/PI
T-station	nubl04	SPILL,PI cherenkov
T-station	nubl05	SPILL,Ionocopter
Muon room	nubl06	SPILL,muon monitor
T-station	nubl07	HOST machine
N-Hall	nubl08	Event display Server machine
N-Hall	nubl12	Data Server machine
T-station	nubl09	DAQ monitor machine
T-station	nubl10	Event Display machine

Table 4: Component of front-end daq

The system is composed of 6 front-end machine, one host machine and one event display server machine. All computer are PC-Linux. At the front-end machines, read out programs are written in C, data send to event builder at host machine, and data transfer program is written in JAVA language. On the host machine, the event builder is running and saving output to local DISK. These data are copied to the CCPS machine which writes them to tape. Another process transfers the output to event display server which is running on the event display machine.

3.3 Data stream

The general data stream is shown in this figure. Each Client gets each station

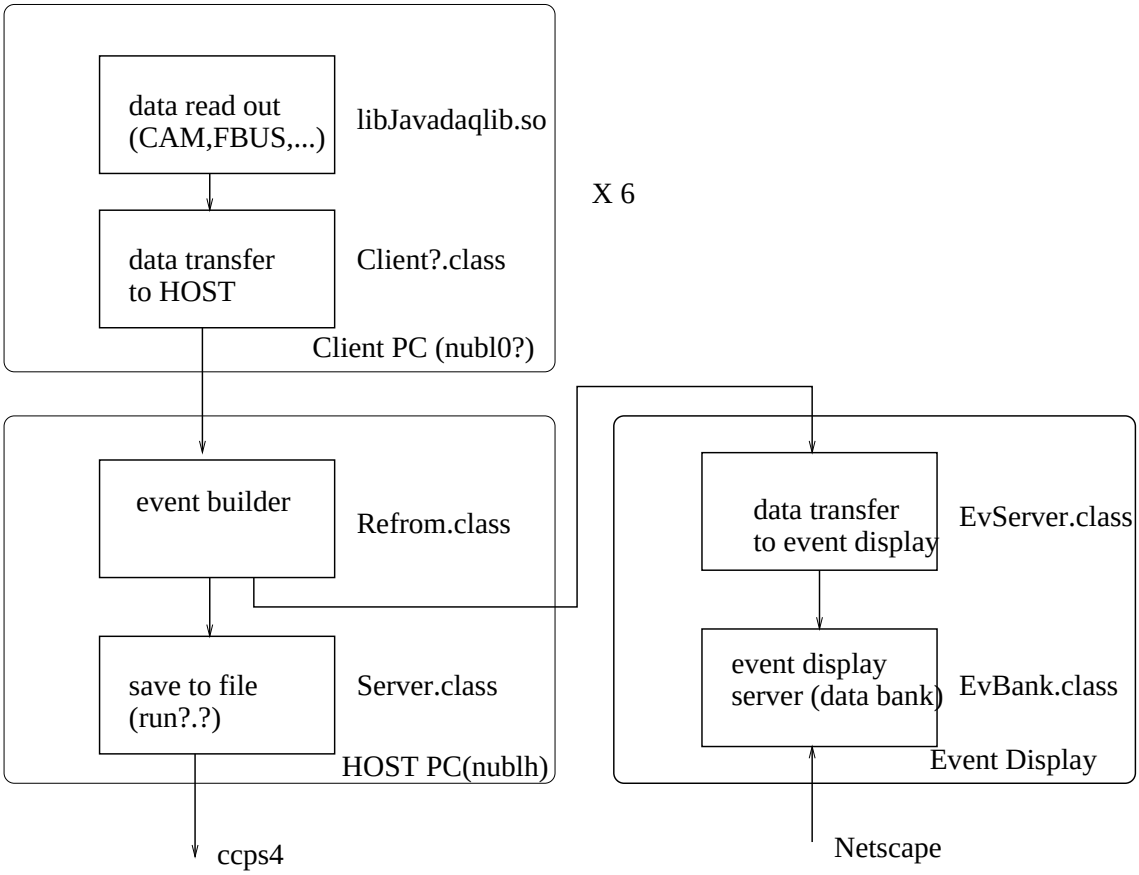


Figure 1: Data stream of Beam line DAQ

data, and sends data to Reform.class process on HOST. Reform.class has 50 buffers for each client data. When each data's spill number is matched, Server.class gets combined data from Reform.class and save it to file. Also at this time,EvServer.class sends spill data to EventDisplay Sever which process is EvBank.class. Event display gets spill data from EvBank.class every 2 sec.

3.4 data format

data file are now in ACSII file.The data format is like this.

1. RUN summary file
   This file will be made, when new sub run starts. At this component,

#	discription	type	comment
000	Length	int	total length of this file
001	RUN #	int	
002	Start event number	int	
003	End event number	int	
004	# of events	int	
005	Start time(sec)	int	
006	Start time(micro sec)	int	
007	End time(sec)	int	
008	End time(micro sec)	int	
009	Trig. type/Detector status	int	bit pattern.
010	Trig. type/Detector status	int	bit pattern.
011	Trig. type/Detector status	int	bit pattern.
012	Trig. type/Detector status	int	bit pattern.
013	Trig. type/Detector status	int	bit pattern.
014	Trig. type/Detector status	int	bit pattern.
015	Delimiter	int	(-length)

Table 5: Run header file

The time is cpu time.

2. event data
   This is event by event data-format of each monitor.
   last bit of Spill number is pedestal flag. If data is pedestal data, that bit is on.

```
#    event Spill     0-14 bit
#  pedestal flag      15 bit    (if pedestal, bit on)
```

Table 6: spill number

#	discription	type	comment
000	Length	int	
001	Data ID	int	(station No # Monitor ID # Monitor No)
002	MONITOR DATA ID	int	
003	Run #	int	
004	Spill #	int	
005	Event #	int	
006	Triger type	int	(pedestal:1 or data:0)
007	monitor Condition	int	(ON:1 or OFF:0)
008	CPU time(sec)	int	
009	CPUtime(micro sec)	int	
010			
:	detector data (ADC etc)		
:			
???			
???+1	Delimiter	int	(-length)

Table 7: event data format

3.5 RUN control

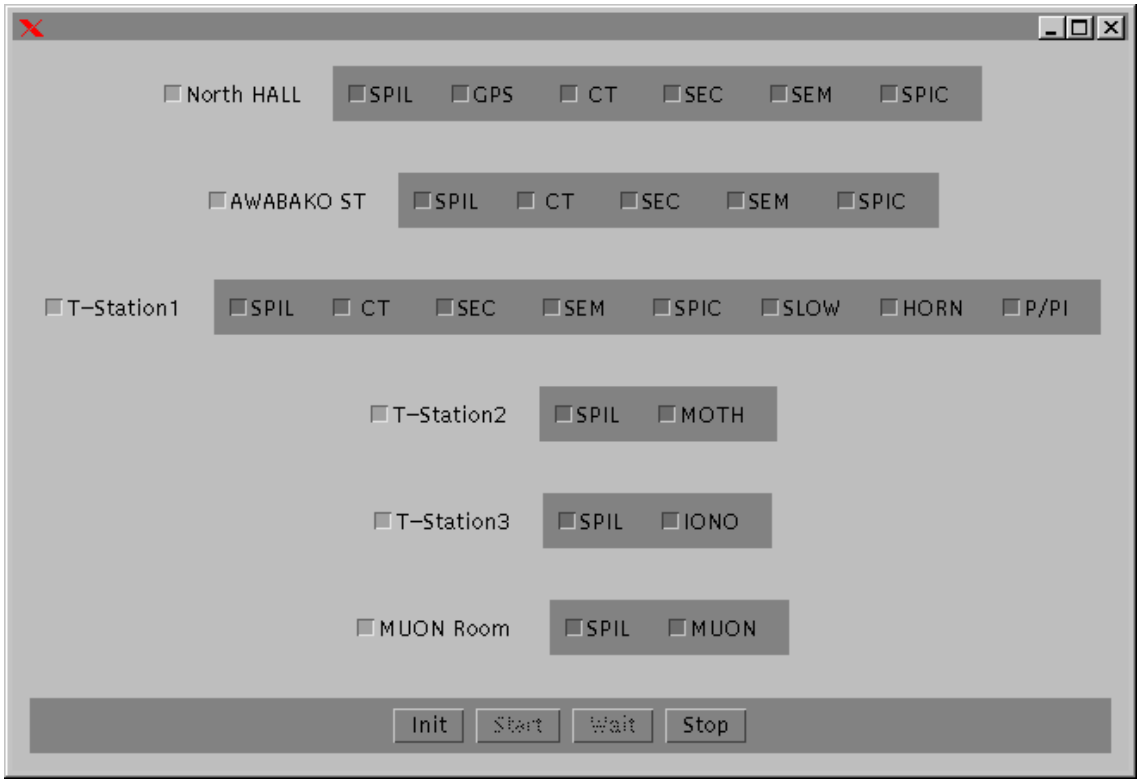


Figure 2: Run Control of Beam line DAQ

Run is controlled by the Run-Control program which name is Control.class. Control panel have theses buttons.

- Station ON/OFF
  This check box can enable / disable daq station.
- Monitor ON/OFF
  This check box can enable / disable each monitors.
- Init
  This button initialize of DAQ.

- Start
  This button start DAQ.
- Wait
  This button pause DAQ. When the Start button is pushed,run will start.
- Stop
  This button stop DAQ. All processes are ended.

3.5.1 How to start

1. Check each stations client process is running.
2. Chooce DAQ stations which you want to take data.
3. Chooce the monitor at each station which you want to take data.
4. Push the Start button.

3.5.2 How to change monitor condition

This is used for changing monitor condition.

1. Check each stations client process is running.
2. Chooce DAQ stations which you want to take data.
3. Chooce monitors at each station which you want to take data.
4. Push the Start button.

3.5.3 How to pause

1. Push the Wait button.
2. Push the Start button,when you want to re-start.

3.5.4 How to finish

1. Push the Stop button.

3.6 event display server

The Event display server is running on the event display server machine.
There are two proceses to provide event display data.

1. EvBank.class
   This is a daemon for event display, the event display processes access data through this program.

2. EvServer.class
   This program transfer event data from the HOST server to EvBank.class. This is also doing pedestal subtraction.

3.7 event display

The Event display works through Netscape.
The URL address is "http://nubl08.kek.jp/horb/evmonitor/" .
Another way to see the eventdisplay is that you install Java package and eventdisplay classes in your machine.
You can get the source file from "http://nubl08.kek.jp/horb/".

3.8 semi-offline process

The semi-offline process is running at nubl12. This process make ntuple file of beam data for each sub-run.
The file name is "run??????.??????.hbk" which is placed on "nubl12:/data2/hbk",and summary file(summary??????.??????) is placed on "nubl12:/data2/log" You can see datas using "sample.kumac" at "/home/online/bin ".

3.9 process list

program	machine	job
Client?.class	front-end machine (nubl0?)	get event data
Reform.class	host machine (nubl07)	event builder
Server.class	host machine (nubl07)	data file out
Control.class	host machine (nubl07)	run control
EvServer.class	event display server	data transfer
EvBank.class	event display server	event display data bank

Table 8: DAQ programs

4 Eventdisplay

Beam line Eventdisplay is written in Java,so Eventdisplay can work through Netscape and in standalone.

4.1 How to start at North Counter Hall(use appletviewer)

The Eventdisplay server(nubl08) is placed in North Counter Hall.

1. login nubl08 (account:shift,password shift99)
2. execute
   % evmonitor.sh
3. If you want to stop,choose "quit" in "applet" bar.

4.2 How to start using Netscape

Now unix and Windows95/98 are supported.

1. execute Netscape
2. access this URL; " http://nubl08.kek.jp/horb/evmonitor/BLmonitor.html"

4.3 How to watch beam monitor status

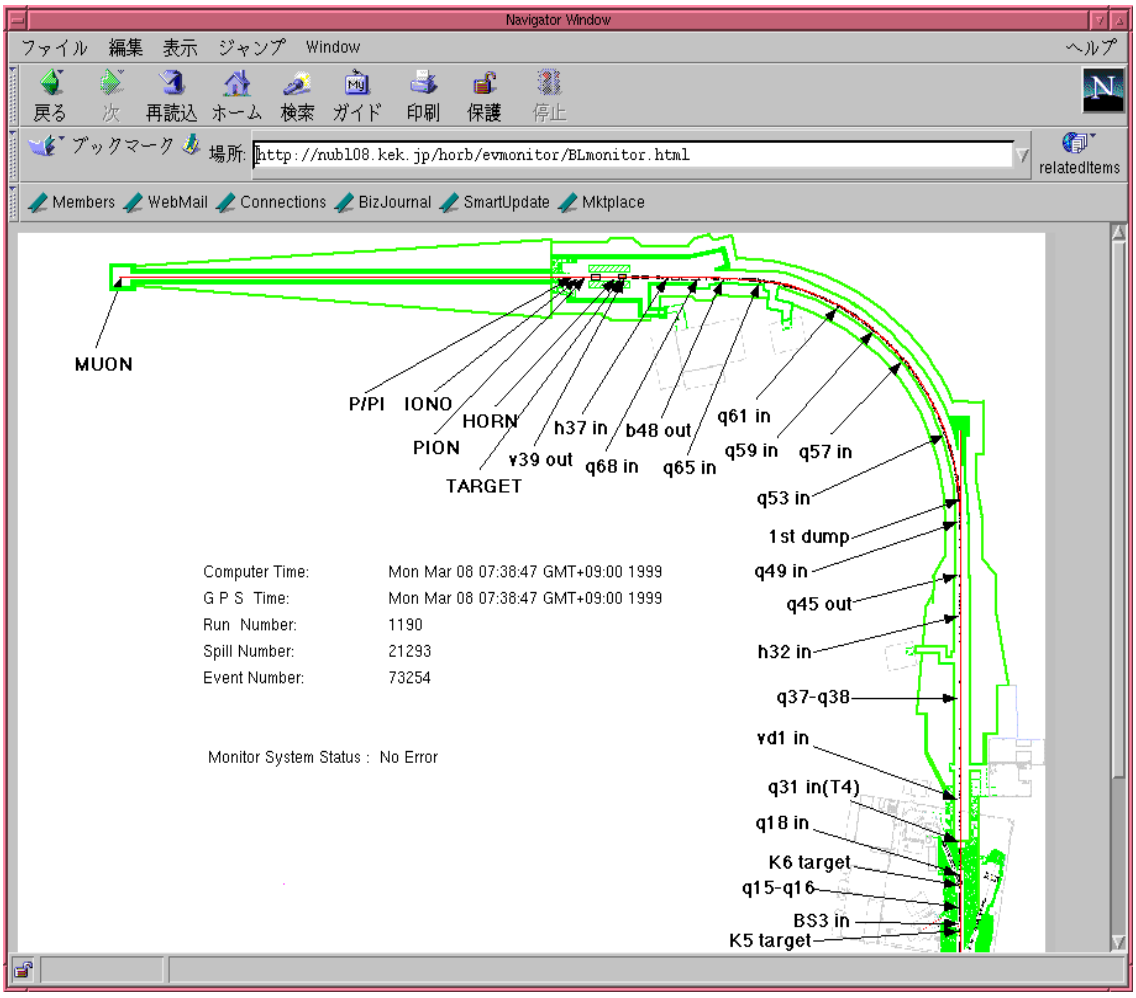
Click the ecah position name,then sub window pulls up.

4.4 How to print

On sub window, click the "function" at top bar, then printer window pulls up. After fixing the printer name and size, push the "print" button.

4.5 trouble shoooting

If Eventdisplay can not work on your unix machine, do
% unsetenv CLASSPATH



Eventdisplay on Netscape

Figure 3: Beam line Eventdisplay

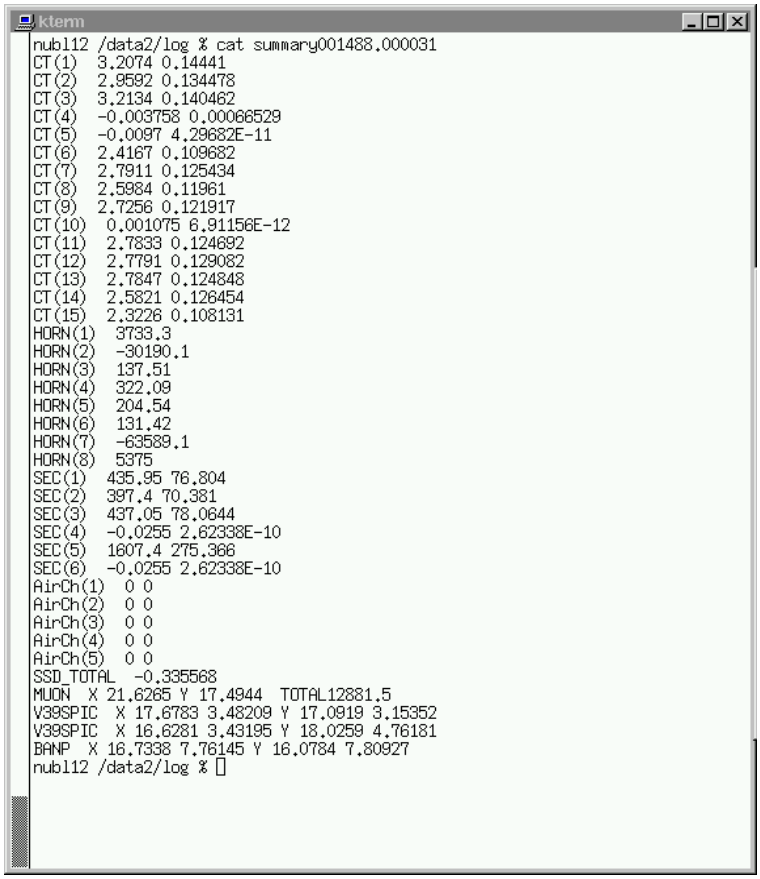
5 semi-Offline process

The semi-offline processes are working at "nubl12". There are two step for offline job.

1. fill hbook
   This real command is "readout" which is placed in /home/online/bin/
   .
   % readout [filename]
   output the "test.hbk" file which is ntuple format data. And "/home/online/bin/read.sh" is more usefule program. This program make run??????.?????.hbk file on /data2/hbk

2. make summary file
   This is kumac program, so this program is working at PAW window.
   "offrun.kumac " is semi-auto process on PAW window.
   "ofl.kumac " is anaylsis process,which makes summary.tmp file.
   These run summary files are placed on "nubl12:/data2/log/".

"sample.kumac" is a sample analysis program. You can see it at "nubl12@/home/online/bin".



Sample of run summary

Figure 4: SEMI-OFFLINE OUTPUT

6 Execution method

This section is only for experts.

6.1 How to auto-start

This is auto-start method.When run is not stable,please do next subsection way.

1. login HOST machine.(machine: nubl07, account:online, password:gamaoil1)
2. change directory to /data/daq.
3. execute
   % daqstart.sh
   The default setting is enable to client1, client2, client3 and client6.
4. Choose the DAQ stations and monitors to take data.
5. push Start button on Run-Control panel.

6.2 How to start

1. Login HOST machine.(machine: nubl07, account:online, password:gamaoil1)
   - change directory to /data/daq.
   - execute
     % java Reform
2. At anothor window,login HOST machine as online account.
   - change directory to /data/daq.
   - execute
     % java Control
3. login each Clinet machine wchih you want to take data.
   (machine: nubl0?, account:online, password:gamaoil1)

- check the daq process.
  you can find daq process pid from "/home/online/client?.pid" file.
  cat client?.pid

- If daq process is stoped,execute
  % daq.sh

4. At anothor window,login HOST machine as online account.
   - change directory to /data/daq.
   - execute
     % java Server

5. Choose the DAQ stations and monitors to take data.
6. push Start button on Run-Control panel.

6.3 How to re-start or change daq-client configuration

1. login each Clinet machine.
   (machine: nubl0?, account:online, password:gamaoil1)
   - check the daq process.
     you can find daq process pid from "/home/online/client?.pid" file.
     cat client?.pid
     % ps aux | grep #pid
   - If process is stoped,
     execute
     % /data/daq/daq.sh
2. Choose the DAQ stations and monitors to take data.
3. push Start button on Run-Control panel.

6.4 How to stop

1. Push Stop button.
2. kill Reform.class process.
   push ctrl-C on Reform.class terminal.

6.5 trouble shooting

1. Push the “Init”button, then push the “start” button
2. If error still happens, push the “stop” button
3. input ctrl-c at the “Reform” process.
4. If client process is still running, kill this process at each client machine.

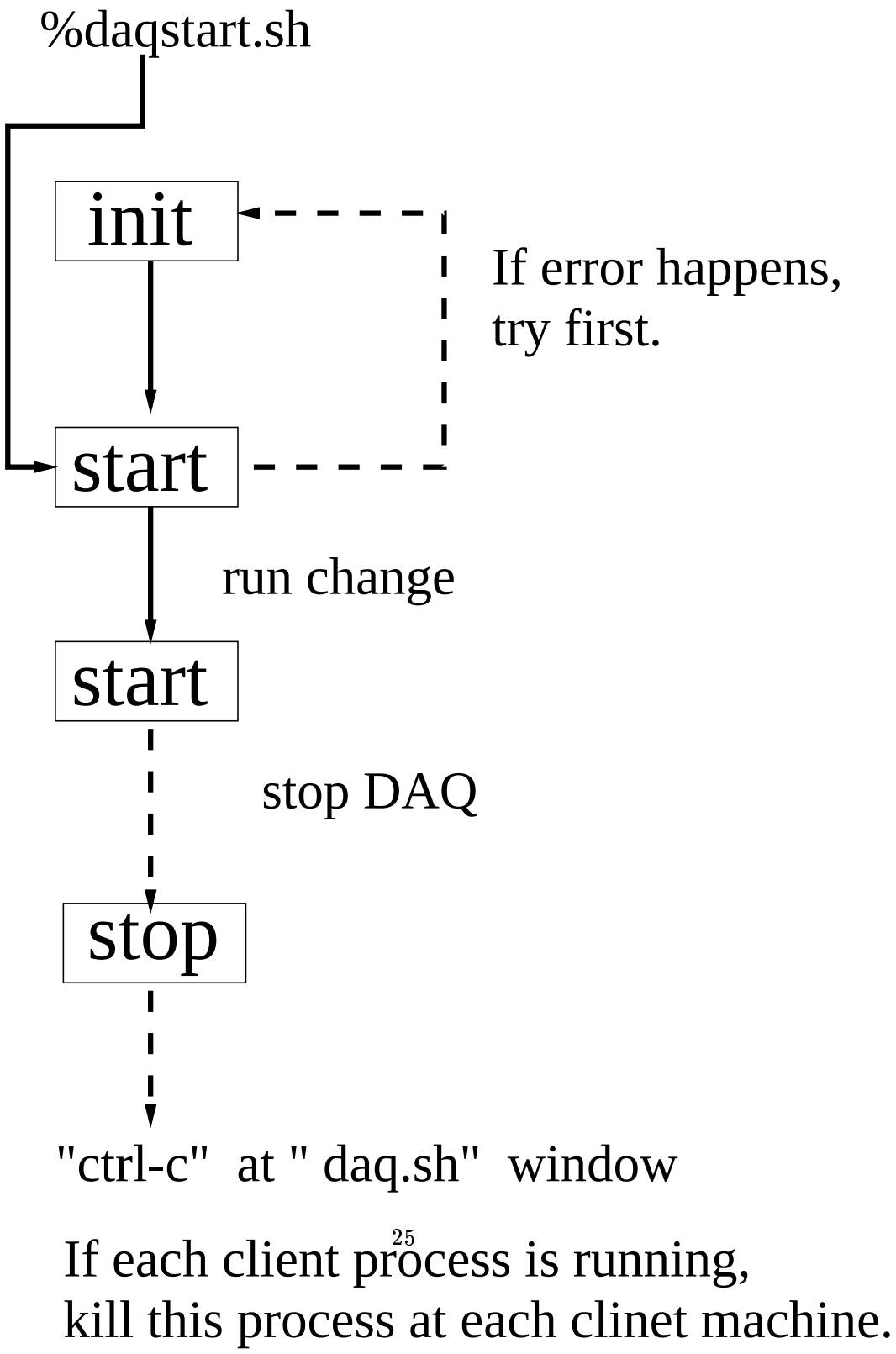


Figure 5: Normal state of main DAQ processes

7 Shift job

Mainly shift work is;
Check "pmon.sh"!

When daq is in its normal state, the shift person should check the process monitor shell (pmon.sh). If some process is stoped, pmon.sh will be beeping. If pmon.sh is stopped, execute
%pmon.sh
at nubl12:/data/daq.

Shift person should check theses processes every one hour at the North Counter Hall, and if trouble happens, each process will be beeping.

- Client process(JAVA program, on nubl01 - nubl06)
  This is collector.The process name is Client?.class (? = 1 - 6). When this process is stoped, Reform process says error.
- Reform process(JAVA program, on nubl07)
  This is main process of event building. The process name is Reform.class. You can find PID in "nubl07:/data/daq/reform.pid".
  When error is happened, it is beeping.
- Server process(JAVA program, on nubl07)
  This is the data storage process of main DAQ. The process name is Server.class. You can find its PID in "nubl07:/data/daq/server.pid".
  If error happenes, it is beeping.
- gps data sending process (shell, on nubl07)
  The process name is rcpicrr5.sh,this process is running on nubl07. Please execute this command on nubl07.
  % ps aux | grep rcpicrr5.sh
- data transfer process (shell, on nubl12)
  This is copy process,this process is running on nubl12. The process name is getdata.sh.Please execute this command on nubl12.
  % ps aux | grep getdata.sh
- data zipping process (shell, on nubl12)
  This is data zipping process at nubl12. The process name is check-

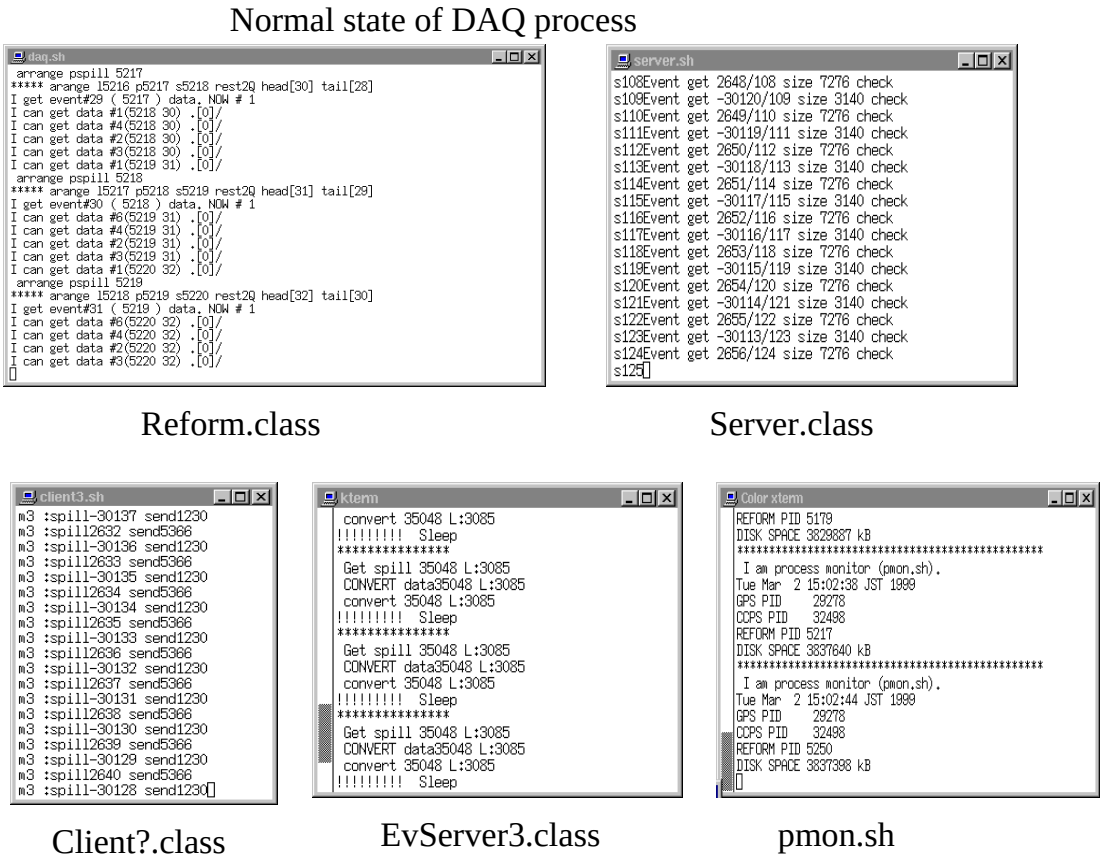


Figure 6: Normal state of main DAQ processes

dir.sh.Please execute this command on nubl12.
% ps aux | grep checkdir.sh

- Eventdisplay server process (JAVA program, on nubl08)
  This is data transfer program from DAQ-HOST to Eventdisplay-HOST.
  The process name is EvServer3.class.
  You can find PID in "nubl08:/home/online/disp/evserver.pid".

- Eventdisplay data bank process (JAVA program, on nubl08)
  This is Eventdisplay data bank, each Eventdisplay get data from this process. The process name is EvBank.class .

- DAQ monitor process (shell, on nubl09)
  This is sub DAQ system.This system has 3 processes.
  One is DAQ process, which name is start.sh.
  Next is spill by spill data transfer shell, which name is rcpspill.sh.
  Last is 3000 spill data transfer shell, which name is rcpbeam.sh.
  For checking theses processes,please execute commands on nubl09.
  % ps aux | grep start.sh
  % ps aux | grep rcpspill.sh
  % ps aux | grep rcpbeam.sh

- ccps tape storage process (shell, on ccps4)
  This is worked on kohama@ccps4.kek.jp, process name is "checkday.sh".
  For checking this process, please execute this command on ccps4.
  % ps aux | grep checkday.sh

When trouble happens, call expert.

8 expert list

We need more experts,please join.

- M.Kohama #5388 or 090-122-55788

9 software package distribution list

- Linux(kernel,packages,etc) " http://www.linux.org"
- Java for Linux "http://www.blackdown.org/java-linux.html"
- HORB "http://ring.etl.go.jp/openlab/horb/"
- vmelib for Linux "http://onlax2.kek.jp/~nakayosi/"
- ccp driver for Linux "http://neutrino.kek.jp/~kohama/ccp/ccp.html"
- this DAQ program "(http://neutrino.kek.jp/~kohama/daq/)"
- this DAQ program "(http://nublxx.kek.jp/~online/java/test3/)"

Beam line datas are placed in such DISK.

HOST DAQ	nubl07:/data/daq	buffer 10days
OFFLILE	nubl12:/data	raw data 15GB
	nubl12:/data2	hbook data 15GB
CCPS DISK	ccps4:/.../ccps.kek.jp/fs/v/4c2/k2k/beam	buffer 3days
CCPS MT(ALL DATA)	ccps1	use "kenzo" account
GPS low data	nubl01:/data/gps	buffer 6GB
GPS beam data	nubl07:/home/online/gpslog	buffer 10GB
DAQ monitor	nubl09:/home/online/data	buffer 6GB

Table 9: DISK palce

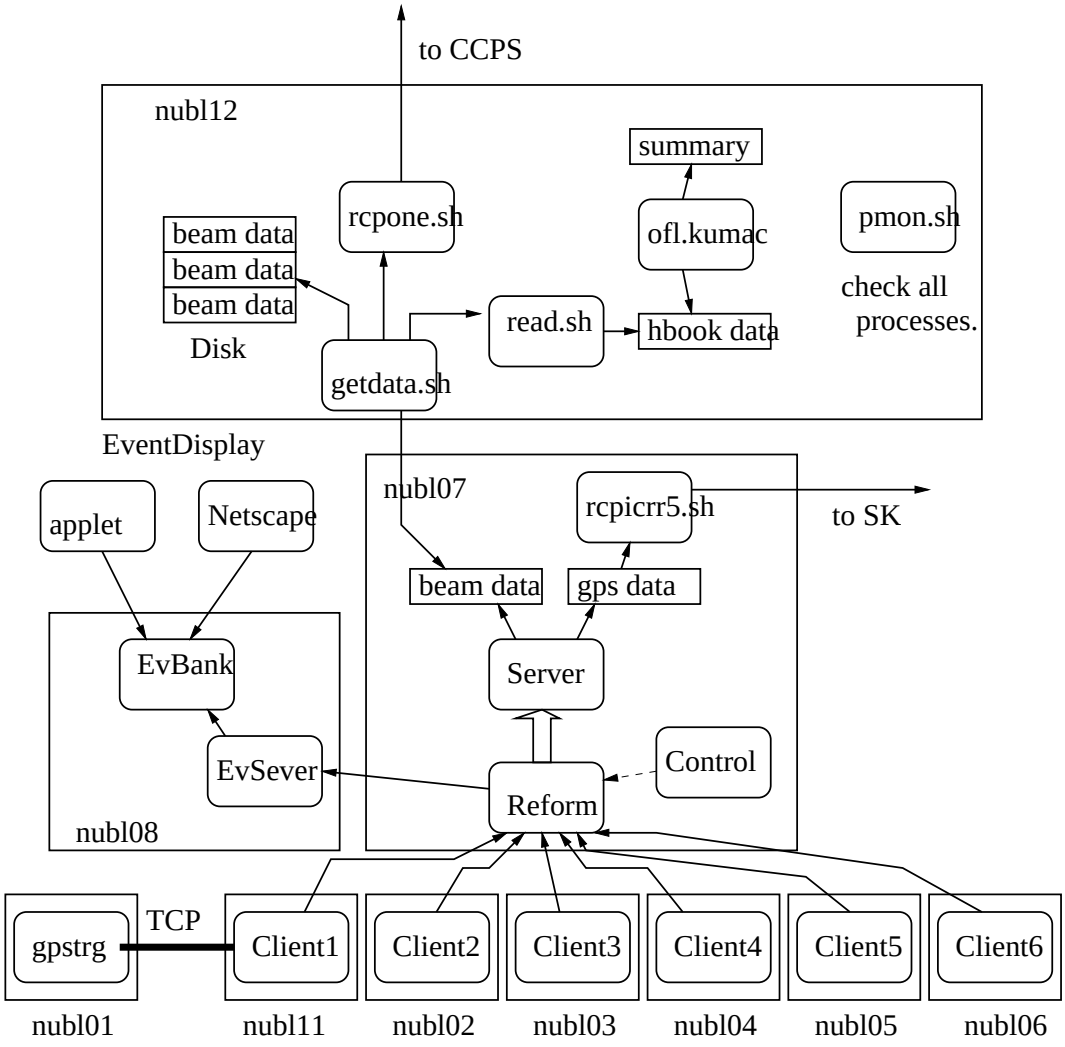


Figure 7: main data stream