CDFVME thoughts

S. Nahn, C. Gay, J. Yu - Yale
S. Bailey, Petar - Harvard
M. Votava, Y. Guo - CD
J Patrick, S. Vecjik- Fermilab
*On*line/$\emptyset$~~ff~~/*On*week DAQ mtg
December 18, 1998

Idea - try to clarify common questions . . .

- Board State, pointers, and variables
- Where can I find my Boards etc.
- NT vs Unix and Config Files
- TestManager (reprise)
- AOB

N.B. $\sim 10^{12}$ bits through ¿ production ? Vrb this AM

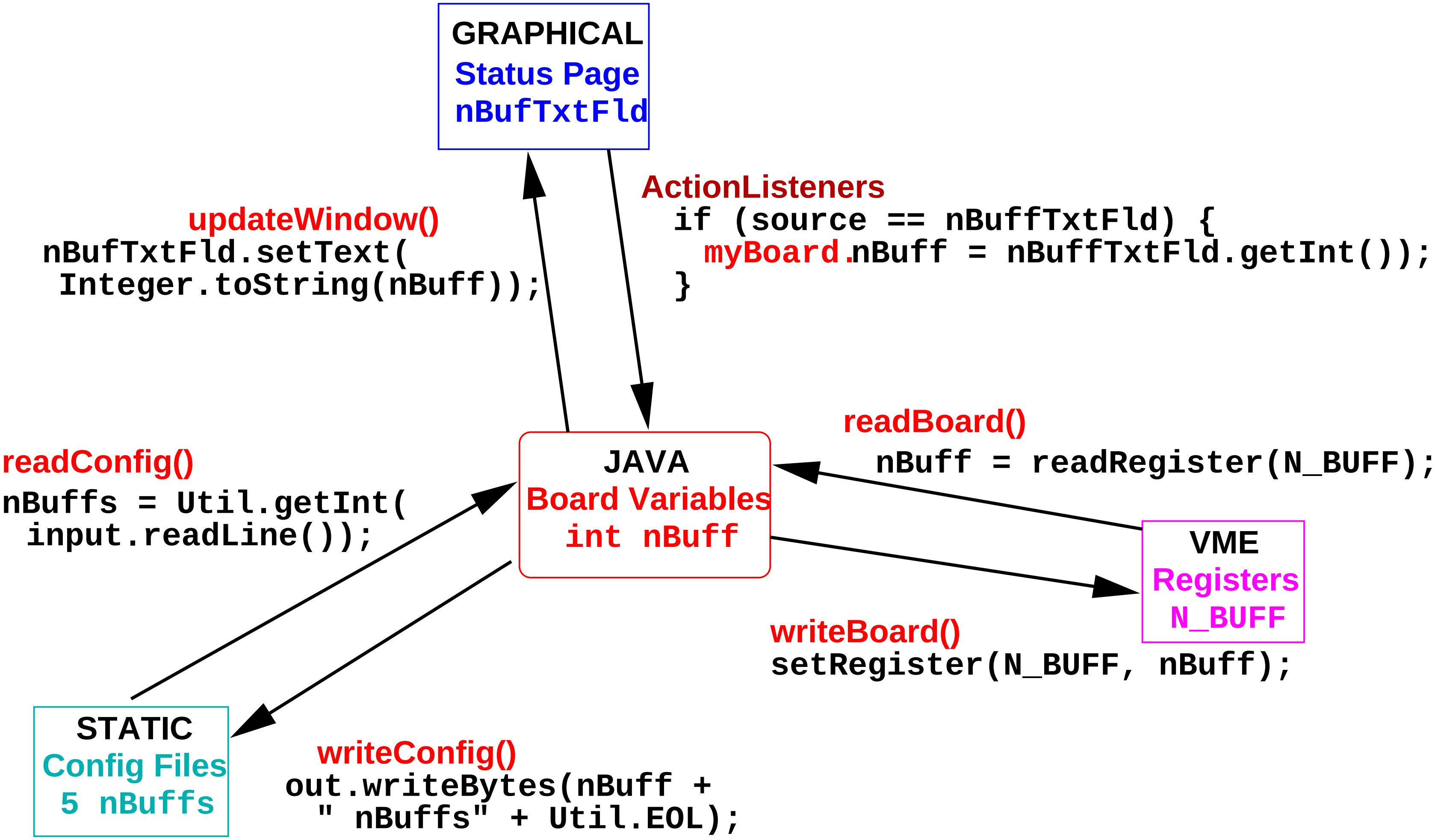
What do I do with the Status Page?

Board State ≡ particular settings of all registers, bits, etc..
Held in several places:

- Static - ASCII Configuration Files ⇒ Database
- Java (`booleans, ints, arrays ...`)
- Graphical (widgets)
  Slow ⇒ static ⇒ 1 *set* per Board type
- VME - Actual values on the registers

Different methods move state from one place to another

- `writeConfig, readConfig`: Java ⇔ Static
  Use to load/save different (or same) board states at init or on the fly
- `updateWindow, Listeners`: Java ⇔ GUI
  GUI use: tricky, more in a sec
- `writeBoard,readBoard`: Java ⇔ VME
  Transferring Entire State to/from board


GRAPHICAL
Status Page
nBufTxtFld
updateWindow()
nBufTxtFld.setText(Integer.toString(nBuff));
ActionListeners
if (source == nBuffTxtFld) {
myBoard.nBuff = nBuffTxtFld.getInt());
}
JAVA
Board Variables
int nBuff
readBoard()
nBuff = readRegister(N_BUFF);
VME
Registers
N_BUFF
writeBoard()
setRegister(N_BUFF, nBuff);
readConfig()
nBuffs = Util.getInt(input.readLine());
STATIC
Config Files
5 nBuffs
writeConfig()
out.writeBytes(nBuff + " nBuffs" + Util.EOL);

Widgets and Pointers

One set of widgets per type, created with first instance of Board type ⇒ Widget Listeners use static pointer

Bad: From template.FirstBoard v1_0 – Look out!

```
clkCB.addActionListener(new ActionListener () {
  public void actionPerformed (ActionEvent e) {
  // Here's where you act
  firstB.clkOn = clkCB.isSelected(); Correct Usage
  readFromBoard = false; Only Changes 1st instance
  updateWindow(); Pushes you back to the 1st instance!!!
  } });
```

Corrected: Use *firstB* pointer which points at displayed instance

```
  // Here's where you act
  firstB.clkOn = clkCB.isSelected(); Correct Usage
  firstB.readFromBoard = false; Changes the correct instance
  firstB.updateWindow(); Updates the displayed instance
  }});
```

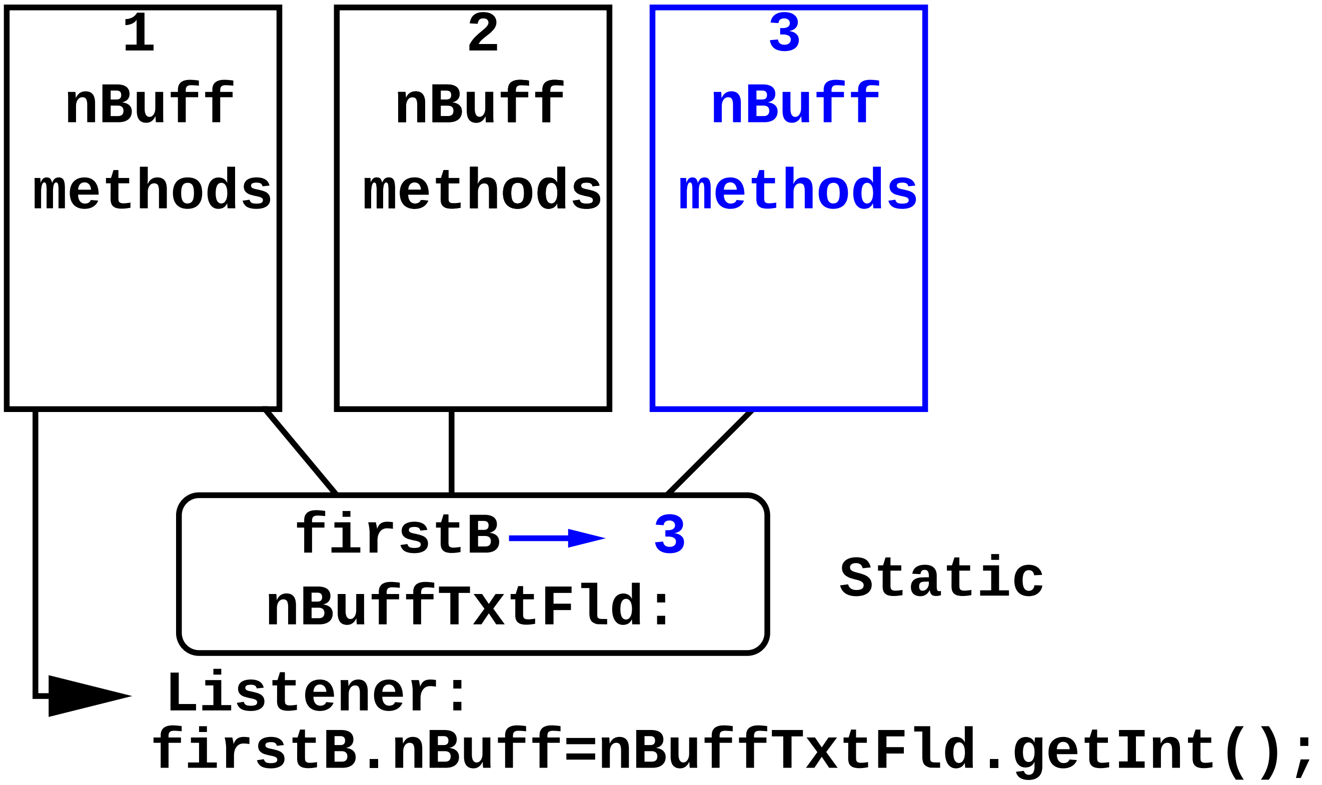
Best: BoardUpdateEvents (new in CDFVME_COMMON)

```
  // Here's where you act
  firstB.clkOn = clkCB.isSelected();
  firstB.fireEvent(new BoardUpdateEvent(
                firstB, BoardUpdateEvent.CHANGED));
  }});
```

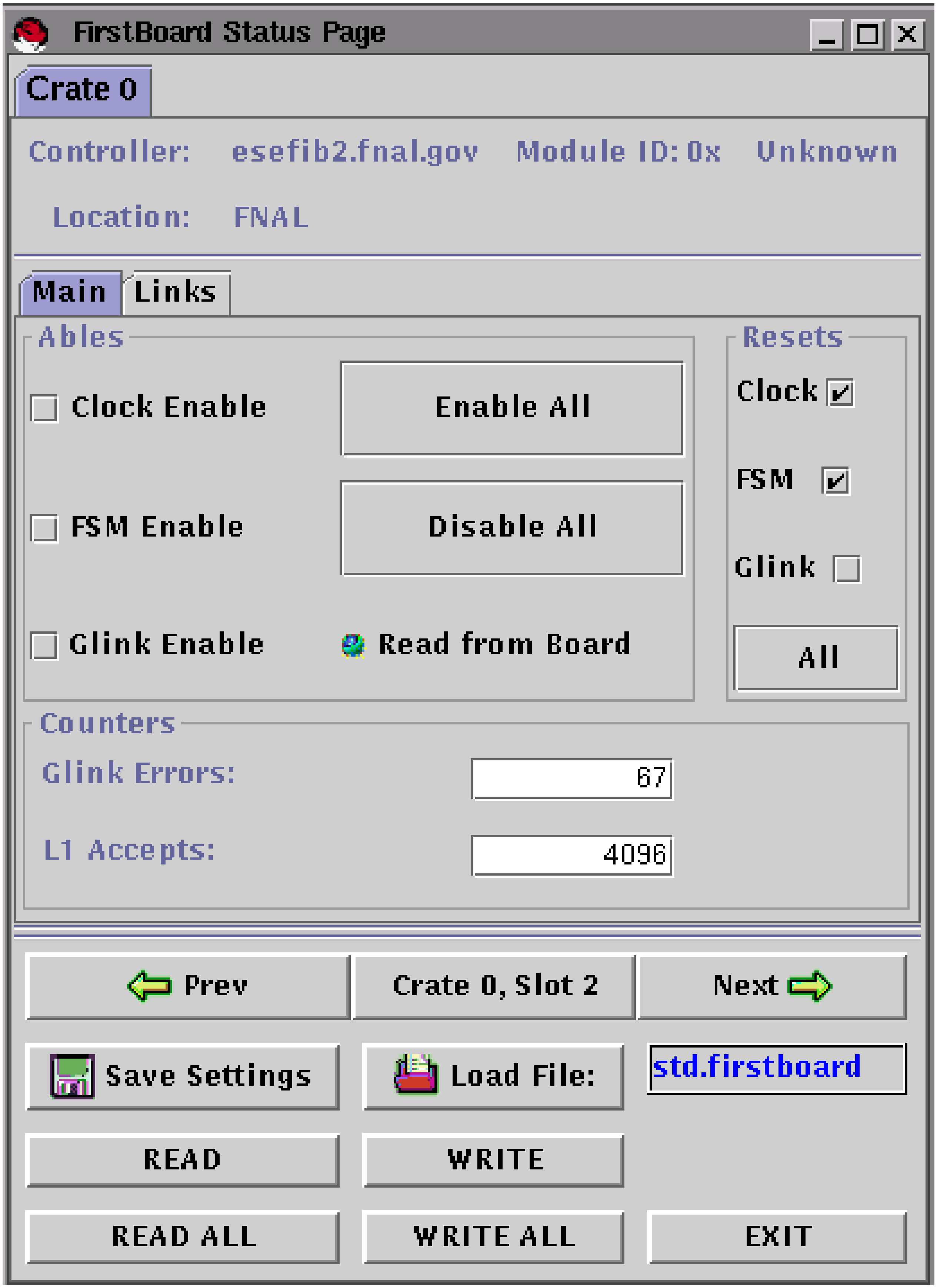
`readBoard, writeBoard`: synchronize VME/GUI

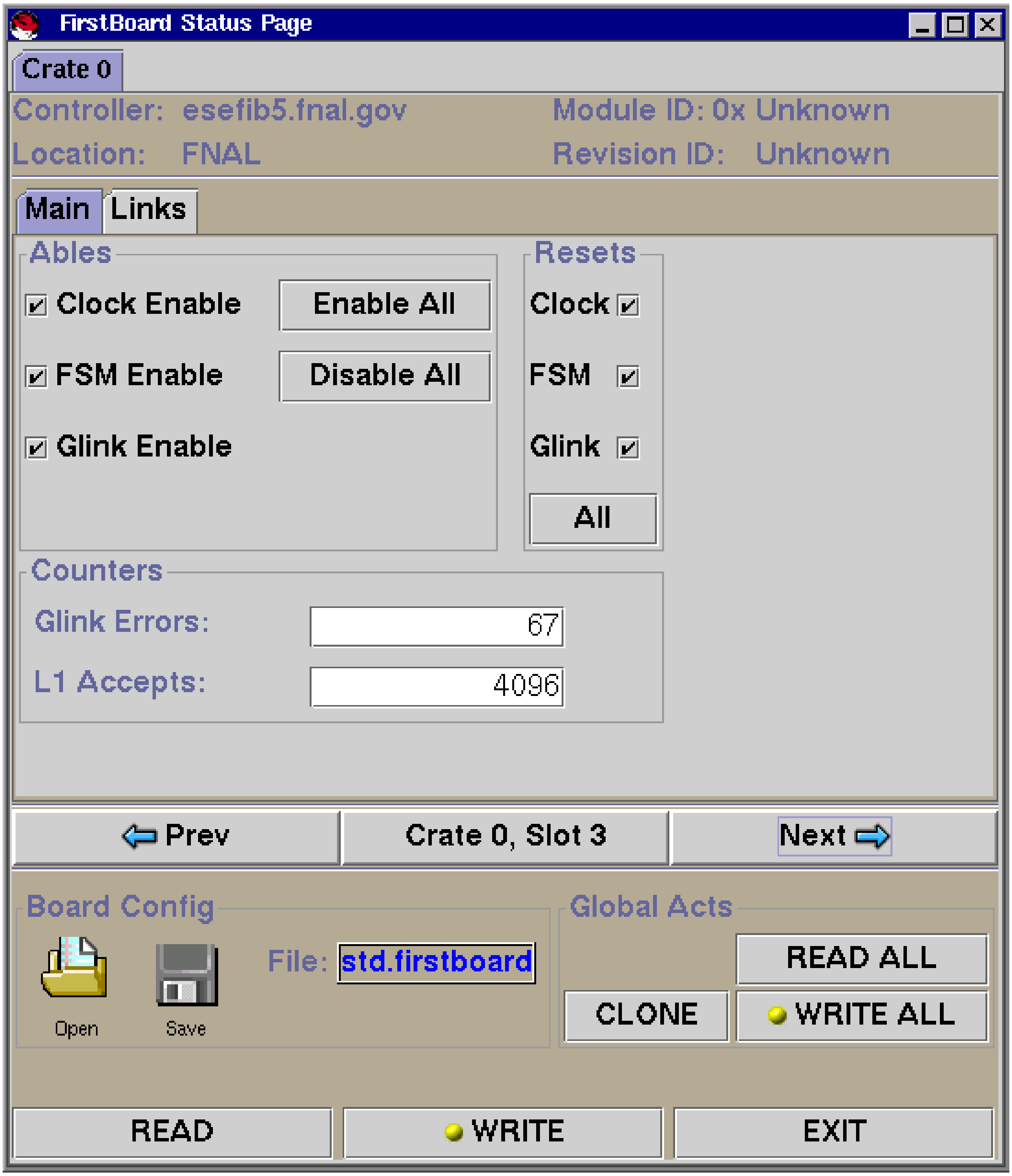
```
  firstB.fireEvent(new BoardUpdateEvent
     (firstB, BoardUpdateEvent.RESYNCHED));
```

Boards



- Board 1 defines listener to change value of *firstB.nBuff*
- *firstB* pointer is set to Board currently displayed (# 3)


FirstBoard Status Page
Crate 0
Controller: esefib2.fnal.gov Module ID: 0x Unknown
Location: FNAL
Main
Links
Ables
Clock Enable
Enable All
FSM Enable
Disable All
Glink Enable
Read from Board
Resets
Clock
FSM
Glink
All
Counters
Glink Errors: 67
L1 Accepts: 4096
Prev
Crate 0, Slot 2
Next
Save Settings
Load File:
std.firstboard
READ
WRITE
READ ALL
WRITE ALL
EXIT


FirstBoard Status Page
Crate 0
Controller: esefib5.fnal.gov
Module ID: 0x Unknown
Location: FNAL
Revision ID: Unknown
Main
Links
Ables
Clock Enable
FSM Enable
Glink Enable
Enable All
Disable All
Resets
Clock
FSM
Glink
All
Counters
Glink Errors:
67
L1 Accepts:
4096
Prev
Crate 0, Slot 3
Next
Board Config
Open
Save
File:
std.firstboard
Global Acts
READ ALL
CLONE
WRITE ALL
READ
WRITE
EXIT

Other Board Stuff

- Variables defined in `CDF.Board`
  int moduleId & *String revisionId* filled in `updateWindow()`:
  ```
  moduleId = readRegister(MY_MODULE_ID);
  int rev = readRegister(MY_REVISION_ID);
  revisionId = Integer.toString(rev);
  ```
  Or fancier

- BoardFrame Modifications:
  - *Clone* - Imposes displayed state on all instances:
    ⇒ Clever use of writeConfig() and readConfig()
    ⇒ Does not write to VME
  - New Look, Icons, ChooseBoard
  - `PackedGridLayout.java`
    ```
    PackedGridLayout(int rw, int cl, int hg,
                               int vg, int fill)
    ```
    `fill = PackedGridLayout.PACK` Tight as possible
    `fill = PackedGridLayout.FILL_X` Stretches in X
    `fill = PackedGridLayout.FILL_Y` Stretches in Y
    `fill = PackedGridLayout.FILL` Stretches both

Where are my boards?

CDF.CDFcfg (= daq) contains all sorts of accessors:

- Crate+Board Array: daq.crate[i].board[slot]

```
for (int i=0; i<daq.nCrates(); i++)
  for (int j=0; j<daq.crate[i].n(Vrb.type); j++)
    int slot = daq.crate[i].slot(Vrb.type, j);
    myVrb = (Vrb) daq.crate[i].board[slot];
```

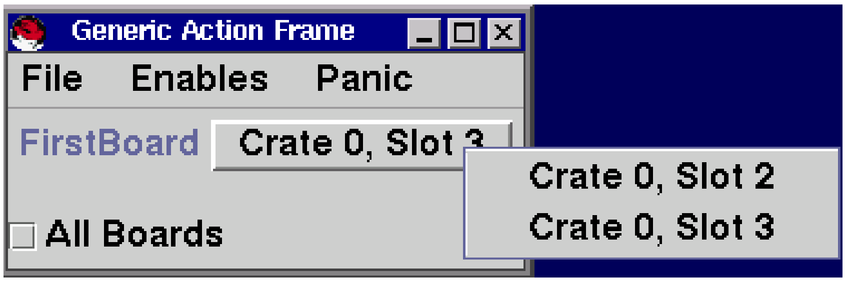
- vector BoardList[type]

```
Enumeration e;
for (e = daq.BoardList[Vrb.type].elements();
     e.hasMoreElements();)
   myVrb = (Vrb) e.nextElement();
```

- Board findBoard(String server, int slot) - returns Board
- int server2crate(String serverName) - returns Crate #
- int type(String className) - returns Board type #
- int nTotal[type] - returns total # of Boards
- Board currentBoard[type] - returns “current” instance
- Vector userObject - place to stick anything with `public String identifier()` method
- Javadoc reasonably up to date, have a look

Expert Frames and Such

TstandAction- Working off the Status Page



- Uses ChooseBoard to get reference to the Board:

```
chooser = new ChooseBoard(FirstBoard.type);
firstB = (FirstBoard) chooser.getBoard();
```

- Actions again use this pointer:

```
if (source == enableClockMI)
  firstB.enable(FirstBoard.CLOCK);
```

- Can execute action on all FirstBoards
- Runs "Standalone"

```
  public static void main(String args[]){
    boolean talkToTheVME = false;
    try {
      talkToTheVME = (args[0].equals("VME"));
    } catch (NullPointerException e){}
    Tstand myTS = new Tstand();
    CDFPrefs.openConnection = talkToTheVME;
    CDFPrefs.initBoards = talkToTheVME;
    myTS.initSystem();
    TstandAction tsAction = new TstandAction();
    tsAction.setVisible(true);
    //Loop while teststand Action is up
    while (tsAction.isShowing()) {
      try {
Thread.sleep(500);
      } catch (InterruptedException e) {}
    }
    System.exit(0);
  }
}
```

Ascii on NT and Unix

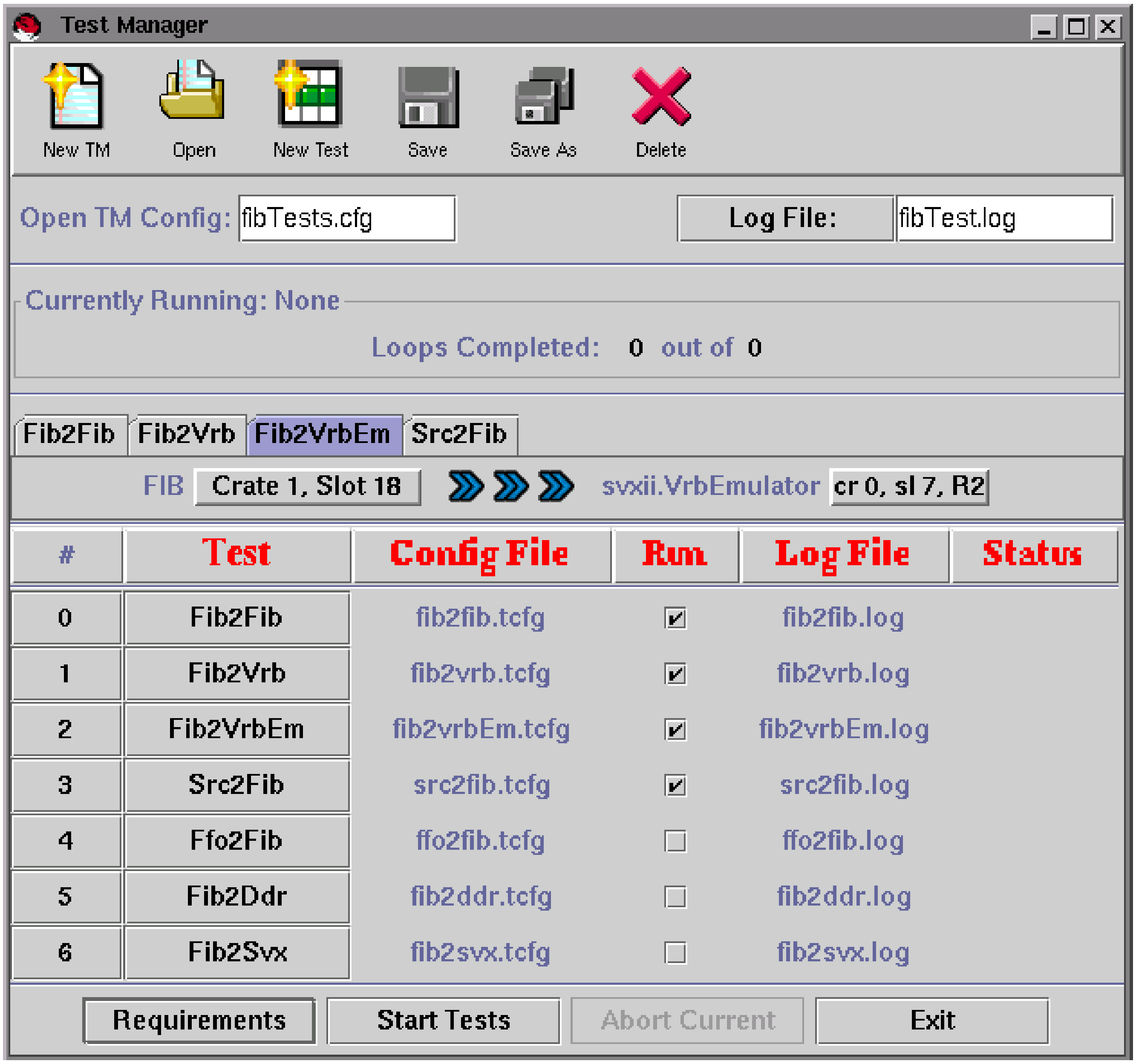
Many ^M characters in the ASCII files - why?
NT linefeed = 0x0a0d vs Unix linefeed = 0x0a
⇒ can cause major heaadaches in parsing files since Java may see a line that your editor does not.
Java not smart before- is now.
CVS is also smart, as is ftp in ASCII mode:

1. Use `CDF.Util.EOL` when writing:
   `out.writeBytes(‘‘My Stuff’’ + Util.EOL);`
2. Never ftp ascii files between NT and Unix in binary mode

Implemented in CDFVME_X

TestManager

Framework for generating batches of tests



Structure

Based on an interface which specifies:

1. What classes the test needs

```
String[] testNeeds = {``tracer.Tracer'',
                      ``admem.Admem''};
```

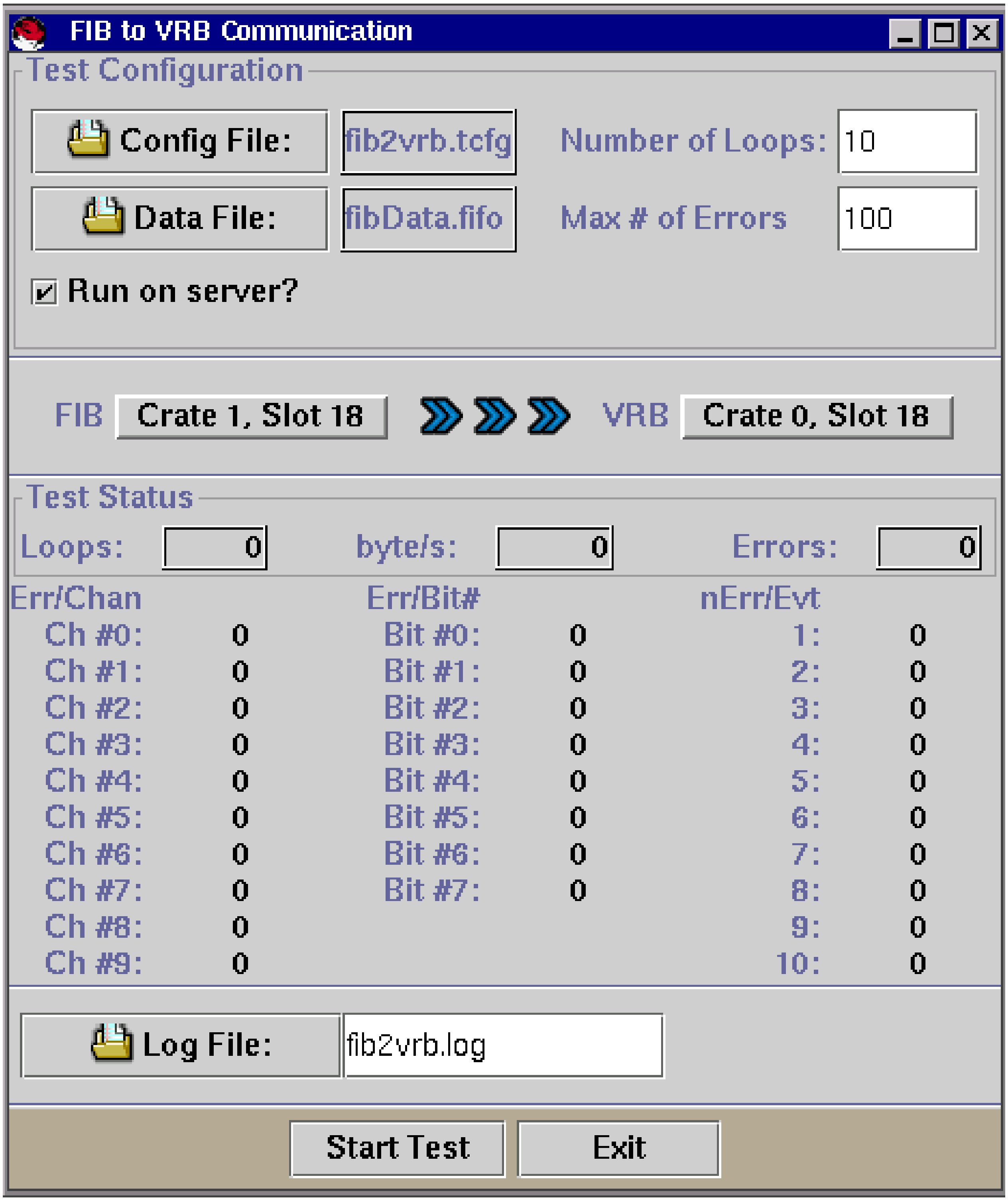
2. How to provide them with GUI

```
getNeedsPanel(), setObjects()
```

3. Log Files, Loop, Config Files, Errors ...

Can adopt as much of CDFVME_COMMON code as you want:

- interface TestModel - minimal interface to implement
- class TestObject - mimimal implementation of TestModel
  Defines "run" as:
  1. `init()`
  2. `nLoops` × (`loopAction()` + `updateStatus()`)
  3. `finish()`
- class TestBase common Objects and methods defines, threads handled
- class TestBoards - lays out some (modifiable) GUI
- class LinkTest - Specific "A to B" test


FIB to VRB Communication
Test Configuration
Config File:
fib2vrb.tcfg
Number of Loops:
10
Data File:
fibData.fifo
Max # of Errors
100
Run on server?
FIB
Crate 1, Slot 18
VRB
Crate 0, Slot 18
Test Status
Loops:
0
byte/s:
0
Errors:
0
Err/Chan
Ch #0: 0
Ch #1: 0
Ch #2: 0
Ch #3: 0
Ch #4: 0
Ch #5: 0
Ch #6: 0
Ch #7: 0
Ch #8: 0
Ch #9: 0
Err/Bit#
Bit #0: 0
Bit #1: 0
Bit #2: 0
Bit #3: 0
Bit #4: 0
Bit #5: 0
Bit #6: 0
Bit #7: 0
nErr/Evt
1: 0
2: 0
3: 0
4: 0
5: 0
6: 0
7: 0
8: 0
9: 0
10: 0
Log File:
fib2vrb.log
Start Test
Exit

Summary

- Hope I did more good than bad
- If you have questions ask! It may take us 5 minutes to find what otherwise may take days.
- CDFVME_X release 1.1 next week or after holidays - hopefully with more documentation
- b0dau30 has everything (finally, one machine does all)
- Items not discussed:
  1. Several Bug fixes to Common code
  2. VxWorks related issues making “global” server code
  3. NT classpath issues (see cygwin mailing list)
  4. ...