

# Running scans in SR1

## Running a Scan in SR1

### Motivation

The motivation for writing this short note comes from the difficulties I had with running any simple scan during my first weeks of working on ATLAS pixel. There were some notes available but most of the help

I obtained was passed down to me by email from various experts whom I had to contact regularly in order to get the necessary help to run a scan. The following should provide a complete set of notes which I used in order to run a scan (which scan for now does not matter, how the various scans differ will be described in another note) on a ROD in SR1. If something is missing feel free to consult me, or suggest the documentation be updated. This original note was written in December 2008/January 2009 and, as such, by the time you are reading this, things may have changed/simplified/moved-on.

### What you will need and how to get it

- CERN (nice/afs) account: I assume if you are reading this you probably have one already. If not check with your group leader or a group member in order to follow the procedures of obtaining the account.
- SR1 account: In order to do this the only way I know of is to send a private email to two individuals, paolo.morettini@cern.ch and nicoletta.garelli@cern.ch. They will ensure you have an account setup and are able to use it. The username and password should be the same as for your nice account.
- Point 1 account: Not entirely necessary for what we will cover in this note but quite important to have taken care of if you will be doing other things such as shifts etc. This is done via EDH (like applying for the CERN GSM phone and booking training etc is done).
- Computer with X11 forwarding enabled.

### Introduction

I will provide here only those details and commands to run a scan. There may be minimal detail of what is actually happening when you execute the commands but for the most part I leave such description for later tutorials. The important lessons to learn from this are the tools required to perform a given scan, which commands to execute and how/where to execute them. The aim is to provide enough detail such that practically any user could perform these scans without expert supervision. Clearly some level of expertise is required to interpret the results of the scan and perform some higher level procedures.

### Procedure

#### Logging in

It may seem like a basic stage but it's good to start things off right.

We perform the scan remotely. That is we begin by opening a shell and logging into our SR1 account. When we login to lxplus we should first be sure to issue the '-X' command so that ssh can then provide

X11 forwarding

```
> ssh -X lxplus.cern.ch -l [username]
```

From lxplus we then do:

```
> ssh atlsr1.cern.ch
```

This should take us into the machine called 'analysis1'. There are machines such as 'control5' and 'control4' and other SR1 machines but for now let's just concentrate on getting things started. It is likely you will need to login to these other machine to compile code or perform other operations. So once you have logged-on to analysis1 you should login to control4:

```
> ssh control4
```

#### Checking out and compiling the code

Check which version of the pixel code is required at this time. In order to check out and compile this code one must execute the following commands. This is true for SR1.

```
> cd
> mkdir myDevelDaq
> ln -s myDevelDaq daq
> cd daq
> cvs co -r PixelDAQ-x-y-z Applications/Pixel/Scripts
> cd Applications/Pixel/Scripts
> ./CheckOut.sh
```

First of all, you need to setup your environment before starting a compilation, then enter in your installation area and execute the compilation script.

In SR1, the installation is performed in the ~/daq directory, even if ~/daq is a soft link:

```
> daq_setup
> cd ~/daq/Applications/Pixel/Scripts
> ./Compile.sh
```

If one wishes to perform the installation in another directory called for instance ~/xx/yy:

```
> daq_setup xx/yy
> cd ~/xx/yy/Applications/Pixel/Scripts
> ./Compile.sh
```

It is important to note that the code, when compiled without optimization, can take a long time. The most recent time I performed this the time taken was on the order of 3 hours. But this could vary a lot.

### An aside on Toothpix

The instruction of using FSM in ToothPix is here:  
[https://twiki.cern.ch/twiki/bin/view/Atlas/ToothpixWiki->ToothPix Manual](https://twiki.cern.ch/twiki/bin/view/Atlas/ToothpixWiki->ToothPix%20Manual)

The ToothPix FSM is the user interface for turning on and off the detector. The link above explains how to turn on and off ToothPix and perform actions therein. Additional expert information is provided on this page. For the most part, performing scans and tests one will not need to turn on and off the detector, so it is more than likely that this information will not be required but it is good to be familiar with it all.

### Launching the remote Desktop

The Finite State Machine (FSM) is controlled using a remote desktop connection to DCS. This is executed as follows:  
`rdesktop -g 85% dcs4`

From here you can control whichever ROD(s) you are using to test code or perform scans. A knowledge of DCS is necessary in order to do this successfully and one should either consult with an expert or read the respective wiki pages to get an understanding of what is actually going on. The DCS shift training should probably be taken by all of those personnel planning to do a considerable amount of work on SR1.

### Starting the infrastructure

Start Infrastructure, and just power up module.

Check the environment:  
`~/daq/Applications/Pixel/PixRCD/PixRCD-00-01-00/PixRCDConfiguration/SR1/partitions/PixelEnv.data.xml`

For running on the NewDSP the most important one defines the path of the slave DSP which is currently NEWDSP-SCHSU - this gives ROD and BOC configuration.  
For example, the slave DSP in that tag is pointed to: `/home/pjackson/DSP/sdsp6713_xcode.bin`  
The above path of the slave DSP needs to be changed in the code in the following location:  
`~/daq/Applications/Pixel/PixRCD/PixRCD-00-01-00/PixRCDConfiguration/SR1/partitions/PixelEnv.data.xml`  
The CONNECTIVITY\_TAG\_CF and CONNECTIVITY\_RUN\_TAG\_CF need to be consistent with the NewDSP path.  
n.b. the default value is currently TOOTHPIX-DEFAULT so one can search and replace this with NEWDSP-SCHSU.

Check the machines used for different services:  
`~/daq/Applications/Pixel/PixRCD/PixRCD-00-01-00/PixRCDConfiguration/SR1/segments/Segment_PixelInfrastructure.data.xml`

Generally one should cd to the following directory:  
`~/daq/Applications/Pixel/PixRCD/PixRCD-00-01-00/PixRCDConfiguration/SR1/`

From this subdirectory the following files will need to be changed:

```
./partitions/PixelEnv.data.xml
./partitions/Partition_Pixel.data.xml
./segments/Segment_PixelInfrastructure.data.xml
./sw/PixSW.data.xml
```

It is important to change all of the instances of "pixeldaq" to be "[username]" for your directory. Also, one should change "PixelInfr" -> "PixelInfr\_[username]". Instances of "PixelDD" should be changed to "PixelDD\_[username]". There are certain caveats to this. One such caveat is that anything beginning with "Segment\_" with PixelInfr in the statement should NOT be changed to include the username.

Note: If not all of these have been changed suitably then one will have trouble successfully starting your infrastructure and initializing the ROD will fail.

In order to start the TDAQ infrastructure one should execute the following command. Normally, I tend to store the command line in an executable file, which I call 'startdaq\_infr'.

```
cat ~/startdaq_infr
setup_daq -i obk -i cdi -i logger PixelInfr_[username]
```

When the TDAQ partition is launched you should then be able to go through the execution of the various commands:  
Boot -> Initialize -> Config -> Start. Do all of these until everything is up and running in Run Control panel.

In order to control the partition, check that it is running, and ultimately kill it, the following two commands will be of help:

How to check the current partition: `ipc_ls`  
How to kill the started partition: `pmg_kill_partition -p PixelInfr_[username]`

### Flashing the ROD

In order to ensure that the ROD which you want to test is using the version of the code you wish to test you have to flash that executable (master binary) onto the ROD. There are a couple of scripts to make this easy for you. In order to do this, you first need to determine which crate the ROD you are using is situated on. There are two choices crate0 and crate1. To make it simple, RODs 11, 15 and 20 are on crate0 and all of the rest (5,12,14,17,21) are on crate1. To flash crate0 you must login to the machine pixrcc04, and for crate1 access pixrcc03. Once you have logged-on to either pixrcc04 or pixrcc03 you then cd to the following location: daq/RodDaq/RodUtils. From there you have to execute two commands:

```
> ./MdspFlashLoad -f
```

When prompted, type the number of the ROD you wish to flash. Then you will be prompted for the binary to flash. This is the master binary which you wish to use. In many cases it will be: ../NewDsp/Binary/master.bin.

Once this is completed, you will need to execute the following:

```
> ./RodReset
```

and when prompted, type the number of the ROD you have just flashed.

Below I provide a list of all of the RODs accessible in SR1 and PP0 with which they are linked:

Crate 0

```
11_AAD__L0_B12_S2_C6_4R_B__ROBIN_8
15_AA___L2_B99_S1_C7_5L_T__ROBIN_10
15_DD___L2_B99_S2_A7_6L_T__ROBIN_10
20_AA___L2_B99_S1_A6_6L_B__ROBIN_11
20_DD___L2_B99_S2_C6_5L_B__ROBIN_11
```

Crate 1

```
05_AAD__L0_B12_S2_A7_3R_T__ROBIN_3
12_AAD__L0_B12_S1_C7_4R_T__ROBIN_0
14_AA___D1A_B05_S1___1R_T__ROBIN_4
14_DD___D1A_B05_S2___2R_T__ROBIN_4
17_AA___L0_B12_S1_A6_2R_B__ROBIN_9
21_AA___L2_B27_S2_A7_5R_T__ROBIN_1
21_BB___L2_B27_S1_C7_6R_T__ROBIN_1
21_CC___L2_B27_S1_A6_5R_B__ROBIN_1
21_DD___L2_B27_S2_C6_6R_B__ROBIN_1
```

### Startup Console and do the SCAN

```
~/daq/Applications/Pixel/CalibrationConsole/Console
```

Then execute ./Console &

- Select TDAQ partition for PixelInfr\_[username] -> make sure PixelInfr is also started since we need DDC service.
- Goto 'command tool' panel, right click to pop out the command menu. Select 'DIGITAL\_TEST'. Select the 'Base tag' and 'latest revision', i.e. just click ok to start the scan.
- We can move to the middle button to select the scan ID and look at the final histograms.

It is useful to note that the Console provides a lot of functionality which I will not touch on at all here. Importantly for testing scans with the NewDsp code is the list which clicking the right mouse button in the 'command tool' panel gives. There are many different scans which one can execute, all of which test slightly different aspects of the pixel detector performance and functionality.

### Summary

In summary, there is considerable overhead involved in running anything successfully in SR1. These are just a few instructions to help you along your way with getting started. More detailed pieces of information are available on the various parts of the pixel wiki that are in place to document these things (DCS, DAQ etc).

### Useful Commands

The directory daq/Applications/Pixel/PixLib/Examples contains several useful scripts to assist the user in their work. I outline just a few of them below but the individual should enquire about others should they prove suitable to solve a given problem.

Please note that before each and every command I will provide here it is assumed either that the user is executing them from the daq/Applications/Pixel/PixLib/Examples directory, or that the path is put before the command.

```
> ChangeCfg --dbPart PixelInfr_pjackson --idTag SR1 --setBinPref /home/pjackson/DSP --connTag TOOTHPIX-V5 --cfgTag NEWDSP-PJACKSON-BOC --pendTag OnD
```

This changes the configuration to my own path. You will need to have the DbServer running (i.e. have launched TDAQ) in order to correctly perform this. The cfgTag flag needs to exist. If you do not have a tag of your own you either need to use someone else's or request one. The setBinPref points to the directory where the slave image is located, or at least provides a softlink to it.

In order to check the EXTFile settings you can use the following command:

```
DbServer_monitor --dbPart PixelInfr_pjackson --domain Configuration-SR1 --tag NEWDSP-PJACKSON-BOC --dumpObj --objName ROD_C1_S17_ROD
```

This will check the settings of the specific ROD you pass to it. You probably want to iterate all RODs to make sure the settings are consistent.

One can dump the information about your partition with the following commands:

```
./DbServer_monitor --dbPart Pixellnfr_pjackson --dump
```

```
./DbServer_monitor --dbPart Pixellnfr_pjackson --domain Configuration-SR1 --tag NEWDSP-PJACKSON-BOC --dumpObj --objName ROD_C1_S17_ROD
```

Again for these commands you need to have the DbServer running in order to run them (there is nothing to monitor or dump if the DbServer isn't up).