The SEE Test Facility Data Acquisition System User's Guide

Ron Fox (fox@nscl.msu.edu)

May 3, 2004

Abstract

This document describes how to use the SEE Test Facility Data Acquisition System (SEETFDAQ). The SEETFDAQ is a turnkey realization of the NSCL Data acquisition system. Full documentation on the components of the NSCL Data Acquisition system is available at various online documents at the NSCL documentation web server: http://docs.nscl.msu.edu

In this document you will find sections that

- Provide a simplified introduction to the SEETFDAQ system.
- Describe the components of the SEETFDAQ system and how to interact with each of them.
- Describe the files that make up the products of a recorded run and how they are organized in the experimental directory system.
- Describe how to write your data to DVD after the experiment is over.
- The online version of this document (hyperlatex) is available at http://docs.nscl.msu.edu/daq/see/seeuser.html.
- The printable version of this document (postscript) is available at http://docs.nscl.msu.edu/daq/see/seeuser.pdf.

Refer to the table of contents below to locate specific sections of the document.

Contents

1 Introduction to the SEETFDAQ System

The SEE Test Facility DAQ (SEETFDAQ) system is a specific realization of the NSCL Data Acquisition system. The SEETFDAQ system has been packaged in a near-turnkey fashion.

The NSCL data acquisition system is an open source system that is described online at http://docs.nscl.msu.edu. To get an overview of the system, you should refer to: http://docs.nscl.msu.edu/daq/overview/.

2 Components of the system

This section describes the system components you will interact with. There are other components that are mostly hidden from view. The components you will learn about are:

SEE Control The SEE Control application allows you to set up the slow control hardware of the SEETFDAQ system.

Readout The Readout subsystem responds to triggers, reads out events and periodically reads scalers. It is the actually the data acquisition part of the system.

Scaler The Scaler display subsystem allows you to view the rates at which detector components of the system are triggered.

SpecTcl The SpecTcl program and its companion data viewer Xamine are the on-line data analysis component of the system. SpecTcl produces various spectra from the online event data and Xamine displays those spectra for you.

2.1 SEE Control

The SEE slow control system control panel is how you will interact with the slow controls system. To start up the panel you can either enter the following in a terminal window:

$ **gomenu &**

or locate and click once on the desktop shortcut labeled *User Controls*

The SEE slow control system panel is shown in the figure below:

The numbers in the description below refer to the numbers on the diagram.

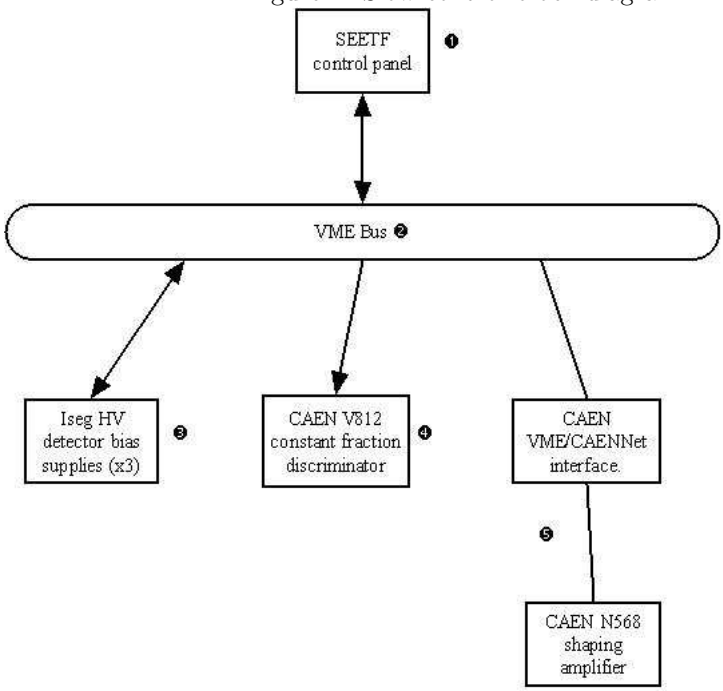
1. The SEETF control panel provides a graphical user interface that allows you to recall slow control settings packages that have been created by the NSCL staff by the beam species and energy.

2. All of the controls devices are plugged into the SEETF VME crate backplane. You will interact with the SEETFDAQ slow control system through the graphical user interface (GUI). provided by this control pane.

3. Detector Bias is supplied by three ISEG VME high voltage VME power supplies. Each supply has two channels. Two of the supplies bias the four quadrant scintillator. The third supply provides bias for the PPAC. The control GUI sets the target voltages and ramp rates for these devices, and allows you to initiate ramps and shutdowns.

4. Detector trigger determination is provided by a CAEN V812 constant fraction discriminator. This module is programmed by the SEETF control panel when a settings package is selected. It is also programmed by the Readout software whenever a run starts.

Figure 1: Slow control block diagram



5. The CAEN N568 is a NIM spectroscopy amplifier module. It is remotely programmable through it's CAENnet interface. CAENnet is a proprietary serial bus and protocol developed by CAEN. A CAENnet controller module in the VME bus allows the SEETF control panel to setup this module. The N568 is programmed as soon as a settings package has been selected. It is also programmed by the Readout subsystem whenever a run starts.

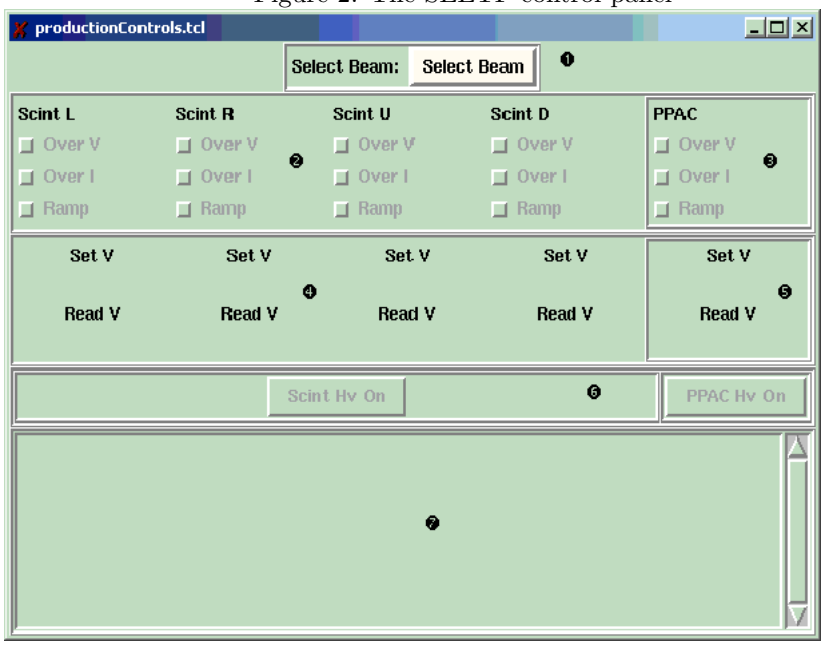
The SEETF slow control system is turnkey. NSCL staff will develop and store appropriate settings packages for each type beam particle and energy you will use. The control panel GUI allows you to:

- Select the settings pacckage appropriate to the beam you will use in your experiment
- Ramp detector bias supplies on and off and monitor the status of the bias supplies.

The SEETF GUI is shown in the figure below:
The numbers in the description below refer to the numbers that label the figure.

1. The "Select Beam" button pops up a cascading menu that allows you to specify the beam you are using by first selecting a particle type and

Figure 2: The SEETF control panel



then the energy. When you select the beam, the button will display the particle and energy. Note that as soon as you select an Ion and energy, the settings package associated with this beam is copied into your experiment directory (/experiment/config), and the control panel sets up both the shaping amplifier and the constant fraction discriminator.

2. This scintillator status section displays the status of the trip conditions for the two supplies that bias the four segment scintillator.

3. This PPAC status ssection displays the status of the trip conditions for the supply that biases the PPAC.

4. This scintillator status section displays the requested and actual voltages on the four channels biased by the scintillator bias supplies.

5. This PPAC status display shows the requested and actual voltage for the PPAC bias supply.

6. The buttons in this section of the GUI allow you to initiate power supply ramps for the bias supplies. The ramp speed is specified in the settings package that you selected when you specified the Ion and energy. When you initiate a ramp, the corresponding buttons turn into off buttons. You may turn off a supply set at any time, even if the ramp has not yet completed. The turn off ramps the power supply to zero at the fastest rate supported by the hardware.

7. This scrolling window captures output information from the control panel. It logs the initiation of ramps, trips and their recoveries. Information sent to this window is also logged to file in /experiment/Controls.log

2.2 The Readout Subsystem

This section describes the SEETFDAQ readout subsystem. In this section you will learn:

- The general structure of the Readout subsystem.
- How to use the control panel for the Readout subsystem.

The Readout subystem is what makes the SEETFDAQ system a data acquisition system. It:

- Manages data taking runs.
- Responds to event triggers, reading out the data associated with the trigger.
- Periodically reads out scaler values.
- Submits data to the remainder of the system, making it available to other components of the system.
- Initiates event recording to disk when appropriate.

subsubsReadout subsystem structurerdosstruct

The structure of the Readout subsystem is shown in the figure below.

For simplicity, the data distribution system has been omitted and event data is shown flowing directly from the Readout program to the EventLog program. The Control Panel application maintains the cohesion of the readout program. Your interactions with this control panel make the system work.

The control panel GUI lets you:

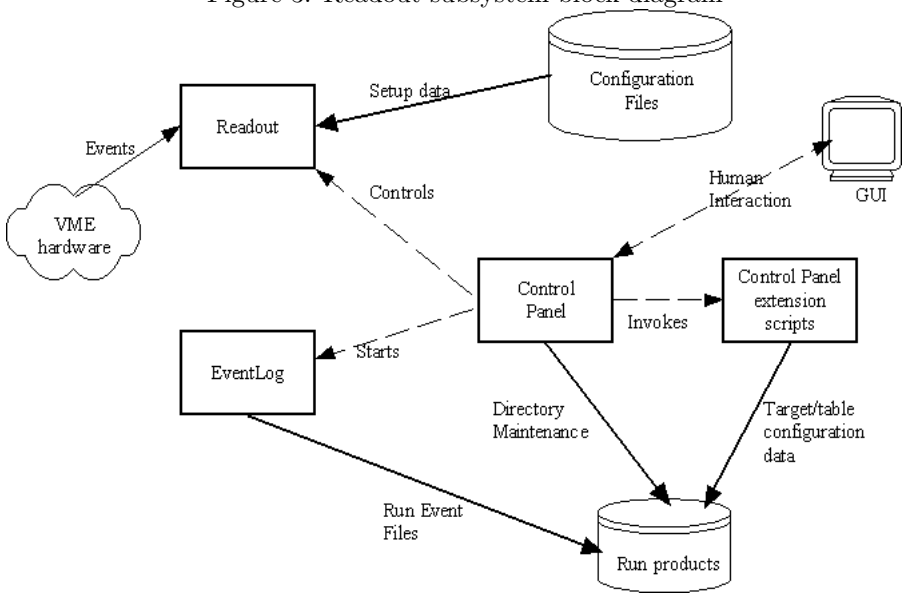
- Start and restart the readout program.
- Begin and end data taking runs.
- Set the title and run number of upcomming data taking runs.
- Produce a "production run", that is a run whose products will be saved when the run exits.

How you interact with the control panel is described in the next section.

In addition, the control panel autonomously manages you run products in keeping with the directory and file structure defined in http://docs.nscl.msu.edu/daq/overview/standard_nscl_di

The control panel also supports user written extensions. A SEETF user written extension ensures that information about the EPICS channels that describe the target and table position are written to file and incorporated amongst the run products for production runs.

Figure 3: Readout subsystem block diagram



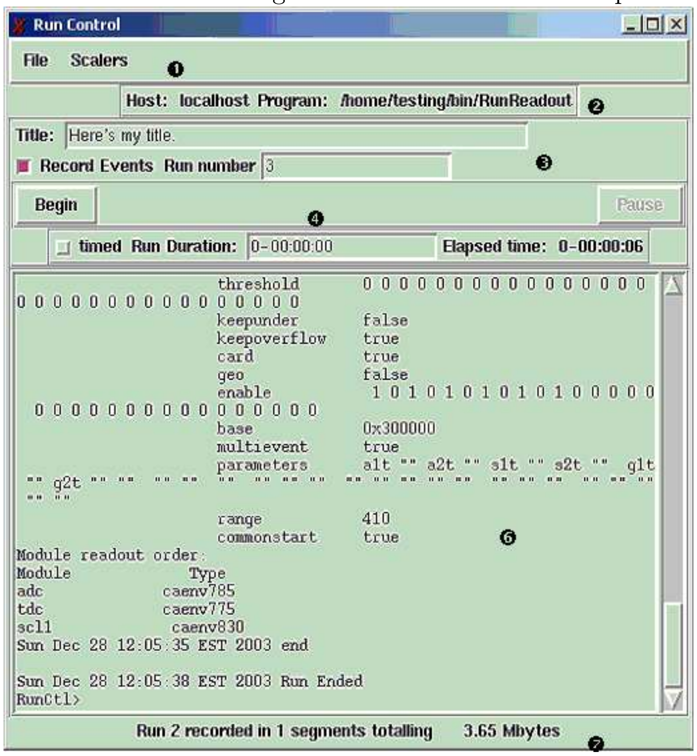
The Readout program is, of course, an important component of the Readout subsystem. It:

- Responds to run control command messages sent to it by the Control Panel application.
- At the beginning of a data taking run, it reads the slow controls setting packages and ensures that the V812 CFD and the N568 shaping amplifier are set in accordance with the settings in the current package.
- At the beginning of a data taking run it reads a configuration file that defines the hardware to read and the order in which to read it.
- During a run responds to triggers from the VME hardware and reads out the event that caused the trigger as described by its configuration files.
- During a run periodically reads out scalers that describe the trigger rates in the detectors.

Finally, the EventLog program is run prior to the start of a production run. EventLog produces one of the most imporant products of a run; the event file. The event file contains the data from events that were readout by the Readout program. The format of this file is described online at:

http://docs.nscl.msu.edu/daq/ProductionReadout/Manual/. Scroll the left pane of this document down until you see the link *Event data format* . Click on that link.

Figure 4: The Readout control panel



2.2.1 Operating the Readout Control Panel

This section describes the Readout Control Panel. You will learn:

- How to start the Readout Control Panel
- How to set the run title and run number.
- How to initiate a run.
- How to indicate a run is a production run.
- How to initiate a timed run.

The figure below shows the Readout control panel.
Refer to the numbers in the figure in the discussion below.

1. The top strip of the control panel includes a menu bar. The File menu allows you to start and stop the readout program, exit the GUI, and select a new readout program. Note that the RPI data acquisition system uses a unique scriptable readout program and therefore you will never need to select a new readout program. The Scaler menu allows you to select

readout parameters for the periodic scalers, such as the interval between periodic scaler readout.

2. This stripe documents the readout program that is actually run. The RPI readout system actually runs a script that in turn runs the Readout program passing appropriate command line parameters to the real Readout program. This script is set-up for you when you create data acquisition accounts. For more information about account setup, see the System installation and management guide.

3. This segment of the user interface allows you to document the run conditions. This section of the interface is disabled while runs are in progress. In this section you can:

   - Set an identifying run title.
   - Set a run number. Note that if runs are being recorded to disk, the number you set will be automatically incremented at the end of each run.
   - Determine whether or not the run is being recorded to disk. Note that when runs are being recorded, the control panel background turns green to indicate this.

4. This strip contains the actual run controls. The Begin button initiates a data taking run with the current set of conditions. Once the run has started, this button becomes an End button and is used to end a run. The Pause button is enabled when data acquisition is in progress and allows you to suspend a run. When the run is suspended, the Pause button becomes a Resume button. Suspended runs can be Resumed by clicking the Resume button or they can be ended without any additional data taking by clicking the End button.

5. This strip allows you to set the parameters of a timed run. The controls in this strip are disabled when data taking is in progress. By checking the timed run box and entering a time in the Run Duration text entry prior to starting a run you indicate you want a fixed length run. Note that regardless, the Elapsed time field of this strip indicates how long a run has been active (Elapsed time does not count during a pause).

6. All output and error messages from the Readout program are captured in this scrolling window. Note that if you have an error in your Readout configuration script this will be indicated here along with recommended corrective action.

7. This status strip indicates how much data has been taken to disk for the active run. It updates periodically while a run is in progress. Note that the EventLog program understands how to segment runs into files that are less than 2Gbytes long (a Linux file size limit), and that the ReadoutGui knows how to handle runs that have been segmented in this way.

2.2.2 Operational procedures

This section describes various task-oriented operational procedures. You will learn:

Starting the Readout contol panel You can start the readout control panel either from the command prompt or from a shortcut on the desktop. To start the panel from a command line type:

```
$ godaq &
```

To start the panel from the desktop shortcut, locate the shortcut labelled "Readout" and click it once with the left mouse button.

Setting the Run Title and number To set the run title and run number:

- Click in the box labeled *Title* and edit in the new title text. The box supports simple editing.
- Clock on the box labeled *Run number* edit in the new run number.

Note that the run parameters like the run title and run number only have meaning for an active run. They may also only be changed when the run is inactive.

Initiating a run The readout software is a state machine. The set of states it can be in are:

Table 1: Readout state-machine states

State	**Description**
Inactive	No current run. No data taking
Active	There is a currnet run. Data taking
Paused	There is a current run. No data taking

- To start a run (transition to the Active state), Click the Begin button of the control panel. Note that the begin button changes into an End button.
- To end a run in any of the Active or Paused states, click the End button. Note that the End button turns into a Begin button when the run ends. The run may automatically end if it is in the Active state, the run is a timed run and the run time expires.
- To temporarily halt data taking, enter the Paused state by clicking the Pause button while data taking is in the Active state. The Pause button becomes a Resume button and can be used to re-enter the Active state. While in the Paused state, the timed run timer does not count down. Clicking the End button while Paused initiates an transition to Inactive.

Production runs Production runs are runs that produce permanent products. Run products include event data files and user defined data associated with them.

To start a production run (system in the Inactive state):

- In the entry labelled *Run Duration* enter the desired length of the run.
- Check the check box labelled *timed*
- Click the *Begin* button as usual to start a run.

Initiating a timed run Timed runs can be initiated when the Readout program is in the Inactive state. In the timed run control portion of the control panel:

- Check the box labeled timed.
- Enter the desired run duration in the text entry labeled Run Duration.
- Start the timed run by clicking the Begin button.

2.3 Scaler display subsystem

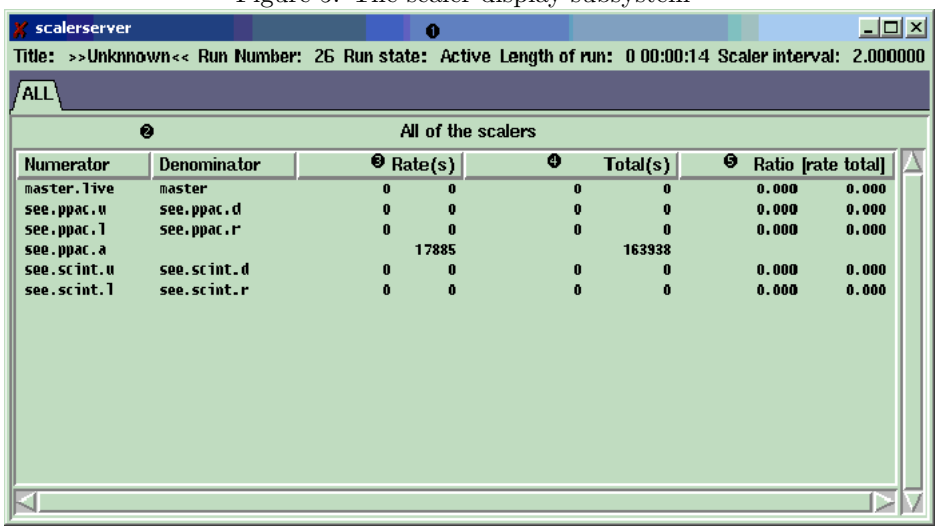
The scaler display subsystem accepts a subset of the data from the Readout software and produces displays that describe the trigger rates and totals over the life of a run. The NSCL scaler display program is extremely flexible. For the SEETFDAQ system it has been pre-configured to display only the set of scalers desired by the SEE experimental groups.

The figure below shows the scaler display program in action.

Refer to the numbers in the picture when following the discussion below.

1. This line displays information about the status of the run. From left to right:
   - The run title.
   - The run number.
   - The run state
   - How long the run has been active.
   - The number of seconds between periodic scaler readouts.
2. Each line of the scaler display program contains either a single scaler or a pair of scaler channels intended to be interpreted as a ratio. For each of these lines, this column displays the name(s) of the scaler(s) on that line.
3. For each display line, this column displays the count rate in *counts/sec* of the scaler(s) displayed in that line. If a ratio is displayed, the first of the two numbers will be the numerator.

Figure 5: The scaler display subsystem



4. For each display line, this column displays the total number of counts for the scaler(s) displayed in that line. If the line is displaying a ratio, the numerator is the first of the two values.

5. For each display line, if the line is displaying a ratio, this column displays the ratio of the rate and total column.

2.4 SpecTcl the data analysis component

SpecTcl is the most complex component of the system. SpecTcl receives event data from the data acquisition system or from file and computes a set of histograms for the user. SpecTcl has been set up in advance for the SEETFDAQ sytsem. Therefore this section will only focus on how to operate SpecTcl from the point of view of the SEETFDAQ setup.

In this section you will learn:

- How to start SpecTcl, connect it to the online or offline system.
- About the components of the SpecTcl user interface.
- Which spectra have been created for SEETFDAQ, and how to set, save and restore variables that influence how these spectra are computed.
- How to interact with Xamine, the SpecTcl display package.

If you are interested in reading more about SpecTcl, complete online documentation is available at: http://docs.nscl.msu.edu/daq/spectcl.

2.4.1 Starting SpecTcl

You can start SpecTcl either by typing:

$ **gospec &**

at a terminal emulator window, or by single clicking on the KDE Desktop shortcut labelled *SpecTcl*. The shortcut should have an icon that looks like a spectrum.

2.4.2 The SpecTcl User interface

The SpecTcl user interface consists of three elements:

- A command window. The command window accepts commands that are an extension of the Tcl/Tk scripting language.
- A GUI window. The GUI window provides buttons for some of the frequently used commands. The GUI window can be extended via Tcl/Tk programming. How to do this is beyond the scope of this document.
- Xamine. Xamine is an X11/Motif program that displays histograms produced by a client program (in this case SpecTcl).

The remainder of this setion will:

- Annotate the command window and its features.
- Annotate the GUI window and describe what each of its buttons do.
- describe how to connect SpecTcl to the online or an offline data source.

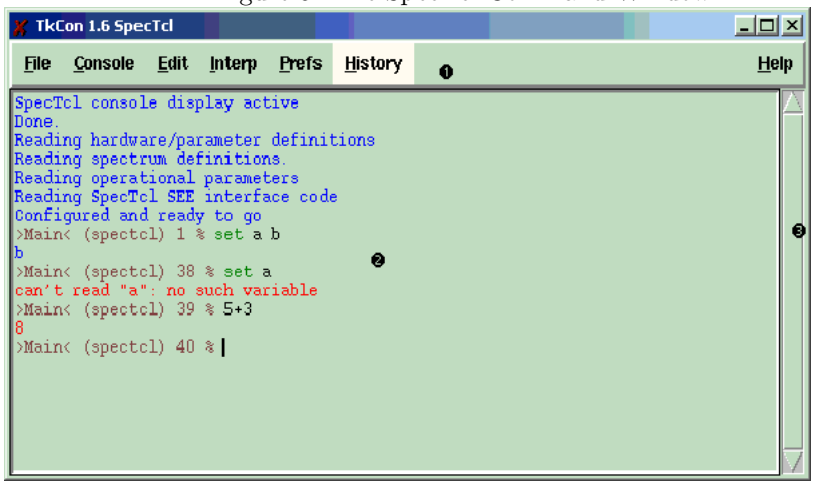
Xamine will be described in a later section.

The Command Window The figure below shows you what the command window looks like.

Refer to the numbers in the figure when reading the text below.

1. The menu bar provides a set of menus that let you access the advanced functions of the SpecTcl command window. The SpecTcl command window is based on the TkCon Tcl/Tk console. TkCon was specialized somewhat to make it harder to make operational mistakes.
2. The text window is where you can enter Tcl/Tk/SpecTcl commands. The output of those commands are displayed here as well. The command interface supports full command line editing, cut and paste, as well as dynamic syntax highlighting.
3. This scrollbar lets you scroll the text window backward to view commands and output that have scrolled off the viewable area of that window.

Figure 6: The SpecTcl Command Window



The GUI Window The GUI Window provides an extensible set of buttons that give you direct manipulative access to common SpecTcl functions. While the GUI window can be extended via Tcl/Tk Scripting, how to do this is beyond the scope of this document. The GUI window is shown in the figure below:

Refer to the numbers inthe figure when reading the description below.

1. The *Start* button begins analyzing data from the current data source. SpecTcl can take data from File, tape, or the online system. When SpecTcl is analyzing data, this button will change into a *Stop* button and you will use it to stop analysis.

2. Clicking this button will clear the contents of all the spectra. This is equivalent to the *clear -all* command.

3. Clicking this button will exit the program (equivalent to *exit* command).

4. Clicking this button will close the current data source, connect SpecTcl to the online data source and start analyzing data from it.

2.4.3 SEETFDAQ spectra

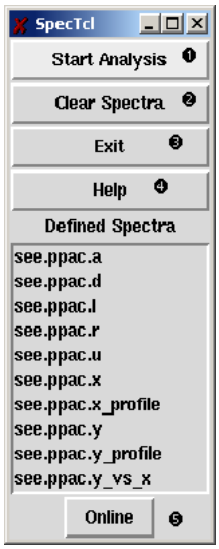
The following spectra have been defined for the SEETFDAQ version of SpecTcl:

see.ppac.{u,d,l,r} These one dimensional spectra histogram the raw signals from the four sides of the PPAC detector. They are mostly used by NSCL staff for debugging purposes.

see.ppac.a This one dimensional spectrum histograms the PPAC anode signal.

see.ppac.{x,y} These one dimensional spectra histogram the computed x and y positions of particles passing through the PPAC.

Figure 7: SpecTcl's GUI window



see.ppac.y_vs_x A two dimensional histogram that plots the computed x position against the computed y position from the PPAC.

see.ppac.{x,y}_profile X and Y profile spectra for the PPAC. profile spectra are essentially compressed position spectra with a settable compression factor.

see.sci.{u,d,l,r} One dimensional energy spectra from the four segments of the segmented scintillator.

see.sci.counts A one dimensional four channel spectrum that histograms the total counts in each of the four segments of the segmented scintillator.

see.trend{u,d,l,r} Trendlines for the four scintillator spectra. A trendline is a spectrum that shows a history of counts vs. time, like a strip chart.

Some of the spectra described above are controlled by parameters. These parameter are read from Tcl variables. The values of the parameters can be set via the Tcl *set* command. The paragraphs below describe:

- How to manipulate TCL Variables.
- How the PPAC position spectra are computed and the parameters that influence this computation
- How the PPAC profile spectra are computed and the parameters that influence this computation.
- How the Scaler trendline spectra are computed and the parameters that influence this computation.

Manipulating TCL Variables The *set* command is used to set and view variable values in Tcl. The form of this command and a pair of examples is shown below:

[htb]
set name

value

% **set ppac.x.scale1 .8993**
.8993
% **set ppac.x.scale1**
.8993
%

Several TCL Variables control the way spectra are incremented. You will naturally want to save the value of these variables so that they can be re-used in later runs of SpecTcl. Several command procedures have been implemented to support this:

SaveParams Saves these parameters to the file named *file*.

SaveStartupParams Saves these parameters to the file named /config/opparams.tcl. This file is read by SpecTcl on startup.

SaveTempParams Saves these parameters to the file named /config/tempparams.tcl

The save files are simple Tcl scripts. The variables saved can be restored via the tcl *source* command. For example in the SpecTcl command window:

[htb]
% **source opparams.tcl**

reloads the default settings for these parameters.

The sample shows two forms of the set command. The first form sets the value of the variable ppac.x.scale1 to .8993. The second form displays the value of the variable without mofidying its value.

PPAC Position Spectra The PPAC position spectra require a calculation of the position. For a given pair of signals (e.g. l and r for left and right), The position calculation is done as follows. First calibrated left and right positions are computed as follows: [

$$l_c = (l + p_l) * m_l + c_l \tag{1}$$

] [

$$r_c = (r + p_r) * m_r + c_r \tag{2}$$

]

Where $[p_l, p_r]$ are random values uniformly distributed between 0 and 1. And the m's and c's are calibration slopes and offsets respectively. Next:

[

$$s_l r = l_c + r_c \tag{3}$$

]

If this value is too small, smaller than $[s_l rmin]$ the computed position is considered invalid. If the position is valid, the final value x is computed via: [

$$x = m_x \frac{(r_c - l_c)}{s_l c} + o_x \tag{4}$$

where $[m_x, o_x]$ are additional linear calibrations for the final position.
The SpecTcl variables are as follows:

Table 2: SpecTcl variables used in position calculations

equation variable	SpecTcl Variable
$[m_l]$	ppac.x.scale1
$[m_r]$	ppac.x.scale2
$[c_l]$	ppac.x.offset1
$[c_r]$	ppac.x.offset2
$[m_u]$	ppac.y.scale1
$[m_d]$	ppac.y.scale2
$[c_u]$	ppac.y.offset1
$[c_d]$	ppac.y.offset2
$[s_l rmin]$	ppac.x.minsum
$[s_u dmin]$	ppac.y.minsum
$[m_x]$	ppac.x.slope
$[o_x]$	ppac.x.offset
$[m_y]$	ppac.y.slope
$[o_y]$	ppac.y.offset

PPAC Profile Spectra The PPAC profile spectrum is essentially a compressed PPAC position spectrum. For each position axis, the position value (e.g. $[x]$) is multiplied by a constant $[p_s cale]$ and then histogrammed. The two scale variables are called: ppac.x.profilechans and ppac.y.profilechans respectively.

Scaler Trendline Spectrum Scaler trendline spectra show the time evolution of the count rates in the scintillator scalers. The periodic scaler increments are summed into a channel for a fixed dwell time that can be set. When time of the sum exceeds the dwell time, either the next channel is selected or, if the previous channel was the last channel, the spectrum is shifted left one channel and the last channel is used again.

Note that the scaler trendline spectra are destroyed and created at the beginning of each run, in order to allow you to change the number of channels of trend information maintained. All four trend spectra are controlled, therefore,

by a channel count and a dwell time. The channel count is the TCL Variable see.trend.Channels. The dwell time (in seconds) is see.trend.SecondsPerChannel.

2.4.4 Using Xamine

This section describes how to use Xamine, the NSCL Spectrum Display program. It is beyond the scope of this document to describe all of the features of the Xamine program. A full user manual is available at: [tp://ftp.nscl.msu.edu/pub/daq/doc/xamine_user.ps] [ftp://ftp.nscl.msu.edu/pub/daq/doc/xamine_user.ps] (warning, this manual is about 130 pages long).

In this section you will learn:

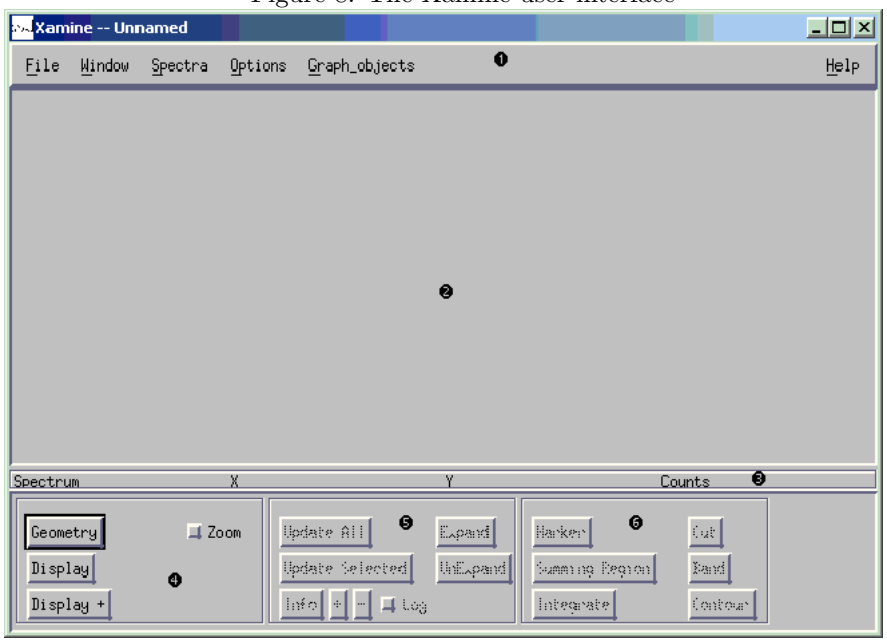
- To recognize the segments of the Xamine user interface.
- How to subdivide the display area so that several spectra can be displayed.
- How to display a set of spectra in the display area.
- How to set the display characteristics of a spectrum.
- How to set a summing region and integrate it.
- How to print spectra.
- How to save and restore a setup so that it can be re-used.

The Xamine User Interface The figure below shows the Xamine user interface. The window can be stretched as desired to provide more spectrum display room.

As you read the text below, note that the numbers refer to the numeric labels on the figure.

1. The menu bar provides access to almost all of Xamine's capabilities.
2. The display area is where spectra will be displayed.
3. The status bar provides several pieces of information. The Spectrum section identifies which spectrum is selected. The X Y areas tell you where in the spectrum the mouse is. The Counts area displays the number of spectrum counts at the mouse's position.
4. This set of shortcut buttons controls what will be displayed in the display area. The buttons in this box provide shortcuts to some of the more frequently used options in the Window and Spectra menus.
5. The buttons in this box control what a spectrum looks like. These buttons provide shortcuts to some of the most frequently used options in the Spectrum menu and the spectrum properties dialog.
6. This set of short cut buttons allows you to create various graphical objects and to perform integrations. Most of these buttons don't have corresponding menu entries.

Figure 8: The Xamine user interface



Take some time to explore the Xamine GUI as the software has quite a bit of functionality that we will not describe in this document.

Subdividing the display Xamine can display up to 100 spectra. This is accomplished by:

1. Using the geometry button to subdivide the display area into a rectangular array of row and columns.
2. using the Display and Display+ buttons to place spectra into the display area.

Click on the Geometry button (note that any spectra that are displayed are removed from the display whenever the geometry is changed). You will see the following dialog. Note that all Xamine dialogs are non-modal, if you have the screen space and will be interacting with a dialog several times feel free to leave it posted on the screen.

To set a geometry, use the radio buttons to select the number of rows and columns you want. Click ok to accept and dismiss the dialog (click apply to accept and keep the dialog posted). click Cancel to do nothing.

The SpecTcl display area is now divided as you have requested. Each box in the display area will be referred to as a *pane*. One pane is always rendered to look like it is pressed in. This pane is called the *selected pane*. All operations on single spectra are performed on the selected pane.

Figure 9: The geometry selection dialog

Single clicking on pane selects it. Double clicking a pane selects it and toggles between zoomed and unzoomed display. Zoomed display only displays the selected pane. Unzoomed displays them all. The zoom checkbox can also be used to flip back and forth between zoomed and unzoomed.

Displaying spectra Once you have selected a pane geometry, you will want to display spectra in the panes of this geometry. To do this you will:

- Select the pane you want to "fill."
- Click on either the Display or Display+ buttons to bring up the Spectrum Choice dialog.
- Select a spectrum from that dialog.
- Click Apply or Ok to accept. Note that double clicking the spectrum name also performs the default action (Ok for Display and Accept for Display+).

The differences between the Display and Display+ buttons are:

- The default action for the Display button dialog is Ok, while that of the Display+ button dialog is Apply.
- when the Display+ button's dialog is "applied", the selected pane will be changed. This allows you to quickly fill the panes in a geometry.

Setting the characteristics of a spectrum You will often want to modify the characteristics of a spectrum so that you can see the features you are interested in. This section will describe how to set the characteristics that are available on the short cut buttons.

Updating spectra To update the selected pane, click the *Update Selected* button. To update the contents of all panes, click the *Update All* button. A periodic update rate can be set using the properties dialog in the *Spectra-¿Properties* menu entry and the *Options-¿Update Rate...* menu entry.

Changing the full scale To change the full scale of a spectrum, use the + and − buttons. These buttons rescale the spectrum to move the peak height in the direction implied by the button label. Thus + actually decreases the full scale and − increases the full scale.

Expanding and unexpanding Given a pane that is displaying a spectrum, you can display a region of interest within that spectrum using the *Expand* button. A region of interest is a slice of the spectrum between lower and upper limits for a 1-spectrum. For a 2-d spectrum, a region of interest is the interior of a rectangle defined by two points on one of its diagonals.

Click on the Expand button. Xamine will display the expand dialog.

Click the limits of the region of interest in the spectrum. If you click on a spectrum that is not selected, any points you have acceped are lost and the new spectrum is selected. Click on Ok or Apply to accept your points and display only the region of the spectrum between them.

Once expanded, you can re-display the entire spectrumby clicking on the *Unexpand* button.

Integrating regions of a spectra You can integrate regions of interest on 1 or 2 dimensional spectra. The Xamine nomenclature for a region of interest is *Summing Region*. Summing regions are a kind of *graphical object* that is created on a spectrum pane and remains in place until deleted. Summing regions are associated with a spectrum and will be displayed any time that spectrum is displayed in a pane.

To accept a summing region:

- Click the *Summing Region* button.
- The Grahical input dialog will be displayed.
- Click in the points of the summing region on the spectrum.
- Click ok, or accept, to accept the summing region.

To perform an integration, click on the *Integrate* button. A text box will pop up to show the results of the integrations on all summing regions and elligible gates in the selected spectrum. If you do another integration without

dismissing the text box, the results will be appended to the text box so that you can compare them.

Note that if you eventually decide to use gates, Integrate will also integrate cuts and contours.

Printing spectra Spectra can be printed. Use the *File-¿Print...* menu to bring up the print dialog. Fill it in as desired (you may need to ask NSCL staff which printers are available and best for you). Click ok to dismiss and print.

Saving and restoring your setups Normally you will create several geometries, with different sets of spectra. Each of these (geometry + layout) is called a *Window definition*. For example, you may have one window definition that displays the key ppac spectra, and another that displays the scintillator spectra.

It would be painful to have to re-create each window definition every time you needed it. Therefore, Xamine allows you to save window definitions to file. Saved window definition files can then be re-loaeded.

To save a window definition file:

- Click the *Window-¿Write Configuration..* menu entry.
- The file selection dialog box will pop up.
- Choose the directory and name of the file you want to write to. To overwrite an existing file, simply double click it.
- Click Ok to accept the filename and create the file.

To restore a window definition file:

- Click the *Window-¿Read Configuration..* menu entry.
- The file selection dialog box will pop up.
- Choose the directory and name of the file you want to write to by simply double clicking it.

3 Run Products

ach production run creates several *run products* these run products are files that contain data relevant to the run. These products are created by several of the components in the system. It is best to keep allcomponents running during production runs to ensure that all of your analysis products are created.

In this section you will learn:

- About the NSCL experiment file structure.
- About the run products created by the SEETFDAQ system and which components create them.
- How to add your own run products to the SEETFDAQ system.

3.1 The NSCL Experiment file structure

The NSCL File structure is described fully in http://docs.nscl.msu.edu/daq/overview/standard_nscl_directo

The basic idea is that the system maintains a directory named /experiment/current that contains all data associated with the active run. These data can be file that are created and maintained by the data acquisition system software, or they can be arbitrary files and links that you create.

When a run ends, a new directory /experiment/run*n* is created. This new directory will be a snapshot of all the data in the current directory at the time the run ended.

3.2 Run products placed in experiment/current

The SEETFDAQ system produces several run products. A *run product* is a file, or link, that is placed in the /experiment/current directory. Most of these files are saved by the system in /experiment/run*n* directories.

The following are the standard run products:

runn-4096.evt A file containing a log of all events taken by the system.

ControlInfo.lis A file that contains the values of some of the control system variables at the end of the run. The set of variables that are recorded are listed, one per line, in /config/channels.dat

config A link to /config that causes a snapshot of that directory to be saved. The /config directory contains the entire experiment software configuration.

scint.log A CSV file that provides time stamped scintillator scaler rate values. This file can be imported to e.g. MS-Excel.

see.trend.*.spec The contents of the SEE scaler trendline spectra at the end of the run.

Two other directories just below the experiment home diretory contain other analysis products. These files are created outside the experiment/runn directories due to synchronization issues.

scalerfiles This directory contains the sums of all the scalers over the life of each run. Files are named: run*n*.scalers where *n* is the run number.

savedspectra This directory contains spectra saved at the end of each run in all cases *n* is the run number. Note the ppac spectra are only meaningful on beam quality runs.

see.ppac.x-run*n*.spec - The ppac x position spectrum.

see.ppac.y-run*n*.spec - The ppac y position spectrum.

see.ppac.y_v_x*n*.spec - The ppac x-vs-y position spectrum.

see.sci.counts-run*n*.spec - The 4 channel scintillator sum spectrum.

see.trend.l-run*n*.spec - The left scintillator trend spectrum.

see.trend.r-run*n*.spec - The right scintillator trend spectrum.

see.trend.u-run*n*.spec - The up scintillator trend spectrum.

see.trend.d-run*n*.spec - The down scintillator trend spectrum.

3.3 How to associate data with runs

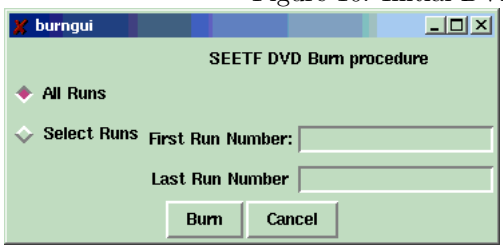
To associate data with a run, copy it, or link it into the /experiment/current directory. For an example that shows how you might do this automatically at the beginning of each run, see the file /ReadoutCallouts.tcl. This file contains the code that creates the ControlInfo.lis file.

4 Writing your data to DVD

To burn your data to DVD login as the experimental user to the data U linux system (useepc2). At a terminal window type:
`burngui`

This will bring up a window that looks like:

Figure 10: Initial DVD Burn dialog.



If you only want to burn some of your runs:

- Select the "Select Runs" radio button.
- enter the *number* of the first run to save.
- optionally enter the *number* of the last run to save. (If omitted, this defaults to the highest numbered run).

Once you accept these settings by clicking the *Burn* button, you will see the Burngui log window. This window displays output that shows the progress the preparation to burn the data to DVD as well as the burn operations themselves.

- First a set of directories are constructed that will appear on the burned disk. At this time, if necessary, the data are divided amongst multiple DVD's.
- Second, ISO image files are built, one for each output DVD.
- You are then prompted to put blank DVD's in the burner as needed to burn the data to each dvd.

When the burn is complete, the Cancel button on the log window will turn into a Dismiss button. Check the log window carefully for errors.

Figure 11: Burn log window

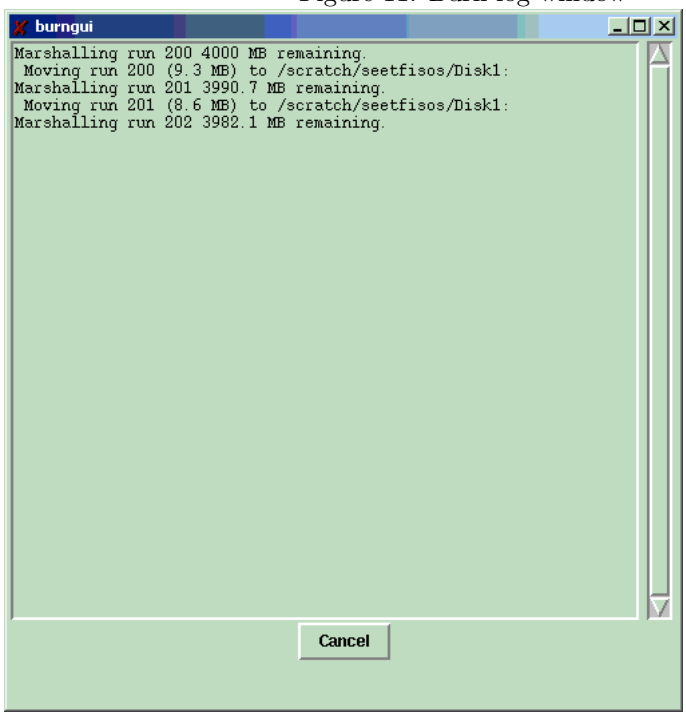


Figure 12: Prompt for a disk