PCA Science Monitoring

Hwa-Ja Rhee

This document illustrates design of the EDS/PCA Data monitoring portion of the XTE SOC's Science Monitoring subsystem.

1.0 Introduction

The PCA Science Monitoring software provides capability of monitoring EDS/PCA science data in various modes. By the time of the XTE launch, which is scheduled on August 31 1995, software for Standard Mode1 (Calibration and Light Curve), Standard Mode2 (Spectra and Rates), Temporal Science (Power and Power History) and Spectral Science (Base Summation, Recommend, Colors vs. Time, and Spectral History) will be developed, emphasizing basic capture and display of the real-time data, and providing very little on analyzing them. Other modes, such as Transparent, Event, Binned, and Single bit mode may be added after the launch. The purpose of this software is to monitor performance of 5 PCUs to insure their acceptable performances.

Although this system is primarily intended for the real-time data monitoring, It has the capability to display archived data also.

2.0 Requirements

Design of this software is directly influenced by the following hand written notes, whose requirements are based on SOC-102 and SOF Requirements document.

Notes on EDS/PCA Science Monitoring for Build 3 --- by Barry Giles, Jan. 1994

2.1 Standard Mode1

Following requirements apply to Standard Mode1:

2.1.1 Light Curves

1. Display Xenon good Light Curve in 125 ms time interval for each detector.
2. Calculate and Display Xenon good Rates summed over all 5 PCUs.
3. Display VLE rate summed over all 5 PCUs.
4. Display rate of good Propane events summed over all 5 PCUs
5. Display all other event rates not counted elsewhere.
6. Add options to compress data into 0.25, 0.5, 1.0, 2.0, 4.0, 8.0, 16.0 sec. bins.
7. Yaxis scale option should include Log scale.
8. Yaxis should be able to plot on the fixed scale to all 5 Xenon Light curve data.
9. Calculate and Display Minimum, Maximum, and Total rate of each Light Curve.

2.1.2 Calibrations

1. Display Calibration spectrum for 3 layers (1 left, 1 right each) for each PCU.
2. An option to select desired detector to be monitored.
3. An option to monitor data in “live” or “Integration” mode.
4. Xaxis should have some limited “Zoom” and “unZoom” capabilities.
5. Yaxis scale option should include Log scale.
6. Yaxis should be able to plot on the fixed scale to the pre-defined group of data.
7. Calculate and Display Area under each energy histogram.
8. Regroup energy channels into 256 from 129.

2.2 Temporal Science

Leahy method is used for FFT normalization.

2.2.1 Power

1. FFT the data to produce 513 point data sets from 1024 point light curve data.
2. Log scale should be the default scale in Yaxis.
3. Xaxis scale in ‘bins’, and X2axis in ‘Hertz’ with maximun=4 Hz.
4. Include option to ‘integrate and average’ with ‘start’, ‘stop’, and ‘clear’ functions.

2.2.2 Power History

1. FFT the 1024 light curve data before co-adding and plotting.
2. Ability to take 1, 2, 4 or 8 FFT data sets and average before plotting.
3. Ability to redraw the contour area after altering the height options.
4. FFT 'height' shown in color scale will need to be on a log or linear scale.

2.3 Standard Mode2

Following requirements applies to Standard Mode2:

2.3.1 Xenon and Propane Energy Spectra

1. Display Energy spectrum of each layer of each detector in Energy Channel vs. Counts.
2. An option to select desired detector to be monitored, including Sum of all 5 detectors.
3. An option to monitor data in "live" or "Integration" mode.
4. Xaxis should have some limited "Zoom" and "unZoom" capabilities.
5. Yaxis scale option should include Log scale.
6. Yaxis should be able to plot on the fixed scale to the pre-defined group of data.
7. Calculate and Display Area under each energy histogram.
8. Calculate and Display Sum of 3 layers (1 left, 1 right each).
9. Calculate and Display Ratio of 3 layers (L1 + R1 + L2 + R2 - 2(L3 + R3)}
10. Regroup energy channels into 256 from 129.

2.3.2 Background Rates - 29 parameters.

1. Display up to 10 user selected Background Rates from the list
2. Data are accumulated in the array, so a plot length of 512 points will show ~2 hours of data
3. Yaxis scale option should include Log scale.
4. Calculate and Display Minimum, Maximum, and Total rate of each Rate Curve.
5. An option to select desired detector to be monitored.
6. Option to show data gaps.

2.4 Spectral Science Mode

Following requirements are common to all options of the Spectral Science Mode, a variation of Standard Mode2.

1. Display Sum of Energy spectrum from all layers and all 5 PCUs in Energy Channel vs. Counts.
2. An option to calculate sum of layers of each detector.
   L1 + R1 + L2 + R2 + L3 + R3 or
   L1 + R1 + L2 + R2 - 2(L3 + R3)

3. An option to use Raw data or Gain/Offset corrected data when adding spectra from 5 detectors.
4. An option to monitor data in “live” or “Integration” mode.
5. Xaxis should have some “Zoom” and “unZoom” capabilities.
6. Yaxis scale option should include Log scale
7. Calculate and Display Area under each energy histogram.
8. Calculate and display statistics in Total counts or in Counts/sec.
9. Regroup energy channels into 256 from 129.
10. Calculate keV using energy scale information and draw a second Xaxis.

2.4.1 Base

No additional requirements to the common Spectral Science Mode.

2.4.2 Recommend

Additional requirements are:

1. Spectrum is displayed in pre-defined energy bands:
   Channel # 0 - 13, Channel # 14 - 17, Channel # 18 - 23,
   Channel # 24 -35, Channel # 36 - 49, Channel # 50 - 249
2. The channel boundaries should be shown in keV and bin numbers.
3. Horizontal axis in keV and bin numbers.

2.4.3 Colors vs. Time

Additional requirements are:

1. Add flexibility of sliding bar to select 2 color energy bands or 3 color energy bands.
2. Calculate ratios of the each energy bands and display them in strip chart form.
3. The channel boundaries should be shown in keV and bin numbers.
4. Horizontal axis in keV and bin numbers.
5. The slider values to be shown in keV and bin numbers

2.4.4 Spectral History

Additional requirements are:

1. Display spectral elements in color coded spectral “heights”.
2. Display spectrum intensity in strip chart plot.
3. Use log scale for height to colors.
4. Show history of up to 200 spectral data.
5. Vertical axis in keV and bin numbers.
6. Add options to average (every data, 2, 4, or 8 sets of data) spectra before plotting.
7. Include option to show data time gaps

3.0 Operating Scenarios

During the mission, certain modes of EDS/PCA science data, indicative of performance of the detectors, will be constantly monitored by a Duty Scientist or a Mission Operator under the SOF operational environment. Also, EDS or PCA instrument team may monitor the data independently in their science facility. In the real-time monitoring mode, SOF's *data ingest* subsystem invokes *rtserver* to send packets to *rtclient.* It is *rtclient* process that receives packets, checks EDS configuration, if appropriate, assembles packets into partitions, which are then read into a *Xclient*, a X Window process with Graphical User Interface, via a standard UNIX IPC mechanism (a pipe or a socket). Finally, *Xclient* program unpack partitions, and display data in interest, after various conversions and calculations of the data.

For the off-line processing (stored data monitoring), Xclient deals with the archived packet data files, not the *DataSet(s)* created by the *Data Management* subsystem.

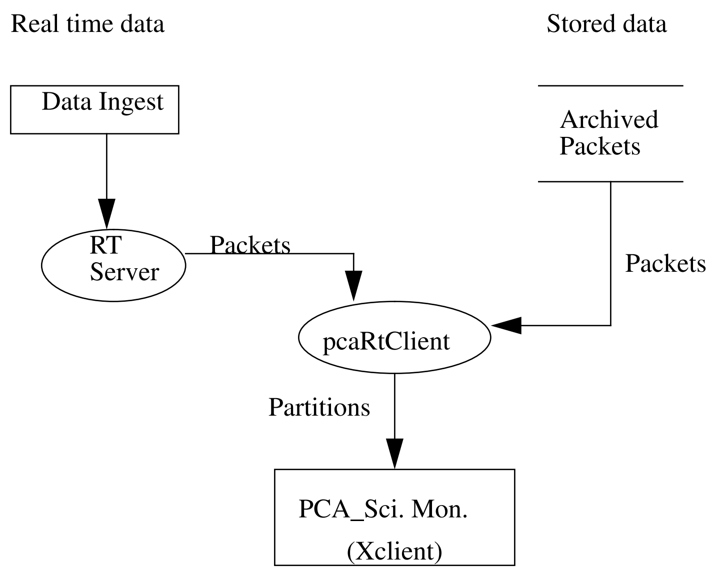
In both operation modes (real-time and off-line), all operations in the *Xclient*, such as selecting a detector, selecting a mode (*live* or *integration*), or axis scaling options, are interactive via buttons in the menubar, created by TAE+. (see *PCA Science Monitoring User's Guide* for detail)

4.0 Open issues

- Various EDS modes to be monitored are not clearly defined yet.
- Distribution of duties between SOC, MIT, and PCA is not clearly defined or agreed upon by all parties involved yet, although some are obvious.
- Do we need 3D plotting? Since Athena widget plotting package doesn't have 3D capabilities, does this mean that we need different X-window plotting package?
- For analysis, which will be used? XMGR or IDL?
- Should Science Monitoring accommodate stored data from Data Management? In other words, should it be able to handle partition objects? Where do we draw a line between SOF and GOF activities?

5.0 Design Context

The following diagram shows the context of the PCA Science Monitoring software system. In real-time monitoring, *pcaRtClient* receives selected packets and generates unpacked partition data stream which is read into the individual *Xclient* program for each EDS mode. *pcaRtClient* program was assembled initially by the SOF Science Monitoring Subsystem Leader, from the initial sets of Rt-clients for EDS/PCA, provided by the MIT. For stored data monitoring, archived packet datasets will be used. PCA Science Monitoring system will consist of, multiple *Xclient*s, one for each EDS mode, and *a pcaRtClient*. It will reside as a component of the SOF's Science Monitoring Subsystem.



6.0 Interfaces

The following indicate interfaces with SOC subsystems and other Instrument Teams.

6.1 SOC Subsystems

- Science Monitoring - Parent subsystem of the PCA Science Monitoring. *SciCommon* library contains many common routines all three instrument teams may need.
- Utilities - *Util* library contains routines to handle Input and Output management through pipes or sockets.
- Data Ingest - starts rtserver which communicates with rtclient.

6.2 Other Instrument Team Software

MIT- supplies rtclient programs, one for each EDS mode.

6.3 External Software

- Rogue Wave Class Library
- TAE+
- Athena Widget plotter

7.0 Processes

- *pcaRtClient* - As a counterpart program of the *rtserver,* this program accepts real time packets from real time server, check Configuration ID, generates partitions, and feed the partitions to the appropriate Xclient programs. TCP/IP network protocol is used.
- *pcaSM* - Top level GUI for the entire system. All Xclients can be invoked from this panel.
- *pcaSMaux* - This program contains late data popup window and MET monitoring strip chart window. This process may be invoked from any of the Xclients listed below.
- *pcastd11LC* - Xclient program for Standard Mode1 light curves.
- *pcastd1CalS* - Xclient program for Standard Mode1 calibration spectra.
- *pca1FFT* - Xclient showing Power. (used Standard Mode1 data as input) Leahy method is used for normalization.
- *pcaStd1LeahyHist* - Xclient for Power History.
- *stdMode2 / pcastd2* - Xclient program for Standard Mode2 spectra.
- *pca2Rate* - Xclient program for Standard Mode2 background Rates.
- *pcaSBase* - Xclient for Spectral science - Base summation.
- *pcaSColor* - Xclient for Spectral science - Colors vs. Times.
- *pcaSRecom* - Xclient for Spectral science - Recommend.
- *pcaSHist* -Xclient for Spectral science - Spectral History mode.

8.0 Subsystem Design

Figure 1 defines object classes of this system, and Figure 2 shows the functional flow diagram. One of the Xclients (Standard Mode2 spectra) of the PCA Science Monitoring system was used as an example, since all Xclients have similar structures, and functionalities of each component are similar also.

.

FIGURE 1. PCA Science Monitoring Object Classes and Routines (Xclient example - stdMode2)

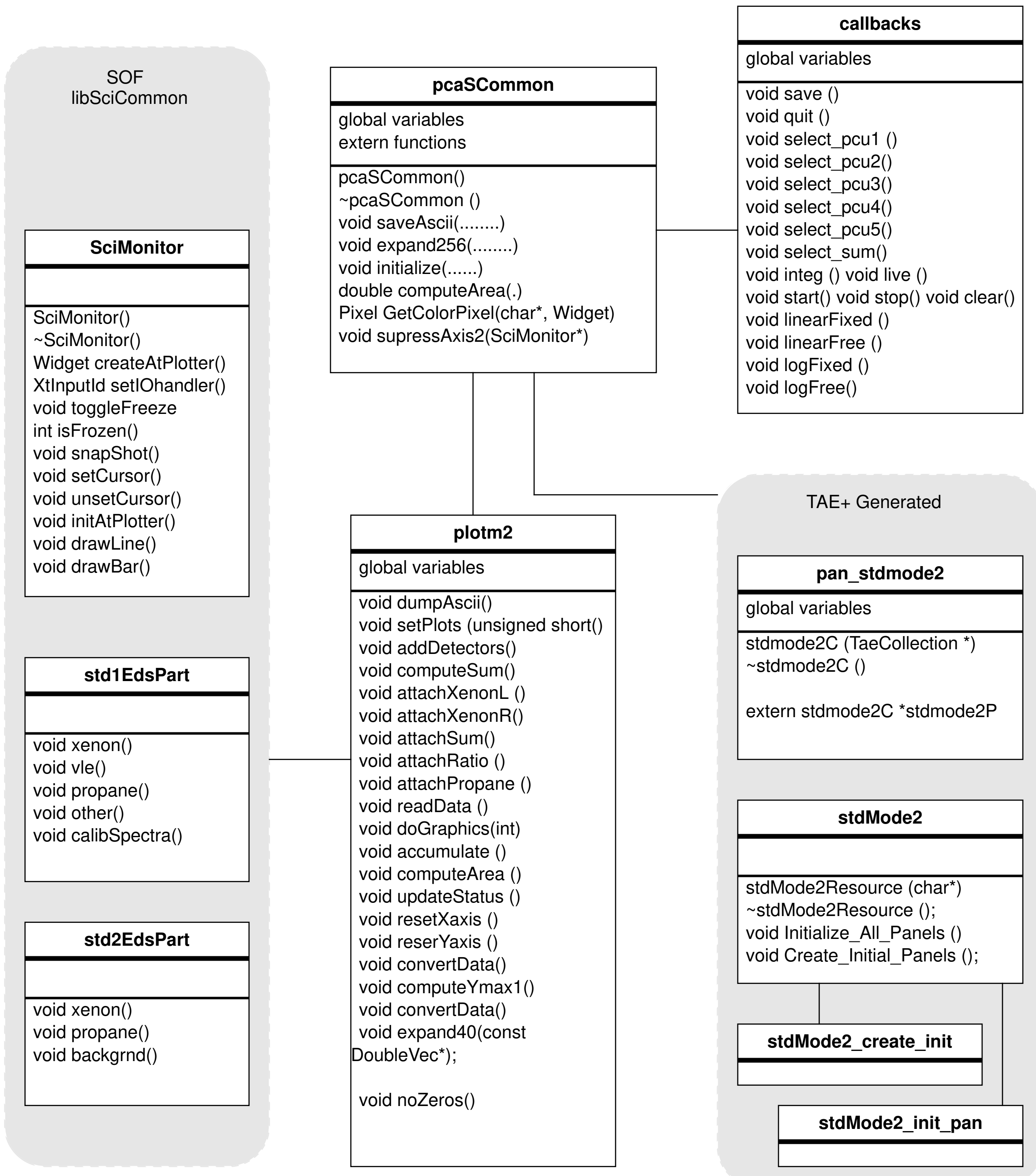
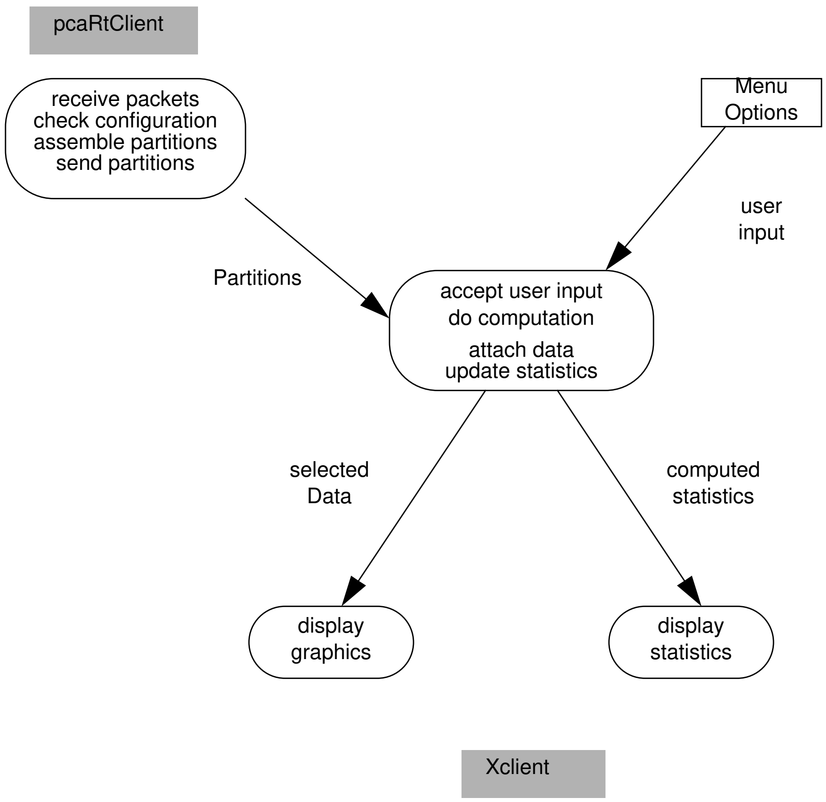


FIGURE 2. PCA Science Monitoring Functional Flow



pcaSCommon

This class contains several methods which maybe shared by all PCA Science Monitoring Xclient programs. There are no true class attribute variables in this class. Since this class serves like a common header among the files in the Xclient program, lots of global variables (logical or integer, mostly) and external functions are defined here. Actual codes of these functions are found in graphics.C/plot*.C file in each Xclient program directories.

To use methods this class, insert following line in your file/class.

#include "pcaSCommon.h"

Attributes

None

Operations

```
void saveAscii (const char* fileName, DoubleGenMat& dgm,
        int ndata, int ncol, char* status, double time, int
        mode, const String yname[]=NULL)
```

Creates a file containing ascii dump of the data you see on the screen plot(s)

Creates one file for each Xworkspace plot.

```
void expand256 (DoubleGenMat& expandXenon, const DoubleVec&
        v, int n)
```

Expand Standard Mode2 Xenon spectra from 129 channels to 256 channels.

```
void initialize(DoubleArray& xL, DoubleArray& xR)
```

Initialize vector array elements to zeros. This is called to clear the counter array at the beginning of the integration.

```
double computeArea (const DoubleGenMat& exXenon, int n)
```

Computes total counts of spectrum.

```
Pixel GetColorPixel (char* colorstr, Widget w)
```

Returns color Pixel value equivalent to the color name string.

```
void supressAxis2(SciMonitor* monX)
```

Do not draw X2axis and Y2axis of the monX.

callbacks.C

This file is a collection of callback routines that accommodate TAE+ generated GUI (Graphical User Interface) functionalities. This is in ANSI C and not a C++ class.

Attributes

`Global variables contained in pcaSCommon.h.`

int selYaxis, selMode, selDet, selInteg, etc.

Operations

`void clear()`

Reset accumulation Counter and Integration Arrays to zeros.

`void linearFixed ()`

Sets global variable selYaxis = 0 (this is default).

`void linearFree ()`

Sets global variable selYaxis = 1.

`void logFixed ()`

Sets global variable selYaxis = 3.

`void logFree ()`

Sets global variable selYaxis = 2.

`void live ()`

Sets global display mode variable selMode = 0.

`void integ ()`

Sets global display mode variable selMode = 1.

`void quit ()`

Terminates from current process.

`void save ()`

Saves current state.(not implemented yet)

`void select_pcu1`

`void select_pcu2`

`void select_pcu3`

`void select_pcu4`

`void select_pcu5`

`void select_sum`

Sets selDet appropriately and call doGraphics(False).

`void start()`

Sets selInteg = 1. (later this is used to start accumulation)

`void stop()`

Sets selInteg = 0. (later this is used to stop accumulation)

plotm2.C (or graphics.C)

This file is a collection of callback routines that accommodate TAE+ generated GUI (graphical User Interface) functionalities. This file is written in ANSI C and not a C++ class.

Attributes

```
Global variables in pcaSCommon.h
DoubleArray xenonCountL(PCU_COUNT,3, std2XVals)
DoubleArray xenonCountR(PCU_COUNT,3,std2XVals)
DoubleGenMat propaneCounts(PCU_COUNT,std2PVals)
DoubleVec sumCounts(std2XVals)
DoubleVec ratioCounts(std2XVals)
DoubleVec xvals(fullXVals)
DoubleVec xvalpr(fullPVals)
DoubleArray xenonIntegL(PCU_COUNT,3, std2XVals)
DoubleArray xenonIntegR(PCU_COUNT,3,std2XVals)
DoubleGenMat propaneInteg(PCU_COUNT,std2PVals)
DoubleVec sumInteg(std2XVals)
DoubleVec ratioInteg(std2XVals)
```

Operations

`void dumpAscii (const char *fileName)`

Writes Ascii file containing plot data.

`void setPlots(unsigned short* theData)`

Converts unpacked data from integer*2 to double to prepare for Athena Plotter.

`void addDetectors()`

Calculates sum of all 5 detector data.

`void computeSum()`

Computes sum and ratio of 6 layers.

sum = L1 + R1 + L2 + R2 + L3 + R3

ratio = L1 + R1 + L2 + R2 - 2(L3 + R3)

`void computeArea()`

Computes total counts under the graphs

`void initialize()`

Initializes integration counters and arrays.

`void accumulate()`

Accumulates data into the integration arrays.

`void resetXaxis()`

Resets maximum number of data points to be plotted.

Applies to “zoom” and “ZOOM”.functionalities.

`void resetYaxis()`

Resets Athena Widget Yaxis properties appropriately....such as LinearFixed (default), LinearFree, LogFixed and LogFree.

`void convertData()`

Expands spectra data from original 128 channels to 256 channels (xenon)

Expands propane spectra from 32 channels to 40.

`double computeYmax1()`

Computes ymax for both L1 and R1.

`double computeYmax2()`

Computes ymax for both L2 and R2.

`double computeYmax3()`

Computes ymax for both L3 and R3.

`double computeYmaxp()`

Computes Ymax of propane spectra.

`double computeYmaxs()`

Computes ymax of sum spectra of all layers.

`double computesYmaxr()`

Computes ymax of ratio spectra of all layers.

`void attachXenonL()`

Attaches 3 left layers of Xenon data. (Athena plotter routine)

`void attachXenonR()`

Attaches 3 right layers of Xenon data. (Athena plotter routine)

`void attachSum()`

Displays summed Xenon data of all layers.

`void attachRatio()`

Displays Xenon ratio data of all layers.

`void attachPropane()`

Displays Propane data.

`void readData()`

Reads data, computes sum and area, set axis properties and display graphs.

`void doGraphics (int init)`

If init = True, initialize all Athena Plotter properties for Xworkspace graphics display. initialize SciMonitor Constructor.

If init = False, simply display the data selected.

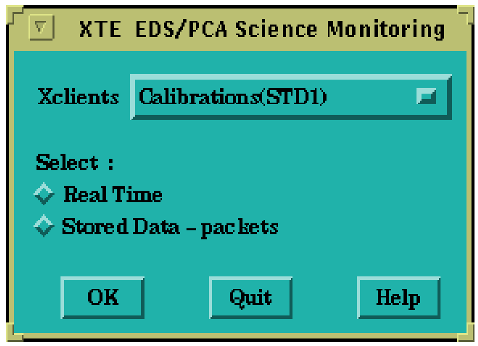
`void updateStatus()`

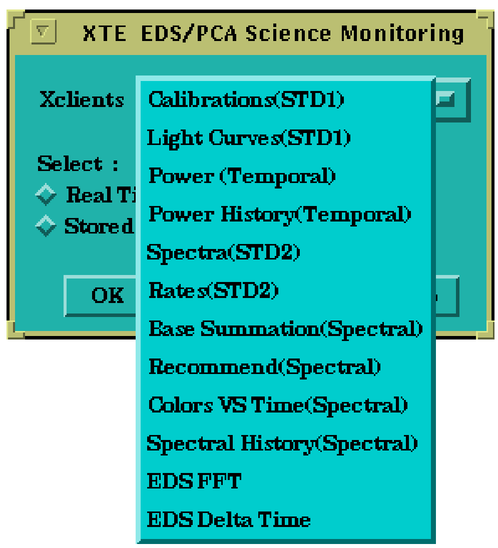
Updates Dynamic Text fields with calculated statistics information.

`void expand40 (const DoubleVec * invec)`

Uncompress Propane spectra from 32 to 40 channels.

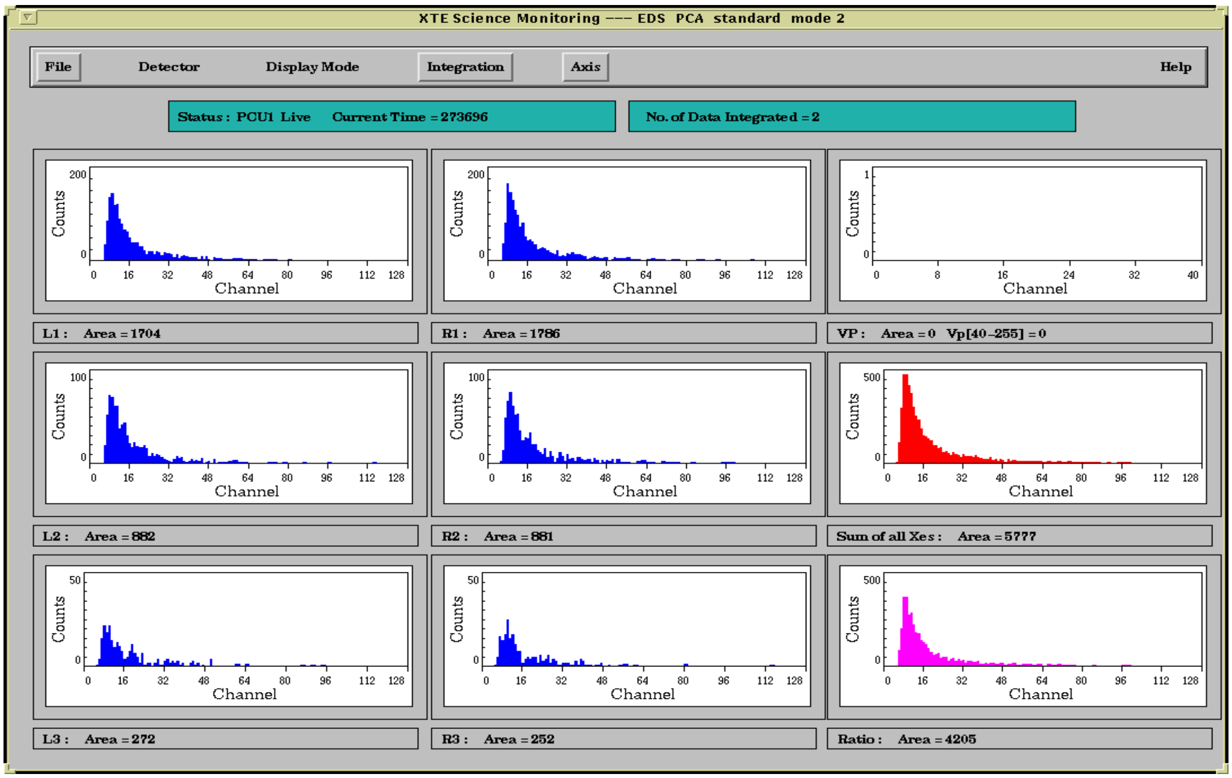
9.0 Top level Graphical User Interface





10.0 Xclients Graphical User Interface

FIGURE 3. Xclient Graphical User Interface



11.0 Common Features

11.1 libSciCommon

This Science Monitoring Common Library, developed by the SOF subsystem leader, supports all instruments science monitoring processes (Figure 1 shows the list of global methods contained in this library).

11.2 TAE+

Graphical User Interface development package used. Current (July, 94) version is 5.3.

11.3 Athena Plotter Package

The Athena Plotter Widget package is used to provide the graphics capabilities by drawing into a TAE+ “XWorkSpace” widgets.

11.4 Rogue Wave Class library

Classes in this library are used for software development, especially for array manipulations and iostream functions.

11.5 pcaRtClient

12.0 Comments and Responses

This section is reserved for Comments and Responses.

Comment 1: Comment 1 will be added here.?

Comment 1 will be answered here.

Comment 2: Comment 2 will be added here.?

Comment 2 will be answered here.