NATIONAL
OPTICAL
ASTRONOMY
OBSERVATORY

MAJOR INSTRUMENTATION GROUP
950 N. Cherry Ave.
P. O. Box 26732
Tucson, Arizona 85726-6732
(520) 318-8000 FAX: (520) 318-8303

MONSOON

PAN Communications

Command/ Response and Data Stream Interface Description

NOAO Document ICD 5.0

Revision: 1.0

Provisional

This document may or may not include all pertinent information. The final released version is forthcoming.

Authored by:
Nick C. Buchholz and Phil N. Daly
1/7/2002
Please send comments:
nbuchholz@noao.edu

Revision History

Version	Date Approved	Sections Affected	Remarks
0.1.1	3/8/2002	All	Second release draft
0.1.2	4/29/2002	All	Restructured and changed nomenclature - ncb
1.0 (Provisional)	8/15/2006	All	Reformatted and edited - aro

Table of Contents

List of Figures

List of Tables

1.0 Introduction

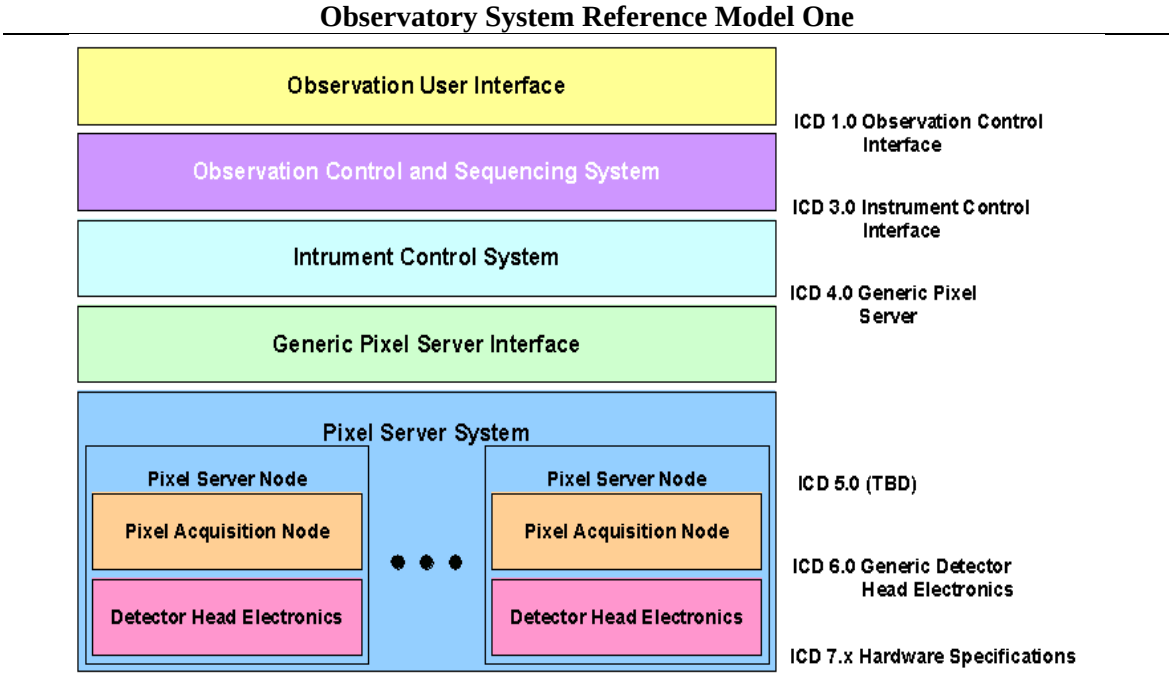
1.1 Scope

The Generic Pixel Server (**GPX**) Interface, as discussed at the Nov. 2001 ACCORD conference, describes an interface between an image acquisition system or pixel server and the external world. The definition describes the interface between the instrument or observatory control system and the electronics and software that configure the detector, generate and preprocess the data, and send the data to an archive for later analysis.

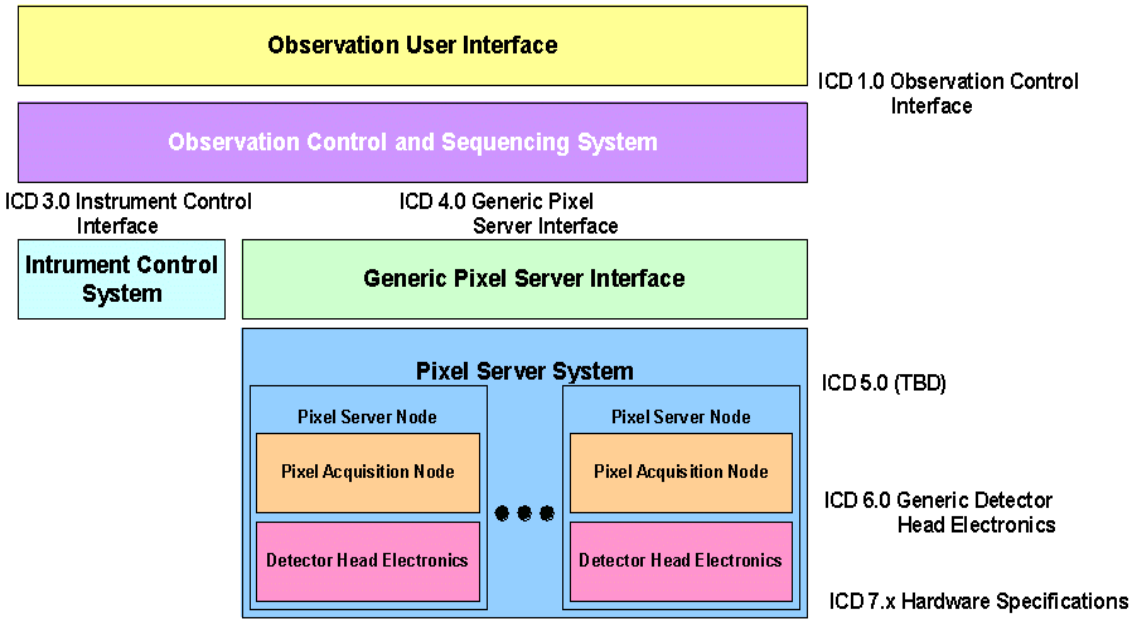
The PAN Communications and Command ICD is a subset of the PAN commands which simplify the decomposition of the PAN in cases where more than one system communicates with the low level pixel generation hardware, namely in mosaic requiring multiple detector controllers. This document delineates the interface from the **GPX** to a PAN.

The **PPX** interface provides for access and control of the data generation and capture hardware. It combines the functionality inherent in the detector head electronics (detector controller) and data preprocessor node. This ICD presents a common interface between the upper levels of a **GPX** system and the lower level hardware Also note that this interface is between two software systems. No attempt has been made to make the commands easy to type or convenient for a human user. The choice of ASCII strings was purely to provide an easy path for debugging and diagnosis of problems.

This document is the interface control document for the PAN Interface. This document contains a communications and connection specification, a command and response specification and a pixel and status data stream proposal.



Observatory System Reference Model One
Figure 1



Observatory System Reference Model Two
Figure 2

1.2 Purpose

This Interface Control Document (ICD) serves three purposes:

- To describe the nature of the communications interface between a **GPX** system and an individual PAN.
- To describe the parameters that will pass between the **GPX** and subordinate PANs.
- To describe the behaviour of the Command/Response interface between a G**PX**-compliant Pixel Server System and the PAN or PANs subordinate to it.

The intended audience for this document is:

- The NEWFIRM instrument group
- The developers of any instrument that plans to use a MONSOON Pixel Server System
- The MONSOON Pixel Server System group

1.3 Acronyms and Glossary

1.3.1 Abbreviations and Acronyms

AC	Acquisition Camera
ADC	Analog to Digital Converter
DAC	Digital to Analog Converter
DCS	Detector Controller System (software)
DHE	Detector Head Electronics
DHS	Data Handling System
ECS	Enclosure Control System
ES	Embedded System
FITS	Flexible Image Transport System
FP	Focal Plane
FPA	Focal Plane Array
GPX	Generic Pixel Server
IAS	Image Analysis System
ICS	Instrument Control System
IDPS	Image Data Preprocessor System
ID	Identifier
IR	Infrared
LAN	Local Area Network
MONSOON	Not an acronym
NICD	NOAO Interface Control Document
N/A	Not Applicable
OCS	Observatory Control System
PDF	Parameter Description File
ROI	Region of Interest
SUS	Status Update System
TBD	To Be Decided

1.3.2 Glossary

Attribute	An entity that describes some aspect of the configuration of a Detector Head Electronics, Image Acquisition System or science instrument. Examples are: the name of a filter, a DAC voltage value, or the tilt angle of a grating. Some attributes will be used by the Instrument Control System as command parameters. The OCS communicates with a science instrument by sending it sets of attributes and values.
Byte	8 bits.
Command	An instruction requiring a system to start some action. The action may result in a voltage changing or some internal parameters being set to particular values. A command may have command parameters (arguments) that contain the details of the instruction to be obeyed.
Data Array	The data, while it is stored in data processing memory, which resulted from one or more readouts of an IR array or CCD detector.
Data Set	A self-contained collection of data generated as a result of a Pixel Server obeying a **gpxStartExp** command. Each **gpxStartExp** command results in one and only one data set.
Detector Head Electronics	The lowest level hardware system, normally closely connected to the detector and the dewar in which the detector resides. Sometimes referred to as the Detector Controller.
Exposure	The process and the data resulting from the process of resetting or clearing a detector, exposing it to photons and then reading one or more frames to determine the photon levels. These frames are processed into a data array, called an exposure, which may be further processed. For example, an exposure would be the data array that results when a single Reset-Readout-Integrate-Readout cycle is performed on an IR detector or a single CCD Clear-Integrate Readout cycle.
Frame	The result of a single readout of an array. Each frame represents the signal values obtained from reading the entire ROI being read out of the detector. Multiple frames may be processed into a single exposure.

1.3.2 Glossary (Cont.)

Generic Pixel Server (GPX)	A set of software routines that implement a pixel server. A Generic Pixel Server does not require any particular set of proprietary hardware or software.
Image	The array of detector pixel and description data representing a science or diagnostic exposure or combined set of exposures. An image is capable of being displayed or processed as a discrete entity. The values in the array may be stored in memory or on disk and are related to the data taken by the detector by some processing algorithm. For example, an image may consist of all the co-added and averaged exposures in one beam of a chop mode **gpxStartExp** command.
Image Acquisition System	A system of software and hardware capable of producing images from a detector on command.
Image Server	A system of software and hardware capable of producing images from a detector on command.
Instrument Control System	A set of software routines designed to control and configure a science instrument to take science observations.
MONSOON Image Acquisition System	A Generic Pixel Server. An extensible, modular Image Acquisition System. The design of the system is, to the extent possible, independent of the hardware being used in a particular implementation. Each component of the system should be capable of replacement by a similar component without having to redesign the rest of the system. Each component of the software is, as far as possible, independent of the underlying hardware and as modular as possible.
Observation	The process of exposing the detector to photons through the telescope in one or more exposures. The result of an observation is a image.
Pixel Acquisition Node	The computer that handles the interface to the Detector Head Electronics and the image pre-processing of the data stream from the Detector Head Electronics.
Pixel Server	A set of software and hardware that accepts commands and produces a set of pixels (an image) related to those commands.
Read	When used as a noun to describe instrument data, a single read of a pixel on the detector. A read may consist of several A/D conversions of the pixel data that are averaged or processed in some other way to produce a single integer output value for the pixel. A readout is made up of one read of each pixel in the detector ROI being read.

1.3.2 Glossary (Cont.)

Readout	When used as a noun to describe instrument data, a single read of every pixel on the detector. A frame is made up of one or more readouts averaged pixel by pixel.
Region of Interest (ROI)	A sub-array of the available detector area. There are two types of sub-arrays that can be defined. The Sequence ROI is an ROI on the active surface of the array used to increase the frequency of the array readout. The Data Reduction ROI is an arbitrary rectangle of any size that fits on the array. Data Reduction ROIs are defined to reduce the volume of data sent to the disk or DHS even when the entire array is being read out.
Value	The value associated with an attribute.
Word	Four bytes or 32 bits.

1.3.3 Reference Documents

SPE-C-G0037, “Software Design Description”, Gemini 8m Telescopes Project.

WHT-PDF-1, “FITS headers for WHT FITS tapes”, Steve Unger, Guy Rixon & Frank Gribbin, RGO.

NOST 100-1.0, “Definition of the Flexible Image Transport System (FITS)”, NASA Office of Standards and Technology.

GEN-SPE-ESO-00000-794, “ESO Data Interface Control Document”, Miguel Albrecht, ESO.

IEEE Std 610.12-1990 - “IEEE standard glossary of software engineering terminology”, Standards Coordinating Committee of the IEEE Computer Society, USA, 19901210

ANSI/IEEE Std 754-1985 - “IEEE Standard for binary floating-point arithmetic” - Standards Committee of the IEEE Computer Society, USA 19850812

xxxx “XDR - Extended data representation Standard” ????

NOAO Document MNSN-AD-01-0002 - ICD 4.0 Version 1.0 - Generic Pixel Server, Communications, Command/Response and Data Stream Interface Description”, Nick C. Buchholz (NOAO), Barry M. Starr (NOAO), 8/8/2006

1.4 Standard Terminology

To avoid confusion and to make very clear what the requirements for compliance are, many of the paragraphs in this standard are labelled with keywords that indicate the type of information they contain. The keywords are:

- RULE
- RECOMMENDATION
- SUGGESTION
- PERMISSION
- OBSERVATION

These keywords are used as follows:

RULE

<Paragraph Number> Subject Describing Text **RULE**

RULEs form the basic framework of this draft standard. They are sometimes expressed in text form and sometimes in the form of figures, tables or drawings. All RULEs shall be followed to ensure compatibility between components. All RULEs use the “shall” or “shall not” words to emphasize the importance of the RULE.

Example:

3.5 Status and Data Stream Interface **RULE**

RECOMMENDATION

<Paragraph Number> Subject Describing Text **RECOMMENDATION**

Wherever a recommendation appears, designers would be wise to take the advice given. Doing otherwise might result in some awkward problems or poor performance. It is possible to design a system that complies with all the RULEs but has poor performance. Recommendations found in this standard are based on this kind of experience and are provided to designers to speed their traversal of the learning curve. All recommendations use the “should” or “should not” words to emphasize the importance of the recommendation.

Example:

2.5.1 GPX Names **RECOMMENDATION**

SUGGESTION

<Paragraph Number> Subject Describing Text **SUGGESTION**

A suggestion contains advice that is helpful but not vital. The reader is encouraged to consider the advice before discarding it. Some design decisions that should be made are difficult until experience has been gained. Suggestions are included to help a designer who has not yet gained this experience.

Example:

2.5.2 Long Variables Names

PERMISSION

<Paragraph Number> Subject Describing Text **PERMISSION**

In some cases, a RULE does not specifically prohibit a certain design approach, but the reader might be left wondering whether that approach might violate the spirit of the RULE or whether it might lead to some subtle problem. Permissions reassure the reader that a certain approach is acceptable and will cause no problems. All permissions use the "may" word to emphasize the importance of the permission.

Example:

2.6 Long Variables Names **PERMISSION**

OBSERVATION

<Paragraph Number>Subject Describing Text **OBSERVATION**

Observations do not offer any specific advice. They usually follow naturally from what has just been discussed. They spell out the implications of certain RULEs and bring attention to things that might otherwise be overlooked. They also give the rationale behind certain RULEs so that the reader understands why the RULEs shall be followed.

Example:

2.7 Long Variables Names **OBSERVATION**

1.5 What's in This Document

This document presents a communications interface, a command/response set, a data stream interface and an error recovery protocol, which represents the **PAN**'s interface to the **GPX**-based MONSOON Supervisory Layer. The interface allows the MSL to command and control a **PAN or PANs** in a consistent manner. The responses to commands are handled as free-form strings except that each response string must start with "OK" or "ERROR" to indicate the status of the command being responded to.

The **PAN** or **PPX** command set is a subset of the **PPX** command set. In addition a **PPX** server can receive **PPX** commands and deal with them in a consistent manner. Each command description contains a section explaining how the **PPX** command routines handle a received **PPX** command.

1.6 What's NOT in this Document

The responses and internal behaviour of the **PPX** command set in the face of errors is left undefined We assume that each command issued will generate a *single* ASCII response and that at a minimum the **PPX** commands will return "OK" for a successful command and "ERROR" if a command fails. In the case of configuration and sequence commands whether the **PPX** halts on the first error or attempts to continue will be left for later discussion by system developers.

No assumption is made about the internal communications protocols or the nature of the internal command passing techniques. In particular, the hardware interconnects and data passing techniques within a PAN are explicitly excluded from this discussion.

1.7 Other Assumptions

An underlying assumption in all of this is that at its heart what we do in astronomy is very similar to the data taking tasks in other areas. That is we *configure and arm* an exposure; we *trigger* or *respond to an external trigger* to start an exposure, we *capture and process* the data from that exposure and we verify that the configuration is as requested. The main difference is that our data sets are large in comparison to most other applications.

It is further assumed that the **PPX** commands will perform equally well with or without the actual array controller hardware connected. If no or only partial hardware exists, the system will automatically enter a simulation mode. This mode will prominently announce what is being simulated and how and ensure that the user does not spend time taking simulated data while believing real data is being gathered.

2.0 Command and Data Communications Structure

2.1 Command/Response Communications Stream Definition RULE

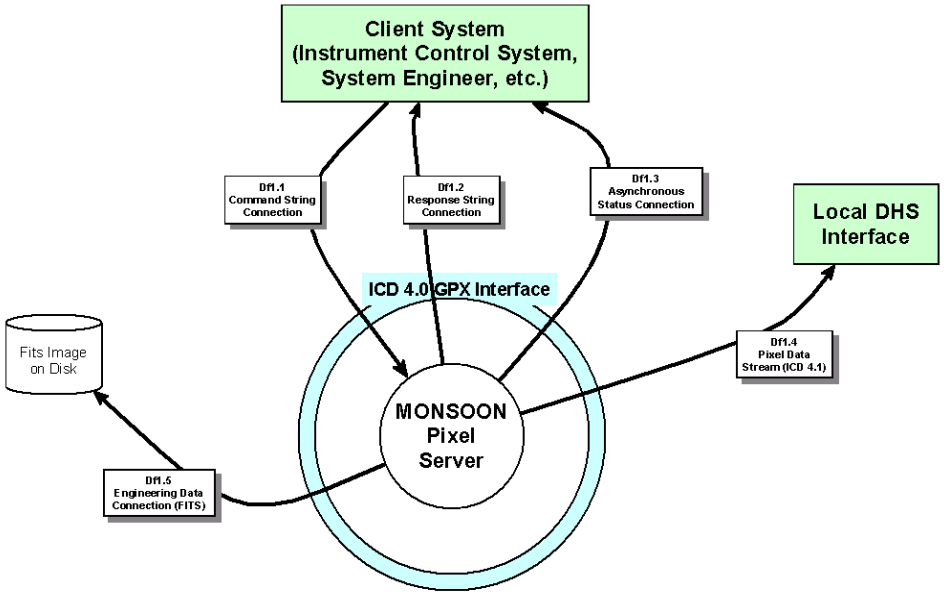
The Command/Response communications streams used by the **PPX** commands take place over a *socket* connection or through a terminal interface to the PAN system. The control layers which use the **PPX** connect to the **PPX** by connecting to a IP address and socket port number.

2.1.1 Socket Connections OBSERVATION

A single **GPX** system may communicate with multiple PANs. The command/response set assumes that the underlying communications protocol delivers a single ASCII string to the command processor for each command issued.

2.2 Socket Definitions RULE

The connection to a **PPX** will be by a socket connection. The primary communication socket (commands) is on Port *N* (TBD). A second socket on port N+1 would be used for the Asynchronous Status message stream.



PPX Communications Connections to Outside Entities
Figure 3

2.2.1 Primary Upper level system RULE

Only one connection to a **PPX** system is permitted at a time.

2.3 Status and Data Stream Interface RULE

The status information and data output from the **PPX** shall be a stream of messages across a socket connection. Any client wishing to use the data produced by the **PPX** may connect to this socket and receive the message stream.

2.3.1 Data Output Minimum RULE

At a minimum, every **PPX** will have a mode which is able to produce a FITS-formatted image on a local disk for engineering and diagnostic purposes. This capability is in addition to the message-based data interface

2.3.2 Status Output Minimum RULE

At a minimum, every **PPX** will have a mode which allows the display of status information on a terminal-like display as lines of status messages.

2.3.3 Local Client / Remote Client Message Formats RULE

For some messages, the message formats, will differ based on whether the client is connected locally or remotely. Local clients, running on the same machine as the **PPX,** will receive pixel data through a shared memory mechanism to avoid data copying where unnecessary. Remote clients will receive data across the socket.

2.3.4 Final Data Stream **OBSERVATION**

The final data stream from the **PPX** will be partially determined by local protocols.

2.3.5 Data Stream Minimum **PERMISSION**

It is possible for an implementation **PPX** to produce a stream of FITS images to export the data. This stream could then be transformed into the **PPX** standard format or into some local format such as Gemini DHS, NOAO PicFeed, IDL, and so forth, by an auxiliary process.

2.4 Status and Data Stream Message Protocol **COMMENTARY**

The status and data stream message protocol is defined in detail in ICD 1.1 DHS Interface - Input - Status and Data Stream Description (under development). The implementation of this protocol will be held in abeyance until Phase II of the **PPX** development. In general, the protocol uses messages passed over a socket stream to send pixel, status and event data to the outside world. The messages are considered to be a stream of bytes which are interpreted by the receiving process/client to be in one of the defined message formats.

For Version 1.0 of the ICD and until further notice, the **PPX** definitions for the data and status output are limited to those described in 2.3.1 through 2.3.5 above.

3.0 PAN Interface Commands

Table 1 - PAN PPX Commands and Equivalent GPX Commands

GPX Command	PPX Equivalent	PPX Usage	Section
gpxSetMode	ppxSetMode	Sets an entire configuration in a single PAN	4.4.2
gpxSetArrConfig	ppxSetArrConfig	Sets an array configuration in single PAN	4.4.3
gpxSetExp Config	ppxSetExp Config	Sets an exposure configuration in a single PAN	4.4.4
gpxSetIDPConfig	ppxSetIDPConfig	Sets the image data processing configuration in a single PAN	4.4.5
gpxSetMemConfig	ppxSetMemConfig	Read a file and download the values in it to DHE memory	4.4.6
gpxDump	ppxDump	Dump information about the interface state	
gpxSetAVP	ppxSetAVP	Set a single AVP in a PAN	4.4.7
gpxStartExp	ppxStartExp	Start the exposure in a single PAN	4.5.1
gpxArmExpTrigger	ppxArmExpTrigger	Arm an exposure in a single PAN	4.5.2
gpxStop	ppxStop	Stop the exposure in a single PAN	4.5.5
gpxAbort	ppxAbort	Abort the exposure in a single PAN	4.5.6
gpxPause	ppxPause	Pause the exposure in a single PAN	4.5.3
gpxResume	ppxResume	Resume the exposure in a single PAN	4.5.4
gpxShutter	ppxShutter	Control a shutter in a PAN	4.6.3
gpxShftImage	ppxShftImage	Send an image shift down to the DHE	4.6.4
gpxSimulate	ppxSimulate	Put part of the PAN-DHE system into simulate mode	4.8.1
gpxTestMode	ppxTestMode	Put part of the PAN-DHE system into test mode	4.8.2

Table 1 - PAN PPX Commands and Equivalent GPX Commands (Cont.)

GPX Command	PPX Equivalent	PPX Usage	Section
gpxPower	ppxPower	Control the DHE power systems	4.6.1
gpxReset	ppxReset	Reset some PAN and/or DHE subsystems	4.6.2
gpxGetState	ppxGetState	Get DHE/PAN state	4.7.1
gpxGetAVP	ppxGetAVP	Single attribute get	4.7.2
gpxAsyncMsg	ppxAsyncMsg	Send an asynchronous status message up to the next level	4.7.3
gpxAsyncResp	ppxAsyncResp	Send an asynchronous response message to the DHE	4.7.4
gpxPass	ppxPass	Pass a command down to the DHE	4.6.5
gpxExit	ppxExit	Shut down the connection or the entire PPX software system	

4.0 PAN Pixel Server Command Set

The command set proposed here is broken down into categories as follows: configuration commands; exposure control commands; internal action commands; status request commands, and a by-pass command.

Additional basic commands will probably be discovered as the command set develops and as new modes of operation are encountered. We believe that most of these can be handled by the by-pass command until a consensus of the user community decides to modify the basic interface with the new command. Once the basic interface is finalized, revisions to the interface will be infrequent and done by a consensus of the user community.

4.1 System Identification **RULE**

Each **PPX**-based PAN system within a **GPX** will be given a unique identifier when it is running. This identification will be assigned as part of the setup of the **GPX** system. The name will be associated with the IP address of the Pixel Acquisition Node.

4.1.1 PPX Names **RECOMMENDATION**

The identifier assigned to a **PPX** should be associated with the **GPX** it is controlled by. Thus, if an R&D Lab **GPX** with four PANs is assigned the name RnDLABGPX, each Pixel Acquisition Node associated with this **GPX** should be named RnDLABGPX_PANxxx and have an IP address appropriate to the location.

4.2 Command Structure **RULE**

All commands are delivered to the **PPX** as an ASCII string.

4.2.1 Command Length. RULE

These strings may be as long as necessary to perform the desired function. At a minimum, the **PPX** should be able to accept a command which is as much as 4096 characters long.

4.2.2 Command Length. RECOMMENDATION

Command strings should be easily human readable and should be kept to fewer than 80 characters long

4.2.3 Command Syntax. RULE

PPX Commands have the following general syntax:

CMDIDT *ppxCommandName* [<DIRECTIVE>] [Positional Parameter] [Attribute-Value Pair]*

The specific syntax for each command is explained in the paragraph explaining the particular command. Note that no CR/LF or newline character is required to end the command. The end of the string signals the end of the command string.

4.3 Whitespace Ignored RULE

Whitespace within a command is ignored. This includes all TAB and SPACE, but not newline characters. Thus the two commands below are equivalent:

“ppxSetMode <SAVE> newModeFileName”

“ppxSetMode <SAVE> newModeFileName

4.3.0 Command Identifier Tag (CMIDT) RULE

The first six characters of the command string will be a command ID determined by the controlling system. The **PPX** will tag all responses to this command with the command ID string.

4.3.1 Command Directives. RULE

Several commands in the **PPX** command set accept parameters called directives. These directives modify the behavior of the commands. A directive is indicated by enclosing the directive word in angle braces, that is, < > (the greater than, less than symbols.)

4.3.1.1 Ignoring Directives PERMISSION

Commands that do not accept directives may ignore a directive which is included in the command string. Directives that do not make sense for a particular command may be ignored. A list of the directives that each command must obey is listed with the command.

4.3.2 Command Behavior RULE

Each **PPX** command is responded to by the **PAN**. The response consists of a single string which gives the current status of the command.

4.3.2.1 Rejecting Commands **PERMISSION**

A PAN may reject **PPX** commands which arrive while it is busy. However, certain commands (noted in the command explanation) may not be rejected because the PAN is busy. As an example, a second *ppxStartExp* may be rejected if a current exposure is in process. However, a *ppxAbortExp* may not be rejected.

4.3.2.2 Multiple Commands **PERMISSION**

A PAN may "queue up" **PPX** commands to be executed in order of arrival. Commands that may not be rejected because the PAN is busy are also not permitted to be "queued up".

4.3.2.3 Response Time **RULE**

Each command response should be returned no more than 150 milliseconds after the command is received.

4.3.2.4 Minimum Response **RULE**

The minimum response to a command will be the strings "OK" or "ERROR".

4.3.2.5 Command Failure RULE

The response to a command that is rejected, ignored or fails, should include a explanation of why the command did not complete.

4.3.3 Command Response **RULE**

The PAN will respond to each command received with a single response string.

4.3.3.1 Minimum Response

The minimum response to a command will be the strings "OK" or "ERROR".

4.3.3.2 Additional Response Information **PERMISSION**

The response to a command in any system may include additional information in the form of strings added to the basic "OK" or "ERROR". The response:

"ERROR - your command failed because the cat died."

is perfectly acceptable, if somewhat confusing.

4.3.3.3 Additional Response Information

Commands may send additional responses while executing a command. These responses are returned as ppxAsyncMsg commands to the client program. The format of these returns is as follows:

ppxAsyncMsg *{ppxCommandBeingRespondedto}* "command Response"

4.4 Mode/Configuration Commands COMMENTARY

The first commands are configuration and/or mode selection commands. These commands are used to configure the detector/data processing system into a particular data taking mode. Several redundant commands are provided. In addition to the ppxSetMode command, which should be able to handle the complete configuration of a Generic Pixel Server, four redundant commands, ppxSetArrConfig, ppxSetIDPConfig, ppxSetExpConfig and ppxSetMemConfig are provided to configure parts of the overall system. Finally, a ppxSetAVP command is provided to set individual attributes of the PAN.

We feel that providing redundant configuration commands that divide the configuration task into functional groupings makes some sense. Much like inheritance in object oriented programs, a particular exposure may have much in common with other similar exposures. The provision of commands that only do a subset of the configuration task allows the observer more control over the exact changes made for each exposure. This may be important for certain arrays/controllers where changing the array behaviour even to the extent of resetting to the same value may disturb the array behaviour and data validity.

4.4.1 Configuration Files RULE

The basis of the configuration commands is the existence of configuration files or records. These files should determine the configuration of all attributes in the **PPX** needed to take data in a certain configuration.

4.4.1.1 Default Configuration File RULE

Each instance of a **PPX** system must have at least one configuration file called "xxx***Default***", where the xxx represents the instance name of the **PPX.** As an example, the Monsoon Pixel Server system for NEWFIRM might have a file called "***NewFirmDefault***". The Lab system for testing ORION IR arrays might be called "***orionLabDefault***".

4.4.1.2 Default File Content RULE

The default configuration file should include information which allows a ppxSetMode command to bring up the system in a configuration which is safe for the electronics and the array.

4.4.1.3 Configuration File Content RULE

The configuration file may contain high level commands from the generic command set, Attribute-Value Pairs or low level commands unique to a particular controller. A particular **PPX** implementation should know how to convert from the configuration command line to its internal command format.

4.4.1.4 Configuration File Content OBSERVATION

Since the implementers of a specific **PPX** system have complete control over the contents of the configuration files they use, they will decide on the usage of the low level command passing feature in the configuration file.

4.4.1.5 Configuration File Format. RULE

The format of the configuration files is defined in detail in Appendix V**Error! Bookmark not defined.**. Briefly, the configuration file consists of Attribute-Value pairs to be set by the configuration command and ppxCommands to be executed. These are organized into sections which relate to various portions of the **PPX** internal structure.

4.4.1.6 Configuration Sections **RULE**

For the purposes of organization and ease of modification, configuration files are divided into sections based on the structure of the exposure control task. Valid sections are [GENERAL], [ARRAY_VOLTAGES], [ARRAY_CLOCKS], [VIDEO_CHANNELS], [READOUT_PARAMS], [EXPOSURE_PARAMS], [DATA_PREPROCESS]. The details are included in Appendix VI.

4.4.1.7 Configuration Sections **COMMENTARY**

Below is a partial description of each configuration file section. The details of what is in each section is included in Appendix VI. [GENERAL] - contains attributes which relate to the entire data acquisition process or to meta parameters in the **PPX**. Information like the IP address of the **PPX** machine, socket port numbers to use, the string name of the **PPX** system and so forth.

- [ARRAYVOLTAGES] - contains information about the bias and clock voltage settings for running the array. It would also include the voltage number to voltage name conversion.
- [ARRAYCLOCKS] - contains information about the clock patterns to be used to run and read out the array.
- [VIDEOCHANNELS] - contains information about the individual video channels in the system, offset levels, gains for each channel and so forth.
- [READOUTPARAMS] - contains information about how the array will be readout, binning, number of digital averages, Fowler Samples to be done and so forth.
- [EXPOSUREPARAMS] - contains information about individual exposures, integration time, shutter state during exposure and so forth.
- [DATAPREPROCESS] - contains information about how the data is to be processed, number of coAdds, processing algorithm, data disposition, Exposure ID, Association ID and so forth.

4.4.1.8 Partial Configuration Files **RULE**

Configuration files may contain some none or all of the defined sections. Commands and Attribute-Value pairs which are listed outside of a section are processed by all of the configuration commands. Each configuration command (except ppxSetMode) handles only some of the configuration file sections. The sections are handled by each command is specified in the command descriptions below.

4.4.1.9 Building Configuration Files **PERMISSION**

It is our intent that these configuration files will be constructed by the detector engineer and/or instrument scientist to place the pixel server into a particular image taking mode. However, since each configuration command should be able to respond to a directive <SAVE>, it is permissible for these files to be built and modified by the observer.

4.4.1.10 Building Configuration Files **PERMISSION**

The development team for a particular **PPX** may restrict the use of the <SAVE> directive and may restrict the observer's ability to modify standard configuration files.

4.4.1.11 Building Configuration Files **RECOMMENDATION**

The development team for a particular **PPX** should always provide some easy-to-use mechanism which allows an observer to save configurations and to return to a previously saved configuration.

4.4.2 Set Pixel Server Mode - ppxSetMode **RULE**

This command sets the system into a particular named system mode. These modes may be constructed in advance by the engineer or scientist responsible for the system or may be done by the observer for a particular observation run.

4.4.2.1 Config File Sections **RULE**

When parsing a configuration file, in the absence of any Attribute-Value pairs defining other behavior this command will handle all sections of the configuration file. It will always handle commands and Attribute Value Pairs included in the [GENERAL] section.

4.4.2.2 Acceptable Parameters RULE

The ppxSetMode command takes a single parameter which is the name of the mode description file to be loaded. This file will contain a set of commands to the pixel server to put the system into the desired exposure mode.

4.4.2.3 Acceptable Attribute-Value Pairs **RULE**

This command has only three Attribute-Value pairs which can be set from the command line:

detConfig, expConfig and idpConfig are the names of the configuration files to be used when the three ppxSetXxxConfig commands are executed. These Attribute-Value pairs are included only if the files for the detector, exposure and IDP setup are to be taken from a file other than the main mode file.

4.4.2.4 Supported Directives **RULE**

This command supports the <SAVE> directive. When called with the <SAVE> directive, it will produce a configuration file that is able to restore the current configuration of the **PPX** at a later time. If the command includes any of the acceptable Attribute-Value pairs, that portion of the save will be done to a file with the name given.

4.4.2.5 Preserving Default Files **RECOMMENDATION**

It is recommended that the implementing software be constructed so that default and standard configuration files are not modified by the observer without some protection such as password or a confirmation question.

4.4.2.6 Preserving Default Files **PERMISSION**

A PPX may restrict the observer's ability to modify certain parts of the system without consulting with the instrument scientist or detector engineer. Some parameters of the configuration may be contained only ion files, which cannot be easily modified.

4.4.2.7 Command Syntax **RULE**

ppxSetMode [<SAVE>] *cfgFileName* [arrConfig=arrFilename] [expConfig=expFileName] [idpConfig=idpFileName]

Example:

To set the **PPX** configuration to the state described by the file *O*RION135

ppxSetMode *ORION135*

save the current configuration in the file ORION135 (**NOTE**: security concerns)

ppxSetMode <SAVE> *ORION135*

Set the configuration to the state described by the file ORION135, except the Exposure configuration is to come from the file *myExposure.Obs*

ppxSetMode *ORION135 expConfig=”*myExposure.Obs”

4.4.3 Set Array Configuration - ppxSetArrConfig **RULE**

This command sets the system into a particular named array configuration mode. Generally these configuration files are set up by the development engineer and/or instrument scientist. The observer for a particular run may set up configurations for a particular observation run. The command can be used with or without a file name to change individual parameters in the current configuration. Command line Attribute-Value pairs will override the default settings in the configuration file or the current settings system settings if the "-" parameter is used.

4.4.3.1 Config File Sections **RULE**

When parsing a configuration file, this command will only handle commands and Attribute-Value Pairs included in the [ARRAYVOLTAGES], [ARRAYCLOCKS], and [VIDEOCHANNELS] sections of the configuration file.

4.4.3.2 Config File Sections **RULE**

This command will not handle commands and Attribute-Value Pairs included under other sections of the configuration file.

4.4.3.3 Acceptable Parameters **RULE**

The ppxSetArrConfig command takes a single parameter which is the name of the configuration file to be loaded or a "-" if no file is to be processed. This file will contain a set of commands to the **PPX** to put the array into the desired exposure mode.

4.4.3.4 Acceptable Attribute Value Pairs **RULE**

This command can accept optional Attribute-Value pairs which can be set from the command line. Any Attribute-Value pair which can appear in one of the three configuration file sections the command processes can be included on the command line as an optional Attribute-Value pair.

4.4.3.5 Supported Directives **RULE**

This command supports the <SAVE> directive. If the command includes any of the acceptable Attribute-Value pairs, that Attribute-Value pair will be substituted for the version in the current configuration.

4.4.3.6 Preserving Default Files **RECOMMENDATION**

It is recommended that the implementing software be constructed so that default and standard configuration files are not modified by the observer without some protection such as a password or a confirmation question.

4.4.3.7 Preserving Default Files **PERMISSION**

A **PPX** may restrict the observer's ability to modify certain parts of the system without consulting with the instrument scientist or detector engineer. In addition, some Attribute-Value pairs may not be permitted to change using these techniques.

4.4.3.8 Command Syntax **RULE**

SetArrConfig [<SAVE>] {*arrModeFileName* | -} [Attribute-Value pairs]*

Example:

load the array configuration stored in the file "*orion1378 *"

SetArrConfig *orion1378*

save the current configuration in the file "orion1378"
(note: security concerns)

SetArrConfig <SAVE> *orion1378*

configure the array in the **PPX** system using the configuration file " tex2048-a23", but override the values of the *binning* and *digitalAvgs* attributes with the values given.

SetArrConfig *tex2048-a23* binning=4 digitalAvgs=8

set array configuration parameter VggCL1 to a new value.

SetArrConfig - VggCL1=-4.6

4.4.4 Set Exposure Configuration - ppxSetExpConfig **RULE**

This command puts the **PPX** into a particular named configuration for an exposure. Generally the configurations are first set-up by the engineer or scientist in charge of development. These configurations are then modified by the Observer. The command can be used with or without a file name to change individual parameters in the current configuration. Command line Attribute-Value pairs will override the default settings in the configuration file named or the current settings system settings if the "-" parameter is used.

4.4.4.1 Config File Sections **RULE**

When parsing a configuration file this command will only handle commands and Attribute-Value Pairs included in the [READOUT_PARAMS], [EXPOSURE_PARAMS] and [DATA_PREPROCESS] sections.

4.4.4.2 Config File Sections **RULE**

This command will not handle commands and Attribute-Value Pairs included under other sections of the configuration file.

4.4.4.3 Acceptable Parameters **RULE**

The ppxSetExpConfig command takes a single parameter which is the name of the configuration file to be loaded or a "-" if no file is to be processed. This file will contain a set of commands to the **PPX** to put the array into the desired exposure mode.

4.4.4.4 Acceptable Attribute-Value Pairs **RULE**

This command can accept optional Attribute-Value pairs which can be set from the command line. Any Attribute-Value pair which can appear in one of the three configuration file sections the command processes can be included on the command line as an optional attribute value pair.

4.4.4.5 Supported Directives **RULE**

This command supports the <SAVE> directive. If the command includes any of the acceptable Attribute-Value pairs, that Attribute-Value pair will be substituted for the version in the current configuration.

4.4.4.6 Preserving Default Files **RECOMMENDATION**

It is recommended that the implementing software be constructed so that default and standard configuration files are not modified by the observer without some protection such as a password or confirmation question.

4.4.4.7 Preserving Default Files **PERMISSION**

A **PPX** may restrict the observer's ability to modify certain parts of the system without consulting with the instrument scientist or detector engineer. In addition, some Attribute-Value pairs may not be permitted to change using these techniques.

4.4.4.8 Command Syntax **RULE**

ppxSetExpConfig [**<SAVE>**] {*expConfigFileName* | **-**} [Attribute-Value pairs]*

Example**:**

Load the exposure configuration stored in the file "*StandardStar* "

ppxSetExpConfig *StandardStar*

load the exposure configuration stored in the file "NGC1721" overriding the integration time.

ppxSetExpConfig *NGC1721* integration=65.345

set the integration time, arrPower and fSamples values to the desired values.

ppxSetExpConfig - integration=65.345 arrPower=On fSamples=16

Save the current configuration in the file "*myConfig"* overriding the current integration time

ppxSetExpConfig <SAVE> *myConfig* integration=20.0

4.4.5 Set Image Data Pre-processor Configuration - ppxSetIDPConfig **RULE**

This command puts the **PPX** into a particular named configuration for the data pre-processing of an exposure. Generally the configurations are first set up by the engineer or scientist in charge of development. These configurations are then modified by the observer. The command can be used with or without a file name to change individual parameters in the current configuration. Command line Attribute-Value pairs will over-ride the default settings in the configuration file named or the current settings system settings if the "-" parameter is used.

4.4.5.1 Config File Sections **RULE**

When parsing a configuration file this command will only handle commands and Attribute-Value Pairs included in the [EXPOSURE_PARAMS] and [DATA_PREPROCESS] sections.

4.4.5.2 Config File Sections **RULE**

This command will not handle commands and Attribute-Value Pairs included under other sections of the configuration file.

4.4.5.3 Acceptable Parameters **RULE**

The **ppxSetExpConfig** command takes a single parameter which is the name of the configuration file to be loaded or a "-" if no file is to be processed. This file will contain a set of commands to the **PPX** to put the array into the desired exposure mode.

4.4.5.4 Acceptable Attribute-Value Pairs **RULE**

This command can accept optional Attribute-Value pairs which can be set from the command line. Any Attribute-Value pair which can appear in one of the three configuration file sections the command processes can be included on the command line as an optional Attribute-Value pair.

4.4.5.5 Supported Directives **RULE**

This command supports the <SAVE> directive. If the command includes any of the acceptable Attribute-Value pairs, that Attribute-Value pair will be substituted for the version in the current configuration.

4.4.5.6 Preserving Default Files **RECOMMENDATION**

It is recommended that the implementing software be constructed so that default and standard configuration files are not modified by the observer without some protection such as a password or confirmation question.

4.4.5.7 Preserving Default Files **PERMISSION**

A **PPX** may restrict the observer's ability to modify certain parts of the system without consulting with the instrument scientist or detector engineer. In addition some Attribute Value pairs may not be permitted to change using these techniques.

4.4.5.8 Command Syntax **RULE**

ppxSetIDPConfig [**<SAVE>**] { *idpConfigFileNa*me | **-** } [Attribute-Value pairs]*

Example:

load the Data Pre-processing configuration stored in the file "*standardStar*"

ppxSetIDPConfig *standardStar*

load the configuration stored in the file "*standardStar*" overriding the value of destination

ppxSetIDPConfig *standardStar* destination=DHS

set directory, file, coadds and procAlgorithm attributes to the desired values.

ppxSetIDPConfig - directory="/home/mrEngineer/20011127" file="ngc1721" coadds=16 procAlgorithm=SUR

Save the current Data Pre-processing configuration stored in the file "*standardStar*"

ppxSetIDPConfig <SAVE> *standardStar*

4.4.6 Set DHE Memory to Desired Values - ppxSetMemConfig **RULE**

4.4.7 **Set** Attribute-Value **Pair - ppxSetAVP** **RULE**

This command is provided to allow a user to set an arbitrary individual parameter in the **PPX**. Any low level internal parameters available in the Generic Pixel Server should be settable with this command. See Appendix IV for a list of all the attributes this command can set

4.4.7.1 Acceptable Parameters **RULE**

The **ppxSetAVP** command takes as a parameter a single Attribute-Value pair to be set.

4.4.7.2 Acceptable Attributes **PERMISSION**

The **ppxSetAVP** command may refuse to set certain attributes based on a level of protection attribute. This is allowed to ensure that critical attributes are not given values which could result in lost or invalid data.

4.4.7.3 Acceptable Attributes **COMMENTARY**

The **ppxSetAVP** command may fail to process an Attribute-Value pair for several reasons. First, the attribute may not exist. Second, the value may not be a valid value for this attribute. Third, the **PPX** may be protecting this attribute from change. Fourth, there may be a hardware failure.

4.4.7.4 Acceptable Attribute-Value Pairs **RULE**

This command can accept optional Attribute-Value pairs which can be set from the command line. Any Attribute-Value pair which can be set in the **PPX** can appear on the command line as an optional Attribute-Value pair.

4.4.7.5 Supported Directives **RULE**

This command supports no directives.

4.4.7.6 Command Syntax **RULE**

ppxSetAVP AttribName=Value [Attribute-Value Pair]*

Example:

set fSamples to 16

ppxSetAVP fSamples=16

set the attributes coadds, fSamples, integrationTime and imageTitle to the values listed

ppxSetAVP coadds=32 fSamples=8 integrationTime=40.5 imageTitle=”NGC2123 K’ Filter”

4.5 Exposure Sequence Control Commands

The Exposure control commands directly control the taking, pre-processing and storage of data. It is assumed that the **PPX** can understand how an exposure is created and it knows how to create a series of exposures all of which use the same **PPX** configuration. The assumption here is a minimal level of intelligence in the **PPX.** It is assumed that complex patterns of exposures involving changes in the **PPX** configuration are understood by the upper level systems.

4.5.0.1 Optional Directives **RULE**

These commands accept no optional directives.

4.5.0.2 Minimum Exposure Sequence **RULE**

The **PPX** must at a minimum be able to sequence the series of steps needed to produce the data which results from a single exposure. This means that it must be able to clear or reset the detector, do a bias readout of the detector if necessary, open any locally controlled shutter if required, integrate for the required integration time, close a locally controlled shutter and read out the final data from the array.

4.5.0.3 Minimum Data Processing. **RULE**

The **PPX** must be capable of performing the data preprocessing required for the detector being used in the system. The system developers should specify the precise nature of that preprocessing for each exposure configurations in the system.

4.5.0.4 Unimplemented Commands **PERMISSION**

Some detectors and **PPX** systems may have physical characteristics which make some of the Exposure Sequence Control Commands less than useful. For instance, IR systems generally do not have shutters which allow the **ppxPauseExp** or **ppxResumeExp** to be used in a meaningful way. These systems may ignore the meaningless commands.

4.5.0.5 Unimplemented Commands **RULE**

Systems that ignore commands must include the fact that the command was ignored in the command response. The response to an ignored **ppxPauseExp** command might be “OK - **ppx**PauseExp ignored.”

4.5.0.6 Optional Command Attribute-Value Pairs **RULE**

All of the Exposure Sequence Control Commands listed in this section accept optional Attribute-Value pairs on the command line. These Attribute-Value pairs will modify some portion of the current system configuration. Optional Attribute-Value pairs are processed prior to processing the Exposure Sequence Control Command.

4.5.0.7 Optional Command Attribute-Value Pairs **RULE**

If the processing of an optional Attribute-Value pair fails, the associated exposure sequence command should also fail.

4.5.1 Start Exposure - ppxStartExp **RULE**

The **ppxStartExp** command will start an exposure sequence using the current configuration. In this context of the observing sequence, the **PPX** performs all tasks under its control which are required to generate a single exposure data set or multiple exposures using the same configuration. As an example, a **PPX** for a CCD may clear the array, open the shutter, wait the integration time, close the shutter and read out the array. An IR **PPX** may reset the array and produce the appropriate number of Fowler samples and coadd frames as directed by the configuration. In either case the data preprocessing and storage side of the system would produce a single data set.

4.5.1.1 Exposure Start Timing **RULE**

No timing guarantee is made when using this command. The exposure sequence begins as soon as the command is processed. If an exact timing or start trigger is required, see ppxArmExpTrigger.

4.5.1.2 Command Syntax RULE

ppxXStartExp [Attribute-Value pairs]*

Example:

take an exposure overriding the current integration time.

ppxStartExp integration=10.0

take an exposure overriding the current integration time and numPics values.

ppxStartExp numPics=10 integration=10.0

4.5.2 Wait for Exposure Trigger - ppxArmExpTrigger **RULE**

The **ppxArmExpTrigger** command arms an exposure sequence to be started when the defined trigger event occurs. The trigger may be a hardware trigger or a specific time or some other trigger mechanism.

4.5.2.1 Acceptable Parameters **RULE**

This command normally takes two attribute value pairs. The first, "*expTrigger*" defines which of the implemented triggers will start the exposure sequence. The second, "*expTriggerTimeOut*" indicates how long the system should wait for the trigger before announcing a failure in the **ppxArmExpTrigger** command. These attributes could be configured before the **ppxArmExpTrigger** is issued with the same result as putting them in the command string.

4.5.2.2 Acceptable Parameters **RULE**

If the "*expTrigger*" and "*expTriggerTimeOut*" attributes have not been configured or supplied on the command line, this command should fail with an "ERROR" return.

4.5.2.3 Trigger Events **RULE**

Assuming the configured trigger occurs, the result of this command will be the same as if a **ppxStartExp** Command was processed at the moment the trigger was received. Each **PPX** will define its own set of usable trigger events.

4.5.2.4 Exposure Start Timing **RULE**

The exposure started by the occurrence of the correct trigger must begin within 10ms of the time the Trigger has been received.

4.5.2.5 Command Syntax **RULE**

ppxArmExpTrigger [Attribute-Value pairs]*

Example:

wait up to 10 seconds for the trigger CHOPTRIGGER then start the exposure

ppxArmExpTrigger trigger=CHOPTRIGGER timeout=10.0

4.5.3 Pause Exposure - ppxPause **RULE**

This command causes the **PPX** to pause the currently running exposure for later restart. The current exposure is halted, (shutter is closed) and the array is put into a state that allows the later restart.

4.5.3.1 Exposure Data **RULE**

In the absence of other events (charge shifting and others), the data taht results from an exposure that is paused and later resumed should be indistinguishable from a continuous integration of similar duration.

4.5.3.2 Pause Ignored PERMISSION

For certain types of detectors and controllers the pause-resume cycle may be impossible to implement in a way which meets 3.4.3.1. These systems may ignore the **ppxPause** command with a response of “OK - Pause ignored.”

4.5.3.3 Acceptable Parameters **RULE**

The **ppxPause** command takes no parameters but may accept optional Attribute-Value pairs which modify the state of the system for the subsequent resume.

4.5.3.4 Command Syntax **RULE**

ppxPause [Attribute-Value pairs]*

Example:

pause the current exposure changing the total integration time to 20.0 seconds for the resumed integration.

ppxPause integration=20.0

4.5.4 Resume Exposure - ppxResume **RULE**

This command causes the **PPX** to resume a currently paused exposure. The current exposure is restarted, (shutter is opened, and so forth) and the array is put into an exposing state

4.5.4.1 Exposure Data **RULE**

In the absence of other events, i.e. charge shifting, etc. the data which results from an exposure which is paused and later resumed should be indistinguishable from a continuous integration of similar duration.

4.5.4.2 Resume Ignored **PERMISSION**

For certain types of detectors and controllers the pause-resume cycle may be impossible to implement in a way which meets 3.4.4.1. These systems may ignore the **ppxResume** command with a response of “OK - Resume ignored.”

4.5.4.3 Acceptable Parameters **RULE**

The **ppxPause** takes no parameters but may accept optional Attribute-Value pairs which modify the state of the system for the subsequent resume.

4.5.4.5 Command Syntax **RULE**

ppxResume [optional Attribute-Value pairs]*

Example:

resume the currently paused exposure changing the total integration time to 20.0 seconds.

ppxResume integration=20.0

4.5.5 Stop Exposure - ppxStop **RULE**

This command stops the current exposure. The system will close the shutter and complete the array readout as appropriate.

4.5.5.1 Data Valid **RULE**

Every attempt should be made in the **PPX** to ensure the validity of the saved data. If the exposure consists of a number of coadds, the integration time will be allowed to complete so that all coadded frames are of the same magnitude. The data captured by the end of the stop command will be saved.

4.5.5.2 Status Information **RULE**

Any status information, such as integration time, included with the final data, should be the actual values achieved, not the requested values.

4.5.5.3 Command Syntax

ppxStop [Attribute-Value pair]*

Example:

Stop the current integration save the data under the imageID “haltedORION123”

ppxStop imageID=”haltedORION123”

4.5.6 Abort Exposure - ppxAbort **RULE**

This command aborts the current exposure. The system will close the shutter and return the array to the ready state as required.

4.5.6.1 Data Discarded **RULE**

The data captured for an exposure prior to an abort will be discarded.

4.5.6.2 System State **RULE**

The **PPX** system will be ready to accept any valid **PPX** command as soon as the actions needed to complete the **ppxAbort** command are performed.

4.5.6.3 Command Syntax **RULE**

ppxXAbort [Attribute-Value pair]*

Example:

Abort the current observation and set VggCL1 to 3.54 volts

ppxAbort VggCl1=3.54

4.6 System Control Commands COMMENTARY

These commands affect the low level system directly. They generally relate to actions to be taken that are orthogonal to the data taking process. That is, they may occur at any time during the data taking process. Some of these commands can destroy the validity of the data obtained during an exposure or cause a messy termination of an exposure in progress.

4.6.1 Power Control - ppxPower RULE

This command controls the power going to the **PPX** subsystems. The exact nature of the available power control will be defined by the underlying hardware. The designers of the **PPX** will define the subsystem names of the controlled hardware.

4.6.1.1 Acceptable Parameters RULE

If the **ppxPower** command is implemented by a **PPX** in a meaningful way, the command must accept at least one attribute “sysPower” with allowed values of “On” or “Off”.

4.6.1.2 Acceptable Directives RULE

This command accepts two directives which are unique to this command. The “<ON>” and “<OFF>” directives are a shorthand for the attribute value pairs normally used for controlling power. The use of these directives will cause ALL subsystems which have remote power control to be turned either on or off.

4.6.1.3 No Power Control Available RULE

A **PPX** may be designed which has no remote power control of subsystems available. Such a system would respond to this command with a response of “OK - No remote power control available”.

4.6.1.4 Command Syntax RULE

ppxPower [**<ON>** | **<OFF>**] [Attribute-Value pairs]*

Example:

turn power to all subsystems on.

ppxPower sysPower=On or **PPXPower** all=On or **PPXPower** <ON>

turn power to unit1 on and power to unit2 off

ppxPower unit1=On unit2=Off

4.6.2 Reset System - ppxReset RULE

This command controls the reset state of the **PPX** hardware and software. The exact nature of the available reset control will be defined by the **PPX** system designers. The **PPX** designers will define the subsystem names and reset levels available.

4.6.2.1 Acceptable Parameters RULE

The **ppxReset** command will take attribute value pairs defined by the underlying hardware and software designers to allow control of the reset state of that hardware or software component. If the command is implemented in a meaningful way, it must accept at least one attribute “sysReset” with value of “TRUE” (1).

4.6.2.2 Acceptable Directives **RULE**

This command accepts a directive which is unique to this command. The "<RESET>" directive is a shorthand for the Attribute-Value pairs normally used for controlling the reset of the system. The use of this directive will cause the **PPX** and ALL subsystems of the **PPX** to reset to some known state.

4.6.2.3 No Reset Control Available **RULE**

A **PPX** may be designed which has no remote reset control of subsystems available. Such a system would respond to this command with a response of "OK - No remote reset available".

4.6.2.4 Reset Levels **PERMISSION**

A **PPX** system may be designed with multiple levels of reset state available. An example might be the ability to reset the system to the last known good configuration. These levels are left to the designers to define and describe.

4.6.2.5 Command Syntax **RULE**

ppxReset [**<RESET>**] [Attribute-Value pairs]*

Example:

Reset the entire **PPX**

ppxReset <RESET> or **PPXReset** sysReset=1 or **PPXReset** all=TRUE

reset the embedded system and Unit2 of the **PPX**

ppxReset embededSystem=1 unit2=1

4.6.3 Shutter Control - ppxShutter **RULE**

This command directly controls the state of any shutter like devices internal to the **PPX**. These might include shutters, polarizers and others which have only a few states. The designers of the **PPX** will define the exact nature of the devices controllable from the **PPX**.

4.6.3.1 Acceptable parameters **RULE**

The **ppxShutter** command may take Attribute-Value pairs defined by the designers of the underlying hardware to allow control of the state of the shutter-like device. If the command is implemented in a meaningful way, it must accept at least one attribute "shutter" with values of "OPEN" or "CLOSE".

4.6.3.2 Acceptable Directives **RULE**

This command accepts two unique directives. The "<OPEN>" and "<CLOSE>" directives will be used for controlling a generic shutter internal to the system. The use of these directives will cause the **PPX**-controlled shutter to open or close.

4.6.3.3 No Shutter Control Available **RULE**

A PPX may be designed which has no locally controlled shutter like device. Such a system would respond to this command with a response of "OK - No shutter available".

4.6.3.4 Command Syntax **RULE**

ppxShutter [<OPEN> | <CLOSE>] [Attribute-Value pairs]*

Example:

Open the default internal shutter

ppxShutter <OPEN> or ppxShutter shutter=OPEN

Close the default internal shutter

ppxShutter <CLOSE>

Set a locally controlled device “Device1” to state “STATE0” defined by the system designers.

ppxShutter device1=STATE0

4.6.4 Control Image Shift - ppxShiftImage **RULE**

This command will cause the low level system to deal with shifting the image on the chip or in memory. The **PPX** designers and integrators for the particular detector system will determine exactly what occurs when this command is executed.

4.6.4.1 How an Image Is Shifted **PERMISSION**

The **PPX** designer/implementers will specify how the image shift is accomplished, whether by shifting the charge on an orthogonal transfer CCD or by shifting the next coadded image in memory before coadding.

4.6.4.2 Acceptable Parameters **RULE**

The **ppxShiftImage** command will take two parameters in the form of Attribute-Value pairs. These attributes “xShift” and “yShift” will define how much the data is shifted. The values associated with the attributes will be the amount to shift in pixels, and may be expressed as floats or integers. The **PPX** will define what the amount of shift will be in each specific case.

4.6.4.3 Acceptable Parameters **RULE**

The **ppxShiftImage** should fail, with a response of “ERROR - Invalid [x|y]Shift amount”, if the xShift, yShift attributes are either not defined or have invalid values.

4.6.4.4 No Image Shift Available **RULE**

A particular **PPX** system may not provide the means to perform an image or charge shift required by this command. In that case the **PPX** will respond to this command with an “OK - No image shifting available”.

4.6.4.5 Command Syntax **RULE**

ppxShiftImage [Attribute-Value pairs]*

Example:

shift the image charge or data in image memory by 4.3 pixels in X and 2.7 pixels in Y

ppxShiftImage xShift=4.3 yShift=2.7

Shift the image or charge the current xShift value in X and -1.3 pixels in Y

ppxShiftImage yShift=-1.3

4.6.5 Command Pass-through - ppxPass **COMMENTARY**

The command pass through is included to allow an arbitrary command to be passed to the subsystems of the **PPX**. The use of this command will be different for every **PPX** since the subsystem interfaces and command structures may be different. The implementers of a **PPX** who wishes to use this facility must provide, in the **PPX** software, a translator module which will parse the incoming command string and convert it to the proper format for the subsystem command processor.

This command should be considered an emergency command it is expected that its use will be infrequent. It should not be used as a standard access to the low level system to bypass the error checking and handling provided by the **PPX** software.

4.6.5.1 By-pass Normal Command Processing - **ppxPass** **RULE**

The **ppxPass** command interacts directly with the DHE registers in a direct way.

4.6.5.2 Acceptable Parameters **RULE**

This command takes three or four parameters;

- An action character, r for read, w for write and s for write with no echo.
- A DHE board number 1 through 6 or 8 depending on the backplane in the system
- The address of the register on the board to access.
- An optional value required for writes to the DHE.

4.6.5.3 Command Purpose **RECOMMENDATION**

This command should be used by the system engineers and developers during debugging and diagnostics only. It may be used to bypass range checking on parameters, change voltages normally not accessible to the user or put the system into a hardware or software debug mode.

4.6.5.4 Command Syntax **RULE**

ppxPass s|r|w slot regAddr [value]

Example:

ppxPass r 3 0x0040

ppxPass w 1 0xfffc 0x21

4.6.6 Shut Down a Connection or the PAN - ppxExit **RULE**

With no directive **ppxExit** closes a connection. With directive <ALL>, shuts down connection and entire PAN software suite.

4.7 Status and Information Commands COMMENTARY

These status and information commands enable the upper level systems to determine the state of the **PPX** by querying the **PPX.** The commands also allow the **PPX** system to asynchronously report the status of the **PPX**.

4.7.0.1 Status Reporting Modes - Request-Response **RULE**

Each **PPX** should support two modes for reporting the status of the **PPX** and its subsystems. The first mode involves an upper level system requesting information about the pixel server. This is covered under normal command-response system used by all of the other **PPX** commands.

4.7.0.2 Status Reporting Modes - Asynchronous Report **RECOMMENDATION**

The second status reporting mode the **PPX** should support is the Asynchronous Reporting mode. This mode is used whenever the **PPX** wishes to inform the upper level system about a change in its status asynchronously, that is, without the upper level system requesting the information.

4.7.0.3 Status Reporting Modes - Asynchronous Report **COMMENTARY**

An Asynchronous Report may occur when an error is detected by the **PPX** or when an ongoing command changes status, such as when a readout starts during an integration. Errors reported might include a loss of a detector control voltage, a low level controller power cycle or any number of events which will affect the observing and should be reported to the higher level system.

4.7.0.4 Status Reporting Modes - Asynchronous Report **PERMISSION**

A **PPX** design team may decide not to implement this reporting mode.

4.7.0.5 Status Reporting Modes - Asynchronous Report - Upper Level Systems **REQUIREMENT**

Upper level systems which expect to interface with a **PPX** must be able to asynchronously receive the **ppxAsyncMsg** message from the **PPX**. They must also respond to that message if necessary with a **ppxAsyncResp** command to the **PPX** to inform it that the problem has been acknowledged.

4.7.1 Retrieve the Server System State - ppxGetState **RULE**

This command will cause the **PPX** to return the values of attributes in the system. The system will return a string consisting of Attribute-Value pairs for the attributes and settings required to define the system state. Not all variables in the system need be reported. The **PPX** Designers need to specify which attributes will be reported

4.7.1.1 Acceptable Directives **RULE**

This command will accept a number of directives unique to this command. These directives will be defined by the **PPX** implementers and will relate a keyword (the directive name) to a set of Attribute-Value pairs.

4.7.1.2 Minimum Directives **RULE**

This command should recognize and have a defined response for at least the following directives: “<MODE>”, “<ARRAY>”, “<EXPOSURE>” and “<IDP>”. The return string associated with these directives should be a set of attributes set by the respective **ppxSetXxxConfig** commands.

4.7.1.3 Additional Directives **RULE**

Additional directives may be provided for obtaining specific information about **PPX** subsystems or for the purpose of debugging and diagnosis of system problems. For example, a **PPX** system might define a directive "<**VOLTS>**" which returns the current levels of all array control voltages in the system.

4.7.1.4 Acceptable Attributes **RULE**

The **ppxGetState** command should accept a single Attribute-Value pair which gives the name of a file into which a copy of the information retrieved should be stored. The information provided would be appended to the file named.

4.7.1.5 Command Syntax

ppxGetState [<**MODE>** | <**ARRAY>** | <**EXPOSURE>** | <**IDP>** | *otherDirectives*]
[logFileName="logfile"]

Example:

retrieve the value of the attributes associated with the **PPX**SetMode command (i.e. everything)

ppxGetState <**MODE>**

retrieve the values of all array voltages and save them in a log file

ppxGetState <VOLTS> logfileName="/**PPX**/tmp/voltageLogFile"

4.7.2 Retrieve the Value of an Attribute - ppxGetAVP **RULE**

This command will cause the **PPX** to return the value of the attributes listed in the command string. The response will be a string consisting of "OK – " followed by the Attribute-Value pairs reporting the current values of the requested attributes.

4.7.2.1 Acceptable Attributes **RULE**

The command should be able to accept any valid attribute name.

4.7.2.2 Acceptable Attributes **RULE**

If the name is not used in the **PPX** or if the attribute has not been set, the **PPX** should return a response of "OK - *attributeName*=N/A".

4.7.2.3 Command Syntax **RULE**

ppxGetAVP attributeName [attributeName]*

Example:

return the current value of the coadds attribute.

ppxGetAVP coadds

The response might be

“OK - coadds=32”

return the values of the three attributes listed

ppxGetAVP waveformName directory fSamples

The response might be

”OK - waveformName=‘HawaiiII’, directory=‘/home/observer/200200127’, fSamples=8”

get the value of the attribute *statusCat*

“OK - statusCat=N/A”

4.7.3 Report an Asynchronous Status Change - ppxAsyncMsg **RULE**

This command is a message from **PPX** to the interested upper level system(s). It is used to report events in the **PPX** which are not a response to a preceding command from the upper level system. These events may be errors in hardware or software, notification of the completion of some long running command, reports on the progress of an exposure or process. This command might report on the progress of the cool down of an array so the upper level system can begin observing as soon as possible.

COMMENTARY

This message is needed to allow the **PPX** and upper level system to handle system failures and changes in status which occur over long periods of time. It allows the **PPX** to report events within the system which may be of interest to the upper level systems and users. It can be used to report on command completion for those commands that do not complete immediately, such as exposures of long duration, slow voltage ramp-ups and others.

This message also allows the **PPX** implementers the freedom to implement a system which responds to each command immediately and then reports command completion at a later time. The addition of the command ID at the beginning of the string will allow a tag to follow each command through the system. This will allow the addition of more parallelism in command execution should we find it unsuccessfully.

4.7.3.1 Acceptable Directives **RULE**

This command accepts the following directives: <RESPOND>, <WARNING> or <FATAL>. Their meanings are as follows:

RESPOND - This directive requires the upper level system to send a command back to the **PPX**. This directive will always be sent from the **PPX** with an attribute name as a parameter. The upper level system should send a **ppxAsyncResp** command with the attribute name as the parameter.

WARNING - This directive requires no response from the upper level system, but may initiate actions at the upper level.

FATAL - This directive informs the upper level that something "really bad" has happened and it is likely that the **PPX** will need some intervention to bring things back to normal.

4.7.3.2 Acceptable Directives **PERMISSION**

Any **ppxAsyncMsg** command received by an upper level system which contains no directive will be treated as an informational message only. The upper level system may choose to ignore these messages.

4.7.3.3 Acceptable Directives **RECOMMENDATION**

Choosing to ignore a **ppxAsyncMsg** message without a directive may result in important status information being lost. It is recommended that as a minimum such messages be displayed and possibly logged so the information is not lost.

4.7.3.4 Acceptable Parameters **RULE**

This command can take as parameters one or more of, a directive, an attribute name, an arbitrary string message or one or more Attribute-Value pairs.

4.7.3.5 Acceptable Parameters RULE

The **PPX** may send this command with only a directive and an arbitrary string message

4.7.3.6 Upper Level System Responses **RULE**

If the **PPX** sends a Directive and an Attribute name, with or without an arbitrary string report, the upper level system should act on the message as appropriate. At a minimum, in the case of a **<RESPOND>** directive, the upper level system should respond to the **PPX** with a **ppxAsyncResp** message containing the Attribute name sent in the **ppxAsyncMsg** message.

4.7.3.7 Additional Actions **PERMISSION**

An upper level system may take such additional actions as are necessary to return to normal processing after receiving this message. The **PPX** implementers should spell out the actions the **PPX** will take when it sends a message of this type

4.7.3.8 Command Syntax **RULE**

ppxAsyncMsg [**<RESPOND>** | **<WARNING>** | **<FATAL>**] *AttributeName* "*Arbitrary String Event Report*"

ppxAsyncMsg [Attribute-Value pair] [Attribute-Value pair]*

ppxAsyncMsg [**<WARNING>** | **<FATAL>**] "*Arbitrary String Event Report*"

The **PPX** announces there are 0 hours 0 minutes and 10.5 seconds left in the current integration

ppxAsyncMsg *IntTimeLeft=00:00:10.5*

The **PPX** announces it has rebooted due to a power glitch the upper level system must respond

ppxAsyncMsg <RESPOND> *PowerGlitch* "***PPX*** *system rebooted due to power loss*"

The upper level system should send the following command back to the **PPX**

ppxAsyncResp *PowerGlitch*

The upper level system is told the **PPX** has fallen and can't get up.

ppxAsyncMsg <FATAL> "*Help me Mister Wizard***."**

4.7.4 Respond to a PPX's Asynchronous Status Message - ppxAsyncResp **RULE**

This command is used in response to reports of events from the **PPX.** When the **PPX** sends a **ppxAsyncMsg** message to the upper level system, it may request, with a **<RESPOND>** directive, that a response be returned. This permits the **PPX** to confirm that a particular message has been received and normal operations can continue.

4.7.4.1 Acceptable Parameters **RULE**

This command will take a single attribute name as a parameter. The Attribute name will be the same attribute name which was sent to the upper level system by the **PPX** in the previous **ppxAsyncMsg** message.

4.7.4.2 Additional Actions **Permission**

An upper level system may take such additional actions as are necessary to return to normal processing after sending this message. The **PPX** implementers will need to spell out the actions the **PPX** will take when it receives a message of this type

4.7.4.3 Command Syntax RULE

ppxAsyncResp *AttributeName*

Example:

The upper level system is responding to a *powerGlitch* **ppxAsyncMsg** message

ppxAsyncResp *powerGlitch*

The **PPX** and upper level system will then perform any additional actions needed to return to normal operations.

4.8 Simulation and Debugging Commands COMMENTARY

The simulation and testing commands are provided for use during system development and integration and during the diagnosis of system failures and problems. These commands are not expected to be used during normal operations of a **PPX** system

4.8.1 Simulate - PPXSimulate RULE

This command will cause the system to enter a simulation mode. The designers of the **PPX** will determine what the valid simulation levels and subsystems should be for their system.

4.8.1.1 Acceptable Parameters RULE

The command will take Attribute-Value pairs which will determine at what level the simulation will take place and which subsystems of the **PPX** are to be simulated. The **PPX** must respond to at least the "all" subsystem and the TRUE and FALSE levels. This would indicate that only the command processor would be running and all other subsystems would be simulated.

4.8.1.2 Simulation Levels RECOMMENDATION

Some reasonable additional simulation levels might be HDWR (simulate the subsystem hardware, DETECTOR (simulate a Detector), ERROR (generate simulated errors). The nature of the simulation is left to the designers.

4.8.1.3 Simulation Levels RECOMMENDATION

It is usually useful for the system to use a simulation level to simulate hardware which is inoperative or nonexistent.

4.8.1.4 Simulation Display RULE

The **PPX** system must prominently indicate that it is in a simulation mode and exactly what is being simulated. Response to the user for commands to simulated subsystems should indicate the fact that the subsystem is a simulation of the real subsystem.

4.8.1.5 Command Syntax RULE

ppxSimulate [Attribute-Value pair] [Attribute-Value pairs]*

Example:

run only the software command processor runs.

ppxSimulate all=TRUE

simulate the entire embedded system hardware

ppxSimulate embededSystem=TRUE

simulate the fiber using a loopback mode

ppxSimulate Fiber=LOOPBACK

simulate the Detector head electronics hardware and software using a socket and the detConSim software simulator

ppxSimulate Fiber=SOCKET embededSystem="rsh decapod detConSim"

4.8.2 System Test Mode - ppxTestMode **RULE**

This command will cause the system to enter a hardware test mode. The designers of the **PPX** will determine what hardware test modes will be for their system.

4.8.2.1 Acceptable Parameters **RULE**

The command will take Attribute-Value pairs to determine the nature of the testing.

4.8.2.2 Minimum Response **RULE**

PPX systems must respond to at least the "all" test mode with either "OK" or "ERROR". If no test modes are supported, the **PPX** should respond with "OK - Test not supported".

4.8.2.3 Test Mode Display **RULE**

The **PPX** system should prominently indicate that it is in hardware test mode and what is being tested

4.8.2.4 Command Syntax **RULE**

ppxTestMode [Attribute-Value pair] Attribute-Value pairs]*

Example:

test all subsystems or respond with "OK - Test not supported"

ppxTestMode all=TRUE

Test the embedded system DACs

ppxTestMode hdwrDacs=TRUE

Test the fiber communications channel

ppxTestMode FiberChannel=TRUE

4.8.3 Dump Some Part of the System Information - ppxDump

5.0 PPX Error Detection and Recovery Behavior

TBD. Details to follow.

Appendix I Legacy Detector Controller Commands

This appendix outlines the command set of the Generic Pixel Server and attempts to cross reference those commands to those used by control some of the controllers in use in astronomy today. An effort was made to eliminate those commands that seemed to be for instrument-level control leaving only the commands which dealt with detector setup, readout and final pixel disposition and processing. Note that some of the systems such as the wildfire ICON board act as both array controller and instrument controller. When this was the case, the instrument control commands were left out of the table below. In addition in some systems such as ARCON, the integration level of array and instrument controller makes it difficult to separate the commands

Table 2 – Legacy Array Controller Command Sets

Generic Pixel Server Command	wildFire	SDSU-II	GEMINI IR Array Controller	Arcon
ppxSetMode	setup,		apply	
ppxSetArrConfig	Dsetup		ArSetup	
ppxSetExpConfig	ask, setRdd, setMode, setb011Mode		ObsSetup	
ppxSetIDPConfig	ask, setRdd, setMode, setb011Mod		ROI Setup	
ppxSetAVP	setVar, set*VoltName*, coAdds, Lnrs, digAvgs			
ppxStartExp	go		observe	
ppxArmExpTrigger				
ppxStop			stop	
ppxAbort	abort		abort	
ppxPause			pause	
ppxResume			continue	
ppx Shutter				
ppxShftImage				
ppxSimulate				
ppxTestMode				
ppxPower				
ppxReset			reboot init, reset	
ppxTestMode				
ppxGetState				
ppxGetAVP				
ppxAsyncMsg				
ppxAsyncResp				
ppxPass				
No equivalent or out of scope			park, guide, datum, endguide, endobserve	

Appendix II PPX Data Stream Description

"FITs" details to follow.

It is assumed that the bytes will be stored in memory with the lowest numbered byte of the message in the lowest memory address.

II.i Byte Order **RULE**

Bytes in a message are labeled from 0 to N (where N is the length of the message) all messages will use network byte ordering. The sending software will build the message using htonl() or htons() or similar routines to convert from host ordering to network ordering. The receiving software will convert the message byte order into something which is usable locally using ntohl() or nto hs() or similar functions.

II.ii Strings **RULE**

Strings which are embedded in a message will be inserted with the left most character of the string in the lowest order message byte. An example of this would be string "ABCD" appearing in message Byte n through n+3 with 'A' in byte n and 'D' in byte n+3.

II.iii Very Long Integers **RULE**

The protocol will make provision for 64 bit integer by using the network ordering decision used for long integers. In the messages the Most Significant Byte of a very long integer will be sent first and will be closest to the start of the message.

II.iv Floating Point and Double Values **RULE**

Real numbers represented by floats or doubles in 'C' will be represented in the **PPX** messages in IEEE floating point format (see Bold]Default ¶ Font). The byte ordering shall be as defined in the XDR/Network standard.

II.v Message Sructure

PPX data messages have the following structure:

Message Header **RULE**
- Message type tag - 1 Bytes
- Message version tag - 1 Bytes
- Message body length - 2 Bytes
- Message source ID - 4 bytes
- Exposure ID - 4 bytes
- Association ID - 4 bytes

Message body RULE
- A number of bytes equal to the message length. The contents of the bytes are interpreted by the receiving program In accordance with the message type and version.

Message error check RULE
- A four-byte CRC value calculated by doing a exclusive or of all the other bytes in the message in groups of four bytes.

II.vi Keyword Messages RULE

Keyword messages are used to inform the final user of the data as to the value of an attribute of the data. A keyword message consists of \

II.vii Header Block Messages

II.viii Event Messages

II.ix Pixel Block Messages

II.x Control Messages

Appendix III Response and Log Message Suggested Formats

It is important that error and log messages from the PAN software are easily parsed by the MONSOON supervisor layer or Engineering consoles. This means that they should be informative and without extraneous information. By the nature of the MSL or an engineering console, the source of the message is known because there is a 1 to 1 connection between the PAN and clients. It would also be nice to know when the message was produced. Below are some suggestions for the format of error and log messages.

III.i OK Messages RECOMMENDATION

OK messages are sent from the **PPX** in response to a command string which is valid and which was completed successfully. Two types of OK messages can be expected.

III.i.i Non-Information Request Commands

These commands do not require the return of specific information about the PAN system. Commands like **ppxStartExp**, **ppxReset, ppxAsyncResp** or **ppxSetMode** do not return any information on success. Responses from these commands should be structured as follows:

"**OK**: **ppx**COMMANDNAME: *Success message* [TIMESTAMP]"

The fields above are defined as:

"**OK:**" - the standard lead in for a successful valid command response message.

"**ppx**COMMANDNAME" - the name of the **PPX** command producing the response.

"*Success message*" - an indication of status, i.e. "DHE Reset Complete", "Mode set to fp_DefaultSetup", "Exposure Started", etc.

TIMESTAMP - an optional time stamp should make it easy to sort messages in time so the events during a session can be verified. The Monsoon Star Date should be used to record the response time to the nearest millisecond, i.e. 2345679.12345678

III.i.ii Information Request Commands

These commands require the return of specific information about the PAN system. Commands like **ppxSetAVP, ppxGETAVP** and **PPXGetState** will always return some information if successful. Responses from these commands should be structured as follows:

"**OK**: **ppx**COMMANDNAME: *Success message* [TIMESTAMP]\n"

The fields above are defined as:

"**OK:**" - the standard lead in for a successful valid command response message.

"**ppx**COMMANDNAME" - the name of the **PPX** command producing the response.

"*Success message*" - an indication of status, i.e. "intTime Set to 4.6 <0x11F8>", mcbCntrl-Reg=0x00300021", "actualIntTime=4.6<0x11F8>", "panState=PAN_ERROR" etc.

TIMESTAMP - an optional time stamp should make it easy to sort messages in time so the events during a session can be verified. The Monsoon Star Date should be used to record the response time to the nearest millisecond, as in 2345679.12345678

III.ii Error Messages RECOMMENDATION

Error messages are sent from the GPX in response to a command string whose command is invalid or which failed. They should be structured as follows:

"ERROR: **PPX**COMMANDNAME: *Long text reason for error* :[TIMESTAMP]\n"

The fields above are defined as:

"**ERROR**:" - the standard lead in for an error response message.

"**ppx**COMMANDNAME" - the name of the **PPX** command producing the response.

"*reason for error*" should be as explicit as possible, giving the acceptable range for OOR attribute values, detailing exactly why the command failed, etc.

E.g. "**ERROR**: **ppxSetAVP**: Attribute VggCl2 request out of range. desired=-8.9, allowed=-0.0 to -7.0\n"

TIMESTAMP - an optional time stamp should make it easy to sort messages in time so the events during a session can be verified. The Monsoon Star Date should be used to record the response time to the nearest millisecond, as in. 2345679.12345678

III.iii ASYNC Messages RECOMMENDATION

Async messages can be sent by the **PPX** software at any time to indicate that an event has occurred which may require the attention of the client software. The most common async messages are those sent upon completion of an exposure. ASYNC messages are formed by prepending "AsyncMsg" onto a standard OK or ERROR message:

"ASYNCMSG: OK: Exposure complete ExpID=2345453.26736556\n"

or

"ASYNCMSG: ERROR: Exposure Failed ExpID=2345453.26736556: Data transfer timeout -1307\n"

or

"AsyncMsg: ERROR: **PPX**CHKTELEMETRY: VggCl2=-6.5 volts This is out of range. requested=-5.5\n"

III.iv Log Messages RECOMMENDATION

Log messages can include a wide variety of messages from many sources. They should include sufficient identification so the events during a run can be reconstructed. They should be held to less than one line of text (80-120 characters). They should end with a newline (CR/LF sequence) to clearly mark the end of the message. They should be structured as follows:

"[TIMESTAMP] **DBG**|**LOG**: **PPX**CommandName:: *log message***n**"

The fields above are defined as:

"TIMESTAMP" - an optional time stamp, see definition in error message description above.

"**DBG**|**LOG**:" - an indication of the type of message

PANROUTINENAME: - the name of the PAN routine producing the message.

"*log message*" - this is the text of the log message. It should be as long as needed but every effort should be made to keep messages to less than 80-120 characters. The message should end with a new line.

E.g. "2652556.14434326: DBG: detCapture(orion_II): - received frame 32, using algorithm Jones-6\n".

Appendix IV Commands and Defined Variables and Parameters

Table 3 – Commands, Defined Variables and Parameters

Command Name	Parameters Set/Controlled	Usage/Explanation
ppxSetMode	All parameters settable from other Config commands	
ppxSetArrConfig	numArrays	An integer value giving the number of arrays in a focal plane.
	arrayDescriptor type rows columns outputsPerArray	A structure describing the characteristics of the array being controlled. The components are: a string giving the type of array, Two integers giving the size in rows and columns and an integer giving the number outputs on the array. Other elements may be needed for certain arrays.
	outputArrangement picQueue baseR, baseC strideR, strideC chunkR, chunkC sizeR, sizeC	A structure which outlines how the array outputs are read. This includes a queue descriptor which tells where to place the pixels for processing and information on the structure of the pixel data block transferred. We describe the block of data as chunks of contiguous pixels separated by a intervening pixels from other blocks. The block is described by a number of integers giving the starting row and column of the block, a row and column stride (the number of pixels to skip when storing chunks) the row and column chunk size and the total size of the final block of data in rows and columns.
	spcDescriptor gain settlingTime offset noiseFoM	A structure which describes the configuration of the signal processor chains in the system. The components of the structure are floating point arrays which describe the gain, settling time, and offset of the signal processing chain. Included is a noise figure of merit (TBD) which will allow the quietest set of chains to be chosen when that is important.
	waveForms	A descriptor for the timing waveforms to be run when running the array. These will be an array of bytes which will either describe of define the timing of the array readout. It is expected that each system will have an idiosyncratic way of describing these waveforms.
	DacValueN - float	An array of floating point voltage values which are to be loaded into any DAC settable voltages used to control the array. Each system will likely have a unique set of these voltages and a mapping from voltage name to DAC number should be provided in the **PPX.**
	Min Integration Time	A floating point number giving the minimum integration time achievable by the system.
	Base Readout Time	A floating point number giving the fastest possible readout time for the entire array.

Table 3 – Commands, Defined Variables and Parameters (Cont.)

Command Name	Parameters Set/Controlled	Usage/Explanation
ppxSetArrConfig (Cont.)	spdRoiDescriptor Row0 Col0 rowSize ColSize	A structure which describes a "speed up" region of interest (ROI). This is provided so a system with a large array can describe a sub array which will be readout to provide faster readout and shorter integration times. (Mostly used for IR systems without an internal cold shutter) The ROI is described by four integers giving the First row and column to be read and the size of the ROI in rows and columns.
ppxSetExpConfig	binning	An integer value giving the binning factor for the array readout. This may be two values if the row and column binning factors are different.
	intTimeSecs	A floating point number giving the desired integration time to use in seconds.
	digAvgs	An integer giving the Number of Digital Averages to use while reading out the pixels.
	numPics	An integer giving the Number of pictures to generate for each **PPX**StartExp.
	ROldescrptors Row0 Col0 rowSize ColSize	A list of structures defining the regions of interest (ROI) to readout and archive. The components of the structure are four values representing the first row and column included in the ROI and the row and column size of the ROI.
	Outputs to Read	An integer giving the number of outputs on the array which will be used during the readout.
	PreFlash	A Boolean value determining if the exposure sequence will include a pre-flash step.
	waveFormsToRun	A list of the waveforms to run during this array readout.
	shutterState	A Boolean value determining if the shutter is to be opened during the integration time.
	arrayPowerState	A Boolean value determining if the array will be activated/powered up during the exposure.
	intraPixelDelay	A floating point number giving the amount of time to allow for settling while reading each pixel.
	idleProcess	An integer tag describing how the array will be run during any idle time in the observing run.

Table 3 – Commands, Defined Variables and Parameters (Cont.)

Command Name	Parameters Set/Controlled	Usage/Explanation
ppxSetIDPConfig	Data Disposition - struct disposition - procedure name Arguments - filename, directory image format, data type, data stream/queue	
	Pre-processing Algorithm	
	Unscrambling Algorithm	
	Image Data Set ID	
	coAdds - integer	
	fSamples - integer	
ppxSetAVP	Every user-settable attribute	
ppxStartExp	binning - Integer	
	intTimeSecs – float	
	digAvgs - integer	
	numPics – integer	
	shutterState	
	arrayPowerState	
	Data Disposition	
	Pre-processing Algorithm	
	Unscrambling Algorithm	
	Image Data Set ID	
	coAdds – integer	
	fSamples - integer	
ppxArmExpTrigger	TriggerSource	
	TriggerTime Out	
ppxStop	Image Data Set ID	
ppxAbort	None	
ppxPause	intTimeSecs – float	
ppxResume	intTimeSecs – float	
ppxShutter	Current shutter state	
ppxShft/Image	Row or Y shift	
	Column or X shift	

Table 3 – Commands, Defined Variables and Parameters (Cont.)

Command Name	Parameters Set/Controlled	Usage/Explanation
ppxSimulate	Units to simulate	
ppxTestMode	Units to test	
ppxPower	System Power State	
ppxReset	System Reset Level	
ppxGetState	Reads all system state attributes	
ppxGetAVP	Reads every individual system attribute	
ppxAsynchMsg	Reports asynchronous events and errors	
	Asynchronously reports attribute values	
ppxRstAsynStatus	Resets asynchronous event and error flags	
ppxPass	Passes string command to underlying system	

Appendix V – Attribute-Value Pair Notation Conventions

coAdds=16 - set attribute coadds to the value 16

integration+=10.0 - add 10 seconds to the integration time

fSamples-=8 - reduce the number of fowler samples by 8, if the result is invalid make the number the minimum.

Appendix VI – Configuration File Format

A configuration file is made up of a number of configuration lines organized into sections. The format of the file is described below in a modified BNF grammar. The BNF grammar is modified by allowing text description of certain entities. For the purposes of this format description the following conventions are used:

- Defined entities are described on the left side of a :: symbol e.g.

sectionName :: **[** { **GENERAL** | **ARRAY_VOLTAGES** | **ARRAYCLOCKS** | **VIDEOCHANNELS** | **READOUTPARAMS** | **EXPOSUREPARAMS** | **DATAPREPROCESSING** } **]** **\n**

- **Items in Bold** indicate that the exact bold text is to appear in the configuration file. **\n** in a description indicates the presence of a newline character or CR-LF sequence in the file.
- Items in *Italics* indicate defined entities in the file description, as in, *SectionName* or Attribute-Value*Pair*. Thus

section :: *sectionName* [*descriptionLines*]$^{+}$

- If two symbols appear in a definition following each other it is assumed they occur in that order in the actual configurations file. Thus

configurationFile :: *modeIDLIne* [*section*]$^{+}$

would indicate that a configurationFile consist of a mode id line followed by one or more sections.

- Items enclosed in curly braces and separated by vertical bars i.e. { A | B } indicate a choice. Thus **[** {**TRUE** | **FALSE**} **]** indicates that either TRUE or FALSE would appear between the square braces.
- Items in square braces i.e. [A] indicate optional items that may appear in the file or may be omitted.
- Items in square braces with a asterisk or plus sign super script indicate 0 (*) or 1(+) or more of the optional items. Thus [*Atribute-Value Pair*]$^{+}$ indicates the presence of one or more attribute value pairs.

The description of the Configuration File follows:

configurationFile	::	*modeIdLine fileSections*
modeIdLine	::	**ModeName =** *FileName* **\n**
fileSections	**::**	**[** *generalSection array*] [*VoltageSection*] [*arrayClockSection*] [*videoChannelSection*] [*readoutParamsSection*] [*exposureParamSection*] [*dataPreProcessSection*]
generalSection	::	**[GENERAL] \n** *sectionLines*
VoltageSection	::	**[ARRAYVOLTAGES] \n** sectionLines
arrayClockSection	::	**[ARRAYCLOCKS] \n** sectionLines
videoChannelSection	::	**[VIDEOCHANNELS] \n** sectionLines
readoutParamSection	::	**[READOUTPARAMS] \n** sectionLines
exposureParamSection	::	**[EXPOSUREPARAMS] \n** sectionLines
dataPreProcessSection	::	**[DATAPREPROCESSING] \n** sectionLines
sectionLines	::	[sectionConfigurationLines]* [AttributeValueLines]*

Appendix VII – Memory Configuration File Format

In order to quickly load setup variables or to load the Waveform sequencer which controls the array readout, the **PPX** command set provides the **ppxSetMemory** command as part of the PAN/DHE configuration command suite. This command takes the name of a file which is assumed to be a download file. The structure of the download file is described in this Appendix.

The description of the download file format follows the conventions described above in Appendix VI**Error! Bookmark not defined.** for configuration files.

The description of the Download File format follows:

configurationFile	::	*dwnLdLinesList*
dwnLdLineList	::	*fileLine* [*dwnLdLineList*]
fileLine	::	{ *commentLine* | *addressLine* | *valueLIne* | *blankLine* }
commentLine	::	**#** *commentString* **\n**
addressLine	::	**addr=** { *attributeName* | *absoluteAddress* } [# commentString] **\n**
valueLine	::	v*alue* [*value*]* [# commentString] **\n**
blankLine	::	**\n**
commentString	::	*char* [*char*]*
attributeName	::	any valid attribute name in the configuration file for the current system.
absoluteAddress	::	{ *decimalAddr* | *hexAddr* }
value	::	{ *decValue* | *hexValue* } { **\b** | **\n** }
decimalAddr	::	a decimal number in the range 65536 to 134283263
hexAddr	::	a hexidecimal format number in the range 0x10000 to 0x0800FFFF
decValue	::	a decimal number in the range 0 - 65535
hexValue	::	a hexidecimal format number in the range 0x0000 to 0xFFFF

Example:

```
############################################################
## PROJECT    : ALADDIN
## TITLE      : SEQUENCER MPU LOAD FILE FOR ALADDIN TESTS
## VERSION    : 003
## DATE       : 08/06/2003
## AUTHOR     : KAVI CHOPRA,JOHN GARCIA,PETER MOORE,GUSTAVO RAHMER
## HISTORY    : 07/11/2003  ORIGINAL VERSION.
##            : 07/21/2003  sources aligned to /monsoonhome/aladdin_test.
##            : 07/25/2003  Version for Fowler sampling + Digital averages
##            : 08/06/2003  Allows normal and row reset reads
## DESCRIPTION : LOADS THE SEQUENCER MPU PROGRAM AND PATTERN MEMORY
##            : SEGMENTS FOR THE ALADDIN II ARRAY.
## RESOURCES  :
############################################################
```

```
# ALADDIN PATTERN MEMORY SEGMENT
# First Give start address
#addr=0x00011000        # you can use absolute address or
#addr=MCB_SEQPATMEM # DHE attribute name or
addr=mcbSeqPatMem# PAN attribute name
# Now list the values to be loaded into the memory
# you can list one per line or several per line
0x0001 0x0000 0x0002 0x0000 0x0004 0x0000 0x0000 0x0000
0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000
0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000
0x2540
53 32456 18
```