MONSOON
Generic Pixel Server
Communications, Command/Response
and Data Stream Interface Description

NOAO Document ICD 4.0
Revision: 1.1

Author by:
Nick C. Buchholz, Phil N. Daly and Barry M. Starr
1/7/2002
Please send comments:
nbuchholz@noao.edu

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

<table>
<thead>
<tr>
<th>Version</th>
<th>Date Approved</th>
<th>Sections Affected</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0.1</td>
<td>3/11/2002</td>
<td>All</td>
<td>First release draft</td>
</tr>
<tr>
<td>0.0.2</td>
<td>4/12/2002</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>8/8/2006</td>
<td>All</td>
<td>Reformat and edit – aro</td>
</tr>
<tr>
<td>1.1</td>
<td>10/24/2006</td>
<td>Appendix II</td>
<td>Added information</td>
</tr>
</tbody>
</table>
# Table of Contents

Revision History ................................................................................................................. 2  
Table of Contents ............................................................................................................... 3  
1.0 Introduction ............................................................................................................... 5  
1.1 Scope ....................................................................................................................... 5  
1.2 Purpose ..................................................................................................................... 7  
1.3 Reference Documents .............................................................................................. 7  
1.4 Acronyms and Glossary ............................................................................................ 8  
1.4.1 Abbreviations and Acronyms ............................................................................. 8  
1.4.2 Glossary ............................................................................................................. 9  
1.5 Standard Terminology ........................................................................................... 11  
1.6 What’s in This Document ..................................................................................... 12  
1.7 What’s NOT in This Document ......................................................................... 12  
1.8 Other Assumptions ............................................................................................... 13  
2.0 Command and Data Communications Structures ............................................. 14  
2.1 Command/Response Communication Stream Definitions ................................... 14  
2.2 Socket Definitions ................................................................................................. 14  
2.3 Status and Data Stream Interface ....................................................................... 15  
2.4 Status and Data Stream Message Protocol ...................................................... 16  
3.0 Generic Pixel Server Command Set .................................................................... 16  
3.1 System Identification ......................................................................................... 16  
3.2 Command Structure .......................................................................................... 16  
3.3 Whitespace Ignored ............................................................................................ 17  
3.4 Mode/Configuration Commands ....................................................................... 18  
3.5 Exposure Sequence Control Commands ......................................................... 26  
3.6 Internal Action Commands ............................................................................... 30  
3.7 Status and Information Commands ................................................................. 33  
3.8 Command Pass Through .................................................................................... 38  
3.9 Simulation and Debugging Commands ........................................................... 39  
4.0 GPX Error Detection and Recovery Behavior ................................................. 41  
Appendix I Legacy Detector Controller Commands ........................................... 42  
Appendix II GPX Data Stream Description ........................................................... 43  
II.i Byte Order ........................................................................................................ 43  
II.ii Message Structure .............................................................................................. 43  
Appendix III Response and Log Message Suggested Formats ......................... 45  
III.i OK Messages ........................................................................................................ 45  
III.ii Error Messages ................................................................................................... 45  
III.iii Log Messages ..................................................................................................... 46  
Appendix IV Commands and Defined Variable and Parameters ....................... 47  
Appendix V Attribute-Value Pair Notation Conventions .................................... 51  
Appendix VI Configuration File Format ................................................................. 52
List of Figures

Figure 1 - Observatory System Reference Model .......................................................... 6
Figure 2 - System Context Diagram ............................................................................. 6

List of Tables

Table 1 - Legacy Array Controller Command Sets .......................................................... 42
Table 2 - Commands and Defined Variables and Parameters ........................................ 47
1.0 Introduction

1.1 Scope

The Generic Pixel Server Interface, as discussed at the November 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 pre-process the data, and send the data to an archive system for later analysis.

The Generic Pixel Server, (GPX, since GPS is already taken), concept combines several layers in an observatory system reference model. Figures 1 and 2 depict two different views of where the GPX fits in the observatory system. In Figure 1, the GPX is a subordinate or subsystem of the instrument and instrument control system. In Figure 2, both the instrument and the GPX are subsystems of the observatory control system as a whole. In either case the interface to the GPX can be clearly defined by this ICD.

The GPX provides for access to and control of the data generation and capture hardware. It combines the functionality inherent in the detector head electronics (detector controller) and data pre-processor node. This ICD then presents a common interface to the upper levels of the observatory system regardless of the underlying detector technologies being used for observing. In this description, note that the Generic Pixel Server (GPX) does not control the telescope or the instrument mechanisms. Other layers of the observatory system handle those jobs. The GPX manages taking and controlling exposures. It may, if one is provided, control a local shutter.

Also note that the GPX interface is an interface 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 Generic Pixel Server. It contains a communications and connection specification, a command and response specification and a pixel and status data stream proposal. Readers of this document and their colleagues are asked to review the interface and return comments about its completeness and usability.
Observatory System Reference Model One
Figure 1

Observatory System Reference Model Two
Figure 2
1.2 **Purpose**

This Interface Control Document (ICD) serves four purposes:

- To describe the nature of the communications interface between a Generic Pixel Server (GPX) Compliant Pixel Server System and external systems.
- To describe the parameters that will pass between the Generic Pixel Server (GPX) and external systems.
- To describe the behavior of the Command/Response interface between a Generic Pixel Server (GPX) Compliant Pixel Server System and the external observatory systems which will control it.
- To describe the behavior of the pixel data interface between a Generic Pixel Server (GPX) Compliant Pixel Server System and Data Handling systems.

The intended audience for this document is:

- The NEWFIRM instrument group
- The developers of any instrument which plans to use a Monsoon Pixel Server System
- The Monsoon Pixel Server System group
- The developers of any instrument which plans to use any Generic Pixel Server (GPX) Compliant Pixel Server System
- The developers of any Generic Pixel Server (GPX) Compliant Pixel Server System
- Anyone else with an interest in the Generic Pixel Server Concept.

1.3 **Reference Documents**

1. SPE-C-G0037, “Software Design Description”, Gemini 8m Telescopes Project.
2. “ICD/16 - The Parameter Definition Format”, Steve Wampler, Gemini 8m Telescopes Project.
3. WHT-PDF-1, “FITS headers for WHT FITS tapes”, Steve Unger, Guy Rixon & Frank Gribbin, RGO.
4. NOST 100-1.0, “Definition of the Flexible Image Transport System (FITS)”, NASA Office of Standards and Technology.
5. GEN-SPE-ESO-00000-794, “ESO Data Interface Control Document”, Miguel Albrecht, ESO.
8. NOAO Document MNSN-AD-01-0004 - ICD 6.0 Version 0.1.2 - “Generic Detector Controller - Command and Data Stream Interface Description”, Nick C. Buchholz(NOAO), Barry M. Starr(NOAO), 20020308
1.4 Acronyms and Glossary

1.4.1 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
MONSOON  Not an acronym
NICD  NOAO Interface Control Document
IAS  Image Analysis System
ICS  Instrument Control System
IDPS  Image Data Preprocessor System
ID  Identifier
IR  Infrared
LAN  Local Area Network
N/A  Not Applicable
OCS  Observatory Control System
PDF  Parameter Description File
ROI  Region of Interest
SUS  Status Update System
TBD  To Be Decided
1.4.2 Glossary

**Attribute** - An entity which describes some aspect of the configuration of a Pixel Server System, such as the Level of a voltage or the state of a shutter. Some attributes will be used by the Pixel Server System as command parameters. The OCS communicates with a science instrument by sending it sets of “attributes” and “values”.

**Command** - An instruction commanding 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”) which contain the details of the instruction to be obeyed.

**Pixel Acquisition Node** - A component of a Generic Pixel Server, this is the computer which handles the interface to the detector head electronics and the image pre-processing of the data stream from the detector head electronics.

**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.

**Exposure** - The name used to describe the process and the data resulting from the process of resetting/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 which results when a single Reset-Readout-Integrate-Readout cycle is performed on an IR detector or a single CCD Clear-Integrate-Readout cycle.)

**Exposure Sequence** - The process by which valid data is produced. Various levels of exposure sequencing occur during an observing run. At the lowest level there are the Reset-Readout-Integrate-Readout or Clear-Integrate-Readout cycles which result in a single IR or OUV exposure. At the highest level are the observing sequences which move the telescope, configure the instrument and take a series of exposures which create an observation.

**Frame** - A frame is the result of one or more readouts of an array averaged pixel by pixel. 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.

**Image** - The array of detector pixel and description data representing a science or diagnostic image or spectrum. 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 coadded and averaged exposures in one beam of a chop mode gpxStartExp command).

**Observation** - The process of exposing the detector to photons in one or more exposures. The result of an observation is a picture.
1.4.2 Glossary (Cont.)

**Read** - When used as a noun to describe instrument data, this refers to a single read of a pixel on the detector. A read may consist of several A/D conversions of the pixel data which 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.

**Readout** - When used as a noun to describe instrument data, this refers to a single read of every pixel in the detector. One or more readouts can be averaged pixel by pixel to create a frame.

**Region of Interest (ROI)** - A sub array of the available detector area. There are two types of sub arrays which 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 which fits on the Array. Data Reduction ROI’s 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”.

**Detector Head Electronics** - The lowest level hardware system, normally closely connected to the detector and the dewar in which the detector resides.

**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 System** - The combination of the Detector Head Electronics and a Pixel Acquisition Node which are coordinating the task of taking exposures and archive the resulting data set.

**Pixel Server** - A system which produces pixel values when requested to do so by some client system.

**Generic Pixel Server** - A pixel server that conforms to the GPX Interface description.

**Supervisory Node** - A computer capable of controlling multiple Image Acquisition systems. The computer that runs the software that conforms to the GPS interface.
1.5 **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 listed below:

- Rule
- Recommendation
- Suggestion
- Permission
- Observation

These keywords are used as follows:

**RULE**

<Paragraph Number> Subject Describing Text

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**

**RECOMMENDATION**

<Paragraph Number> Subject Describing Text

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:

3.2.1 **GPX Names**
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 need to be made are difficult until experience has been gained. Suggestions are included to help a designer who has not yet gained this experience.

Example:

2.2.4 Long Variables Names  SUGGESTION

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:

3.2.1 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:

2.2.5 Long Variables Names  OBSERVATION
1.6 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 represent the GPX’s interface to external systems. The interface allows external systems to command and control a GPX compliant system. 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.

We have mapped our command set to the low level commands used by three legacy detector controllers (wildFire, SDSU-II, GNAAC) and to the internal command set proposed for use by the Monsoon Image Acquisition System being developed by NOAO. (See Appendix I for a comparison.) We think provision has been made for each of the reviewed controllers to be fit into the Generic Pixel Server model. We also believe we can extend the generic command set to deal with future detector controller systems should it be necessary.

1.7 What’s NOT in This Document

Several things have been left out of this document. First, no attempt is made in this document to propose the low level, internal command set of the Pixel Acquisition Node and Detector Head Electronics. However, an example Generic Detector Head Electronics ICD has been developed to be used by the Monsoon Detector Head Electronics. [9]

Second, the responses and internal behaviour of the GPX in the face of errors are left undefined at this time. We assume that each command issued will generate a single ascii response and that at a minimum the GPX will return “OK” for a successful command and “ERROR” if a command fails in some way. In the case of configuration and sequence commands whether the GPX halts on the first error or attempts to continue will be left for later discussion by system developers.

Third, no assumption is made about the internal communications protocols or the nature of the internal command passing techniques, except that it is assumed that the communications protocol will be some form of byte stream. In particular, the internal hardware interconnects and data passing techniques are explicitly excluded from this discussion.

1.8 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 Generic Pixel Server 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 insure that the user does not spend time taking simulated data while believing real data is being gathered.
2.0 Command and Data Communications Structures

2.1 Command/Response Communications Stream Definitions

The Command/Response communications streams used by the Generic Pixel Server take place over a *socket* connection. The control layers which use the GPX connect to the GPX by connecting to an IP address and socket port number.

2.1.1 Socket Connections

While it is not necessary that a socket connection be used, it will simplify the system to make this a rule. It is possible to serialize the command response streams in another way i.e. as a result of an RPC call, OS driver call or a serial hardware line. However, the consensus is that the *socket connection* is the most reliable and OS independent method to use.

The command/response set only assumes that the underlying communications protocol to the Pixel Server delivers a single ASCII string to the command processor for each command issued.

2.2 Socket Definitions

The connection to a GPX will be by a set of socket connections. Assuming the primary communication socket (commands) is on Port N (TBD), a second socket on port N+1 may be used for the command response stream and a third, on port N+2, may be used for the Asynchronous Status message stream.
2.2.1 Primary Upper Level System
The first connection to the GPX system sockets listed above will be the primary upper level system. It will be in control of the GPX and will be the only system that responds to any gpxAsyncStatus messages received.

2.2.2 Secondary Upper Level System
Additional upper level systems that connect to the GPX system sockets are considered secondary systems. They will see all activity initiated by the Primary system but will only be able to issue commands to the GPS under password control.

2.3 Status and Data Stream Interface
The status information and data output from the GPX shall be a stream of messages across a socket connection. Any client wishing to use the data produced by the GPX may connect to this socket and receive the message stream.

2.3.1 Data Output Minimum
Every GPX will, at a minimum, have a mode that is able to produce a FITS formatted image on a local disk for engineering and diagnostic purposes. This capability should conform to the data interface API described in ICD 1.0, “Data Handling System Interface”.

2.3.2 Status Output Minimum
Every GPX will, at a minimum, have a mode that allows the display of status information as lines of status messages on a terminal- like display.

2.3.3 Local Client / Remote Client Message Formats
The message formats, for some messages, will differ based on whether the client is connected locally or remotely. Local clients, running on the same machine as the GPX, 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
The final data stream from the GPX will be determined by local protocols.

2.3.5 Data Stream Minimum
It is possible for an implementation GPX to produce a stream of FITS images to export the data. This stream could then be transformed into the GPX standard format or into some local format such as Gemini DHS, NOAO PicFeed, IDL and others by an auxiliary process.
2.4 Status and Data Stream Message Protocol

The status and data stream message protocol is defined in detail in ICD 1.1, “DHS Interface - Input - Status and Data Stream Description”. The implementation and details of this protocol is considered to be internal to the DHS system being used and is left to the developers of the DHS systems. In general, the protocol uses messages to send pixel, status and event data to the outside world. The messages are considered to be a stream of bytes that are interpreted by the receiving process/client.

The GPX definitions for the data and status output are limited to those described in preceding paragraphs 2.3.1 through 2.3.5.

3.0 Generic Pixel Sever Command Set

The external command set proposed here is designed to allow full flexibility in the internal and functional design of the pixel server system. The commands break down categories as follows: configuration commands; exposure control commands; internal action commands; status request commands, and a generic 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.

3.1 System Identification

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

3.1.1 GPX Names

The identifier assigned to a GPX should be associated with the current function of that GPX. Thus if a R&D Lab Pixel Acquisition Node has name DHCP-150-5, it should also be assigned rndLab150-5. The NEWFIRM Pixel Acquisition Node might be assigned the name newfirmMonsoon and an IP address appropriate to the location.

3.2 Command Structure

All commands are delivered to the GPX as an ASCII string.

3.2.1 Command Length

These strings may be as long as necessary to perform the desired function. At a minimum, the GPX should be able to accept a command which is as much as 1000 characters long.

3.2.2 Command Length

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

GPX Commands have the following general syntax:

```
[CMDIDT] gpxCommandName [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.

**NOTE:** A closing newline or CR/LF may be required to distinguish between commands sent over certain interfaces. Socket read routines may require the use of a newline or CR/LF to terminate the read.

3.3 Whitespace Ignored

Whitespace within a command is ignored. This includes all TAB, SPACE and newline characters. Thus the following two commands are equivalent:

```
“gpxSetMode <SAVE> newModeFileName”
“ gpxSetMode <SAVE> newModeFileName”
```

3.3.0.1 Command Identifier Tag (CMDIDT)

The first six characters of the command string may be a command ID determined by the controlling system. The **GPX** will tag all responses to a command containing an ID string with the command ID string.

3.3.1 Command Directives

Several commands in the **GPX** 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 (<>)(the greater than, less than symbols.)

3.3.1.1 Ignoring Directives

Commands that do not accept directives may ignore a directive that 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 included with the command description.

3.3.2 Command Behavior

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

3.3.2.1 Rejecting Commands

A **GPX** may reject commands that arrive while it is busy. However, certain commands (noted in the command explanation) may not be rejected because the **GPX** is busy. As an example, a second `gpxStartExp` may be rejected if a current exposure is in process. However, a `gpxAbortExp` may not be rejected.

3.3.2.2 Multiple Commands

A **GPX** may “queue up” commands to be executed in order. Commands that may not be rejected because the **GPX** is busy are also not permitted to be “queued up”.

---

Doc. File: ICD_4.0_GPX_Interface_v1_1.doc
Doc. Number MNSN-AD-01-0002
Created on 10/24/2006
3.3.2.3 Response Time RULE
Each command response should be returned no more than 150 milliseconds after the command is received.

3.3.2.4 Minimum Response RULE
The minimum response to a command will be the strings “OK” or “ERROR”.

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

3.3.2.6 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.

3.4 Mode/Configuration Commands COMMENTARY
The first commands described 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 gpxSetMode command, which should be able to handle the complete configuration of a Generic Pixel Server, three redundant commands, gpxSetArConfig, gpxSetIDPConfig and gpxSetExpConfig are provided to configure parts of the overall system. Finally a gpxSetAVP command is provided to set individual attributes of the GPX.

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.

3.4.1 Configuration Files RULE
The basis of the configuration commands is the existence of configuration files. These files should determine the configuration of all attributes in the GPX required to take data in a certain configuration.

3.4.1.1 Default Configuration File RULE
Each instance of a GPX system must have at least one configuration file called “xxxDefault”. Where the xxx represents the instance name of the GPX. For example, the Monsoon Pixel Server system for NEWFIRM might have a file called “NewFirmGPXDefault”. The Lab system for testing ORION IR arrays might be called “orionLabDefault”.

3.4.1.2 Default file Content RULE
The default configuration file should include information which allows a gpxSetMode command to bring up the system in a configuration that is safe for the electronics and the array.
3.4.1.3 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 GPX implementation should know how to convert from the configuration command line to its internal command format.

3.4.1.4 Configuration File Content OBSERVATION
Since the implementers of a specific GPX 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.

3.4.1.5 Configuration File Format RULE
The format of the configuration files is defined in detail in Appendix VI. Briefly, the configuration file consists of attribute-value pairs to be set by the configuration command and gpxCommands to be executed. These are organized into sections that relate to various portions of the GPX internal structure.

3.4.1.6 Configuration Sections RECOMMENDATION
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.

3.4.1.7 Configuration Sections COMMENTARY
Following 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 GPX. Information like the IP address of the GPX machine, socket port numbers to use, the string name of the GPX system. etc.
- [ARRAY_VOLTAGES] - contains information about the bias and clock voltage settings for running the Array. It would also include the voltage number to voltage name conversion.
- [ARRAY_CLOCKS] - contains information about the clock patterns to be used to run and readout the array.
- [VIDEO_CHANNELS] - contains information about the individual video channels in the system, offset levels, gains for each channel, and so forth.
- [READOUT_PARAMS] - contains information about how the Array will be readout, binning, number of digital Averages, Fowler Samples to be done, and so forth.
- [EXPOSURE_PARAMS] - contains information about individual exposures, integration time, Shutter State during exposure, and other exposure information.
- [DATA_PREPROCESS] - contains information about how the data is to be processed, number of Coadds, processing algorithm, data disposition, Exposure ID, Association ID, and so forth.
3.4.1.8  Partial Configuration Files  
Configuration files may contain some none or all of the defined sections. Commands and Attribute-value pairs that are listed outside of a section are processed by all of the configuration commands. Each configuration command (except gpxSetMode) handles only some of the configuration file sections. Which sections are handled by each command is specified in the command descriptions that follow.

3.4.1.9  Building Configuration Files  
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.

3.4.1.10 Building Configuration Files  
The development team for a particular GPX may restrict the use of the <SAVE> directive and may restrict the observer’s ability to modify standard configuration files.

3.4.1.11 Building Configuration Files  
The development team for a particular GPX should always provide some easy to use mechanism which allows an observer to save configurations and to return to a previously saved configuration.

3.4.2  Set Pixel Server Mode – gpxSetMode  
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.

3.4.2.1  Config File Sections  
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.

3.4.2.2  Acceptable Parameters  
The gpxSetMode 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.

3.4.2.3  Acceptable Attribute Value Pairs  
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 gpxSetXxxConfig 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.

3.4.2.4  Supported Directives  
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 GPX 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.
3.4.2.5 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 a password or confirmation question.

3.4.2.6 Preserving Default Files  
**PERMISSION**

A GPX 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 in files that cannot be easily modified.

3.4.2.7 Command Syntax  
**RULE**

`gpxSetMode [<SAVE>] cfgFileName [arrConfig=arrFilename] [expConfig=expFileName] [idpConfig=idpFileName]`

Examples:

- To set the GPX configuration to the state described by the file `ORION135`.
  
  `gpxSetMode ORION135`

- Save the current configuration in the file `ORION135`.  (note: security concerns)
  
  `gpxSetMode <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`
  
  `gpxSetMode ORION135 expConfig="myExposure.Obs"`

3.4.3 Set Array Configurations – `gpxSetArrConfig`  
**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 be 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 over-ride the default settings in the configuration file or the current settings system settings if the “-” parameter is used.

3.4.3.1 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.

3.4.3.2 Config File Sections  
**RULE**

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

3.4.3.3 Acceptable Parameters  
**RULE**

The `gpxSetArrConfig` 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 GPX to put the array into the desired exposure mode.

3.4.3.4 Acceptable Attribute Value Pairs  
**RULE**

This command can accept optional attribute-value pairs that can be set from the command line. Any Attribute-value pair that 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.
3.4.3.5 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.

3.4.3.6 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 confirmation question.

3.4.3.7 Preserving Default Files PERMISSION
A GPX 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.

3.4.3.8 Command Syntax RULE
SetArrConfig [<SAVE>] {arrModeFileName | -} [attribute-value pairs]*

Examples:
- 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 GPX 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.
  
  SetArrConf - VggCL1=-4.6

3.4.4 Set Exposure Configuration - gpxSetExpConfig RULE
This command puts the GPX 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 over-ride the default settings in the configuration file named or the current settings system settings if the “-” parameter is used.

3.4.4.1 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.
3.4.4.2 Config File Sections  
RULE  
This command will not handle commands and Attribute-value Pairs included under other sections of the configuration file.

3.4.4.3 Acceptable Parameters  
RULE  
The gpxSetExpConfig 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 GPX to put the array into the desired exposure mode.

3.4.4.4 Acceptable Attribute-value Pairs  
RULE  
This command can accept optional Attribute-value pairs that can be set from the command line. Any Attribute-value pair that 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.

3.4.4.5 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.

3.4.4.6 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.

3.4.4.7 Preserving Default Files  
PERMISSION  
A GPX 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.

3.4.4.8 Command Syntax  
RULE  
gpxSetExpConfig [<SAVE>] {expConfigFileName | -} [attribute-value pairs]*

Examples:
- load the exposure configuration stored in the file “StandardStar”
  gpxSetExpConfig StandardStar

- load the exposure configuration stored in the file “NGC1721” overriding the integration time.
  gpxSetExpConfig NGC1721 integration=65.345

- set the integration time, arrPower and fSamples values to the desired values.
  gpxSetExpConfig - integration=65.345 arrPower=On fSamples=16

- Save the current configuration in the file “myConfig” overriding the current integration time
  gpxSetExpConfig <SAVE> myConfig integration=20.0
3.4.5 Set Image Data Pre-processor Configuration – gpxSetIDPConfig

This command puts the GPX 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 override the default settings in the configuration file named or the current settings system settings if the “-” parameter is used.

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

3.4.5.2 Config File Sections
This command will not handle commands and Attribute-value Pairs included under other sections of the configuration file.

3.4.5.3 Acceptable Parameters
The gpxSetExpConfig command takes a single parameter that 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 GPX to put the array into the desired exposure mode.

3.4.5.4 Acceptable Attribute-value Pairs
This command can accept optional Attribute-value pairs that can be set from the command line. Any Attribute-value pair that 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.

3.4.5.5 Supported Directives
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.

3.4.5.6 Preserving Default Files
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.

3.4.5.7 Preserving Default Files
A GPX 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.
3.4.5.8 Command Syntax RULE

gpxSetIDPConfig [ <SAVE> ] \{ idpConfigFileName | - \} \{ attribute-value pairs \]*

Examples:

- Load the Data Pre-processing configuration stored in the file “standardStar”
  gpxSetIDPConfig standardStar

- Load the configuration stored in the file “standardStar” overriding the value of destination
  gpxSetIDPConfig standardStar destination=DHS

- Set directory, file, coadds and procAlgorithm attributes to the desired values.
  gpxSetIDPConfig - directory="/home/mrEngineer/20011127" file="ngc1721" coadds=16
  procAlgorithm=SUR

- Save the current Data Pre-processing configuration stored in the file “standardStar”
  gpxSetIDPConfig <SAVE> standardStar

3.4.6 Set Attribute Value Pair – gpxSetAVP RULE

This command is provided to allow a user to set an arbitrary individual parameter in the GPX. 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.

3.4.6.1 Acceptable Parameters RULE

The gpxSetAVP command takes as a parameter a single attribute-value pair to be set.

3.4.6.2 Acceptable Attributes PERMISSION

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

3.4.6.3 Acceptable Attributes COMMENTARY

The gpxSetAVP 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 GPX may be protecting this attribute from change. Fourth, there may be a hardware failure.

3.4.6.4 Acceptable Attribute-value Pairs RULE

This command can accept optional attribute-value pairs that can be set from the command line. Any Attribute-value pair that can be set in the GPX can appear on the command line as an optional Attribute-value pair.

3.4.6.5 Supported Directives RULE

This command supports no directives.

3.4.6.6 Command Syntax RULE

gpxSetAVP AttribName=Value [Attribute-value Pair]*

Examples:

- set fSamples to 16
  gpxSetAVP fSamples=16

- set the attributes coadds, fSamples, integrationTime and imageTitle to the values listed
  gpxSetAVP coadds=32 fSamples=8 integrationTime=40.5 imageTitle="NGC2123 K' Filter"
3.5 Exposure Sequence Control Commands

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

3.5.0.1 Optional Directives

These commands accept no optional directives.

3.5.0.2 Minimum Exposure Sequence

The **GPX** must, at a minimum, be able to sequence the series of steps required to produce the data that 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.

3.5.0.3 Minimum Data Processing

The **GPX** must be capable of performing the data pre-processing 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.

3.5.0.4 Unimplemented Commands

Some detectors and **GPX** systems may have physical characteristics that make some of the Exposure Sequence Control Commands less than useful. For example, IR systems generally do not have shutters that allow the `gpxPauseExp` or `gpxResumeExp` to be used in a meaningful way. These systems may ignore the meaningless commands.

3.5.0.5 Unimplemented Commands

Systems that ignore commands must include the fact that the command was ignored in the command response. The response to an ignored `gpxPauseExp` command might be “OK - gpxPauseExp ignored.

3.5.0.6 Optional Command Attribute-value Pairs

All of the Exposure Sequence Control Commands listed in this section (i.e. 3.4.x) 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.

3.5.0.7 Optional Command Attribute-value Pairs

If the processing of an optional attribute-value pair fails, the associated Exposure sequence command should also fail.

3.5.1 Start Exposure - gpxStartExp

The `gpxStartExp` command will start an exposure sequence using the current configuration. In this context of the observing sequence, this means the **GPX** performs all tasks under its control that are required to generate a single exposure data set or multiple exposures using the same configuration. As an example, a **GPX** for a CCD may clear the array, open the Shutter, wait the integration time, close the shutter and read out the array. An IR **GPX** may reset the array and produce the appropriate number of Fowler sample frames as directed by the configuration. In either case, the data pre-processing and storage side of the system would produce a single data set.
3.5.1.1 Exposure Start Timing
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 gpxArmExpTrigger.

3.5.1.2 Command Syntax
\texttt{gpxStartExp} [ attribute-value pairs]*

\textbf{Examples:}
take an exposure overriding the current integration time.
\texttt{gpxStartExp} integration=10.0
take an exposure overriding the current integration time and numPics values.
\texttt{gpxStartExp} numPics=10 integration=10.0

3.5.2 Wait for Exposure Trigger - gpxArmExpTrigger
The \texttt{gpxArmExpTrigger} 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.

3.5.2.1 Acceptable Parameters
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 gpxArmExpTrigger command. These attributes could be configured before the gpxArmExpTrigger is issued with the same result as putting them in the command string.

3.5.2.2 Acceptable Parameters
If the “expTrigger” and “expTriggerTimeOut” attributes have not been configured or supplied on the command line, this command should fail with an “ERROR” return.

3.5.2.3 Trigger Events
Assuming the configured trigger occurs, the result of this command will be the same as if a gpxStartExp Command was processed at the moment the trigger was received. Each GPX will define its own set of usable trigger events.

3.5.2.4 Exposure Start Timing
The exposure started by the occurrence of the correct trigger must begin within 10ms of the time the trigger has been received.

3.5.2.5 Command Syntax
\texttt{gpxArmExpTrigger} [attribute-value pairs]*

\textbf{Examples:}
wait up to 10 seconds for the trigger CHOPTRIGGER then start the exposure
\texttt{gpxArmExpTrigger} trigger=CHOPTRIGGER timeout=10.0
3.5.3  **Pause Exposure - gpxPause**

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

3.5.3.1  Exposure Data

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

3.5.3.2  Pause Ignored

For certain types of detectors and controllers, the pause-resume cycle may be impossible to implement in a way which meets 3.4.3.a above. These systems may ignore the gpxPause command with a response of “OK - Pause ignored.”.

3.5.3.3  Acceptable Parameters

The gpxPause takes no parameters but may accept optional Attribute-value pairs that modify the state of the system for the subsequent resume.

3.5.3.4  Command Syntax

**gpxPause [attribute-value pairs]**

Examples:

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

```plaintext
gpxPause integration=20.0
```

3.5.4  **Resume Exposure - gpxResume**

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

3.5.4.1  Exposure Data

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

3.5.4.2  Resume Ignored

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

3.5.4.3  Acceptable Parameters

The gpxPause takes no parameters but may accept optional Attribute-value pairs that modify the state of the system for the subsequent resume.

3.5.4.4  Command Syntax

**gpxResume [optional attribute-value pairs]**

Examples:

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

```plaintext
gpxResume integration=20.0
```
3.5.5  Stop Exposure – gpxStop

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

3.5.5.1 Data Valid

Every attempt should be made in the GPX to insure the validity of the saved data. That is, 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.

3.5.5.2 Status Info

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

3.5.5.3 Command Syntax

gpxStop [attribute-value pair]*

Examples:
Stop the current integration save the data under the imageID “haltedORION123”.
gpxStop imageID=”haltedORION123”

3.5.6  Abort Exposure - gpxAbort

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

3.5.6.1 Data Discarded

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

3.5.6.2 System State

The GPX system will be ready to accept any valid GPX command as soon as the actions required to complete the gpxAbort command are performed.

3.5.6.3 Command Syntax

gpxAbort [attribute-value pair]*

Examples:
Abort the current observation and set VggCL1 to 3.54 volts.
gpxAbort VggCL1=3.54
3.6 Internal Action Commands

These commands directly effect the low level system. 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.

3.6.1 Power Control - gpxPower

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

3.6.1.1 Acceptable Parameters

If the gpxPower command is implemented by a GPX in a meaningful way, that is, there is some kind of remote power control available the command must accept at least one attribute “sysPower” with allowed values of “On” or “Off”.

3.6.1.2 Acceptable Directives

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.

3.6.1.3 No Power Control Available

A GPX 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”.

3.6.1.4 Command Syntax

gpxPower [ <ON> | <OFF> ] [attribute-value pairs]*

Examples:

Turn power to all subsystems on.
gpxPower sysPower=On or gpxPower all=On or gpxPower <ON>

Turn power to unit1 on and power to unit2 off.
gpxPower unit1=On unit2=Off

3.6.2 Reset System - gpxReset

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

3.6.2.1 Acceptable Parameters

The gpxReset 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).
3.6.2.2 Acceptable Directives

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 GPX and ALL subsystems of the GPX to reset to some known state.

3.6.2.3 No Reset Control Available

A GPX may be designed that 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”.

3.6.2.4 Reset Levels

A GPX 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.

3.6.2.5 Command Syntax

gpxReset [<RESET>] [attribute-value pairs]*

Examples:

Reset the entire GPX.

gpxReset <RESET> or gpxReset sysReset=1 or gpxReset all=TRUE

Reset the embedded system and Unit2 of the GPX.

gpxReset embededSystem=1 unit2=1

3.6.3 Shutter Control - gpxShutter

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

3.6.3.1 Acceptable Parameters

The gpxShutter 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”.

3.6.3.2 Acceptable Directives

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 GPX controlled shutter to open or close.

3.6.3.3 No Shutter Control Available

A GPX may be designed which has no locally controlled shutter like device. Such a system would respond to this command with “OK - No shutter available”.

---

Doc. File: ICD_4.0_GPX_Interface_v1_1.doc
Doc. Number MNSN-AD-01-0002

Created on 10/24/2006
3.6.3.4 Command Syntax RULE

gpxShutter [ <OPEN> | <CLOSE> ] [attribute-value pairs]*

Examples:

Open the default internal shutter.
gpxShutter <OPEN> or gpxShutter shutter=OPEN

Close the default internal shutter.
gpxShutter <CLOSE>

Set a locally controlled device “Device1” to state “STATE0” defined by the system designers.
gpxShutter device1=STATE0

3.6.4 Control Image Shift - gpxShiftImage RULE

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

3.6.4.1 How An Image Is Shifted PERMISSION

The GPX designer/implementer 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.

3.6.4.2 Acceptable Parameters RULE

The gpxShiftImage 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 reals or integers. The GPX will define what the amount of shift will be in each specific case.

3.6.4.3 Acceptable Parameters RULE

The gpxShiftImage 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.

3.6.4.4 Acceptable Parameters PERMISSION

The GPX designer may elect to allow shifts to be described as an array of shift values assuming there is more than one entity in the GPX that might be shifted. In this case the third Syntax example would be used.

3.6.4.5 Acceptable Parameters PERMISSION

The GPX designer may elect to restrict shifts to integer numbers of pixels where appropriate.

3.6.4.6 No Image Shift Available RULE

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

3.6.4.7 Command Syntax

RULE
gpxShiftImage [attribute-value pairs]*

Examples:

- shift the image charge or data in image memory by 4.3 pixels in X and 2.7 pixels in Y.
  
gpxShiftImage xShift=4.3 yShift=2.7

- Shift the image or charge the current xShift value in X and -1.3 pixels in Y.
  
gpxShiftImage yShift=-1.3

- Shift the image or charge by the amounts given using the designers convention for determining which object receives each shift
  
gpxShiftImage xShift=4,5,-3,3 yShift=2,5,3,3

3.7 Status and Information Commands

COMMENTARY

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

3.7.0.1 Status Reporting Modes - Request-Response

RULE

Each GPX should support two modes for reporting the status of the GPX 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 GPX commands.

3.7.0.2 Status Reporting Modes - Asynchronous Report

RECOMMENDATION

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

3.7.0.3 Status Reporting Modes - Asynchronous Report

COMMENTARY

An Asynchronous Report may occur when an error is detected by the GPX or when an ongoing command changes status, that is, 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 effect the observing and should be reported to the higher level system.

3.7.0.4 Status Reporting Modes - Asynchronous Report

PERMISSION

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

3.7.0.5 Status Reporting Modes - Asynchronous Report - Upper Level Systems Requirement

Upper level systems which expect to interface with a GPX must be able to asynchronously receive the gpxAsyncStatus message from the GPX. They must also respond to that message if necessary with a gpxAsyncRespond command to the GPX to inform it that the problem has been acknowledged.
3.7.1 Retrieve the Server System State - gpxGetState RULE

This command will cause the GPX 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 have to be reported. The GPX Designers should specify which attributes will be reported.

3.7.1.1 Acceptable Directives RULE

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

3.7.1.2 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 gpxSetXxxConfig commands.

3.7.1.3 Additional Directives RULE

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

3.7.1.4 Acceptable Attributes RULE

The gpxGetState 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.

3.7.1.5 Command Syntax RULE

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

Examples:
Retreive the value of the attributes associated with the gpxSetMode command (i.e. everything).
gpxGetState <MODE>

Retreive the values of all array voltages and save them in a logfile.
gpxGetState <VOLTS> logfileName="/gpx/tmp/voltageLogFile"
3.7.2  Retrieve the Value of an Attribute – gpxGetAValue

This command will cause the GPX 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.

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

3.7.2.2 Acceptable Attributes
If the name is not used in the GPX or if the attribute has not been set, the GPX should return a response of “OK - attributeName=N/A”.

3.7.2.3 Command Syntax

gpxGetAValue attributeName [attributeName]*

Examples:

- Return the current value of the coadds attribute.
  gpxGetAValue coadds
- The response might be:
  “OK - coadds=32”
- Return the values of the three attributes listed.
  gpxGetAValue 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”

3.7.3  Report an Asynchronous Status Change – gpxAsyncStatus

This command is a message from GPX to the interested upper level system(s). It is used to report events in the GPX 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, or 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 required to allow the GPX and upper level system to handle system failures and changes in status that occur over long periods of time. It allows the GPX to report events within the system that 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, namely, exposures of long duration, slow voltage ramp-ups and so forth.

This message also allows the GPX implementers the freedom to implement a system that 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 that become a necessity.
3.7.3.1 Acceptable Directives

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 GPX. This directive will always be sent from the GPX with an attribute name as a parameter. The upper level system should send a gpxAsyncRespond 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 GPX will require some intervention to bring things back to normal.

3.7.3.2 Acceptable Directives

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

3.7.3.3 Acceptable Directives

Choosing to ignore a gpxAsyncStatus 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.

3.7.3.4 Acceptable Parameters

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.

3.7.3.5 Acceptable Parameters

The GPX may send this command with only a directive and an arbitrary string message.

3.7.3.6 Upper Level System Responses

If the GPX 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 GPX with a gpxAsyncRespond message containing the Attribute name sent in the gpxAsyncStatus message.

3.7.3.7 Additional Actions

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

RULE


gpxAsyncStatus [attribute-value pair] [attribute-value pair]*


Examples:

- The GPX announces there are 0 hours 0 minutes and 10.5 seconds left in the current integration.
  gpxAsyncStatus IntTimeLeft=00:00:10.5

- The GPX announces it has rebooted due to a power glitch the upperlevel system must respond.
  gpxAsyncStatus <RESPOND> PowerGlitch “GPX system rebooted due to power loss”

- The upper level system should send the following command back to the GPX.
  gpxAsyncRespond PowerGlitch

- The upper level system is told the GPX has fallen and can’t get up.
  gpxAsyncStatus <FATAL> “Help me Mister Wizard.”

3.7.4 Respond to a GPX’s Asynchronous Status message - gpxAsyncRespond

RULE

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

3.7.4.1 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 GPX in the previous gpxAsyncStatus
message.

3.7.4.2 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 GPX implementers will need to spell out the actions the GPX will take
when it receives a message of this type.

3.7.4.3 Command Syntax

RULE

gpxAsyncRespond AttributeName

Example:

The upper level system is responding to a powerGlitch gpxAsyncStatus message.

gpxAsyncRespond powerGlitch

The GPX and upper level system will then perform any additional actions required to return to normal
operations.
3.8 Command Pass Through

The command pass through is included to allow an arbitrary command to be passed to the subsystems of the GPX. The use of this command will be different for every GPX since the subsystem interfaces and command structures may be different. The implementer of a GPX who wishes to use this facility must provide, in the GPX software, a translator module that 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 and 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 GPX software.

3.8.1 By-pass Normal Command Processing - gpxPass

The gpxPass command takes a string and passes that string down to the subsystems of the GPX intact or converted to an appropriate internal protocol as desired by the GPX system designers. This command allows the knowledgeable user to directly affect the subsystem hardware without interference or protection provided by the GPX software.

3.8.1.1 Acceptable Parameters

This command takes and parameters an arbitrary string that contains a command to a GPX subsystem and optionally a subsystem name defined by the GPX designers and implementers.

3.8.1.2 Command Purpose

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.

3.8.1.3 Command Syntax

Examples:

- Send the command string in quotes to the subsystem.
  gpxPass “setVoltage array=0 VggLo=-4.6”
  gpxPass “setVar 0 coAdds -i 16”

- Send the quoted string to the AO subsystem.
  gpxPass aoSubSystem “WRM 0x1F3 23345”
3.9 Simulation and Debugging Commands

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 GPX system.

3.9.1 Simulate - gpxsimulate

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

3.9.1.1 Acceptable Parameters

The command will take attribute-value pairs that will determine at what level the simulation will take place and which subsystems of the GPX are to be simulated. The GPX 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.

3.9.1.2 Simulation Levels

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.

3.9.1.3 Simulation Levels

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

3.9.1.4 Simulation Display

The GPX 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.

3.9.1.5 Command Syntax

gpxsimulate [attribute-value pair] [attribute-value pairs]*

Examples:

- Run only the software command processor runs.
  gpxsimulate all=TRUE
- Simulate the entire embedded system hardware
  gpxsimulate embededSystem=TRUE
- Simulate the fiber using a loopback mode
  gpxsimulate Fiber=LOOPBACK
- Simulate the Detector head electronics hardware and software using a socket and the detConSim software simulator.
  gpxsimulate Fiber=SOCKET embededSystem="rsh decapod detConSim"
3.9.2 System Test Mode - gpxTestMode

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

3.9.2.1 Acceptable Parameters

The command will take attribute-value pairs to determine the nature of the testing.

3.9.2.2 Minimum Response

GPX Systems must respond to at least the “all” test mode with either “OK” or “ERROR”. if no test modes are supported the GPX should respond with “OK - Test not supported”.

3.9.2.3 Test Mode Display

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

3.9.2.4 Command Syntax

gpxTestMode  [attribute-value pair] attribute-value pairs]*

Examples:

- Test all subsystems or respond with “OK - Test not supported”.
gpxTestMode all=TRUE

- Test the embedded system DACs.
gpxTestMode hdwrDacs=TRUE

- Test the fiber communications channel
gpxTestMode FiberChannel=TRUE
4.0 GPX 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 to control some of the controllers in use in astronomy today. An effort was made to eliminate those commands which seemed to be for instrument-level control, leaving only the commands that 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. Additionally, in some systems such as ARCON, the integration level of array and instrument controller makes it difficult to separate the commands.

**Table 1 – Legacy Array Controller Command Sets**

<table>
<thead>
<tr>
<th>Generic Pixel Server Command</th>
<th>wildfire</th>
<th>SDSU-II</th>
<th>GEMINI IR Array Controller</th>
<th>Arcon</th>
</tr>
</thead>
<tbody>
<tr>
<td>gpxSetMode</td>
<td>setup,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>gpxSetArrConfig</td>
<td>dsetup</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>gpxSetExpConfig</td>
<td>ask, setRdd,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>setMode, setb011Mode</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>gpxSetDPConfig</td>
<td>ask, setRdd,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>setMode, setb011Mode</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>gpxSetAVP</td>
<td>setVar, SetVoltName, coAdd, Lnrs, digAvgs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>gpxStartExp</td>
<td>go</td>
<td></td>
<td></td>
<td>observe</td>
</tr>
<tr>
<td>gpxArmExpTrigger</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>gpxStop</td>
<td></td>
<td></td>
<td></td>
<td>stop</td>
</tr>
<tr>
<td>gpxAbort</td>
<td>abort</td>
<td></td>
<td></td>
<td>abort</td>
</tr>
<tr>
<td>gpxPause</td>
<td></td>
<td></td>
<td></td>
<td>pause</td>
</tr>
<tr>
<td>gpxResume</td>
<td></td>
<td></td>
<td></td>
<td>continue</td>
</tr>
<tr>
<td>gpxShutter</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>gpxShftImage</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>gpxSimulate</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>gpxTestMode</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>gpxPower</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>gpxReset</td>
<td></td>
<td></td>
<td></td>
<td>reboot init, reset</td>
</tr>
<tr>
<td>gpxTestMode</td>
<td></td>
<td></td>
<td></td>
<td>test, verify, endverify</td>
</tr>
<tr>
<td>gpxGetState</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>gpxGetAVP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>gpxAsyncStatus</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>gpxAsyncRespond</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>gpxPass</td>
<td>No Equivalent or out of scope</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Appendix II  GPX 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 labelled 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 that is usable locally using ntohl( ) or into hs( ) or similar functions.

II. i.i  Strings  RULE

Strings that are embedded in a message will be inserted with the leftmost character of the string in the lowest order message byte. That is, string “ABCD” will appear in message byte n through n+3 with ‘A’ in byte n and ‘D’ in byte n+3.

II.i.ii  Very Long Integers  RULE

The protocol will make provisions for 64-bit integers 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 so it will be closest to the start of the message.

II.i.iii  Floating Point and Double Values  RULE

Real numbers represented by floats or doubles in ‘C’ will be represented in the GPX messages in IEEE floating point format (see [7]). The byte ordering shall be as defined in the XDR/Network standard.

II.ii  Message Structure

GPX data messages have the following structure:

II.ii.i  Message Header

- Message type tag – 1 Byte
- Message version tag – 1 byte
- Message body length – 2 bytes
- Message source ID – 4 bytes
- Exposure ID – 4 bytes
- Association ID – 4 bytes

II.ii.ii  Message Body

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.
II.iii Message Error Check
A 4-byte CRC value calculated by doing an exclusive of all the bytes in the message in groups of four bytes.

II.iii Keyword Messages
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.iii.i Header Block Messages
II.iii.ii Event Messages
II.iii.iii Pixel Block Messages
II.iii.iv Control Messages
Appendix III    Response and Log Message Suggested Formats

It is important that error and log messages are as usable as possible. This means that they should be informative, identify clearly the source of the message and be stored in a form that allows easy retrieval and review. It would also be nice to know when the message was produced. Following are some suggestions for the format of error and log messages.

III.i    OK Messages

OK messages are sent from the GPX in response to a command string whose command is valid and was completed quickly. They should be structured as follows:

- “OK - CMDIDT - SOURCE_ID - indication of command success - [TIMESTAMP]

The fields above are defined as:

- “OK” - the standard lead in for a successful or valid command response message.
- CMDIDT - the six-character command identifier tag from the command that caused the response.
- SOURCE_ID - an identifier for the source of the response message that would uniquely identify which node sent the response.
- “indication of success” should be as explicit as possible, giving the achieved attribute values, etc.

Example: “OK - s9tzer - 140.252.23.5.DHE - VggCl2 set to 4.3 Volts”

TIMESTAMP - an optional time stamp should make it easy to sort the error messages in time so the events during the failure can be verified. The date-time Mar 23, 2002 - 21:34:5.74 might be encoded as 20020323.213405.74. Recording the time to the nearest second should be adequate for most purposes. Alternatively it might be sufficient to encode dd.hhmmsss.ss if the failures are of short duration.

III.ii Error Messages

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 - CMDIDT - SOURCE_ID - Long text reason for error - [TIMESTAMP]

The fields above are defined as:

- “ERROR” - the standard lead in for an error response message.
- CMDIDT - the six-character command identifier tag from the command which caused the error.
- SOURCE_ID - an identifier for the source of the error message this would uniquely identify which node sent the error message.
- “reason for error” should be as explicit as possible, giving the acceptable range for OOR attribute values, detailing exactly why the command failed, and so forth.

Example:

“ERROR - s9tzer - 140.252.23.5.DHE - VggCl2 request out of range. desired=-8.9, allowed=-0.0 to -7.0”
TIMESTAMP - an optional time stamp should make it easy to sort the error messages in time so the events during the failure can be verified. The date-time Mar 23, 2002 - 21:34:5.74 might be encoded as 20020323.213405.74. Recording the time to the nearest second should be adequate for most purposes. Alternatively it might be sufficient to encode dd.hhmmss.ss if the failures are of short duration.

### III.iii Log Messages

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

- "TIMESTAMP - SOURCE_ID - log message as long as needed\n"

The fields above are defined as:

- TIMESTAMP - see definition in error message description above.
- SOURCE_ID - see definition in error message description above.

`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.

An example of that line is:

```
20020309.144326 - 128.40.252.55.IDP - received coadd frame 32, using algorithm Jones-26\n```

### Appendix IV  Commands and Defined Variables and Parameters

#### Table 2 - Commands and Defined Variables and Parameters

<table>
<thead>
<tr>
<th>Command Name</th>
<th>Parameters Set/Controlled</th>
<th>Usage/Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>gpxSetMode</td>
<td>All parameters settable from other Config commands</td>
<td></td>
</tr>
<tr>
<td>numArrays</td>
<td>An integer value giving the number of arrays in the focal plane.</td>
<td></td>
</tr>
<tr>
<td>arrayDescriptor</td>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>outputArrangement</td>
<td>A structure that outlines how the array outputs are read. This includes a queue descriptor that 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 an 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.</td>
<td></td>
</tr>
<tr>
<td>spcDescriptor</td>
<td>A structure that 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) that will allow the quietest set of chains to be chosen when that is important.</td>
<td></td>
</tr>
<tr>
<td>waveForms</td>
<td>A descriptor for the timing waveforms to be run when running the array. These will be an array of bytes that will either describe or define the timing of the array readout. It is expected that each system will have an idiosyncratic way of describing these waveforms.</td>
<td></td>
</tr>
<tr>
<td>DacValueN – float</td>
<td>An array of floating point voltage values that 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 gpx</td>
<td></td>
</tr>
<tr>
<td>Command Name</td>
<td>Parameters Set/Controlled</td>
<td>Usage/Explanation</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>Min Integration Time</td>
<td>A floating point number giving the minimum integration time achievable by the system.</td>
<td></td>
</tr>
<tr>
<td>Base Readout Time</td>
<td>A floating point number giving the fastest possible readout time for the entire array.</td>
<td></td>
</tr>
<tr>
<td>spdRoiDescriptor Row0 Col0 rowSize ColSize</td>
<td>A structure that describes a “speed up” region of interest (ROI). This is provided so a system with a large array can describe a sub array that 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.</td>
<td></td>
</tr>
<tr>
<td>binning</td>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>intTimeSpecs</td>
<td>A floating point number giving the desired integration time to use in seconds.</td>
<td></td>
</tr>
<tr>
<td>digAvgs</td>
<td>An integer giving the Number of Digital Averages to use while reading out the pixels</td>
<td></td>
</tr>
<tr>
<td>numPics</td>
<td>An integer giving the Number of pictures to generate for each gpxStartExp</td>
<td></td>
</tr>
<tr>
<td>ROI descriptors Row0 Col0 rowSize ColSize</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>Outputs to Read</td>
<td>An integer giving the number of outputs on the Array which will be used during the readout.</td>
<td></td>
</tr>
<tr>
<td>PreFlash</td>
<td>A boolean Value determining if the exposure sequence will include a pre-flash step.</td>
<td></td>
</tr>
<tr>
<td>waveFormsToRun</td>
<td>A list of the Waveforms to run during this array readout.</td>
<td></td>
</tr>
<tr>
<td>shutterState</td>
<td>A boolean Value determining if the shutter is to be opened during the Integration time.</td>
<td></td>
</tr>
<tr>
<td>arrayPowerState</td>
<td>A boolean Value determining if the array will be activated/powered-up during the exposure</td>
<td></td>
</tr>
<tr>
<td>intraPixelDelay</td>
<td>A floating point number giving the amount of time to allow for settling while reading each pixel</td>
<td></td>
</tr>
<tr>
<td>idleProcess</td>
<td>An Integer tag describing how the array will be run during any Idle time in the observing run.</td>
<td></td>
</tr>
<tr>
<td>Command Name</td>
<td>Parameters</td>
<td>Usage/Explanation</td>
</tr>
<tr>
<td>-----------------</td>
<td>------------------------------------------------------------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>gpxSetIDPConfig</td>
<td>Data Disposition-struct disposal-procedure name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Arguments-filename, directory image format, data type, data stream/queue</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Pre-Processing Algorithm</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Unscrambling Algorithm</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Image Data Set ID</td>
<td></td>
</tr>
<tr>
<td></td>
<td>coAdds - integer</td>
<td></td>
</tr>
<tr>
<td></td>
<td>fSamples - integer</td>
<td></td>
</tr>
<tr>
<td>gpxSetAVP</td>
<td>Every User</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SettableAttribute</td>
<td></td>
</tr>
<tr>
<td>gpxStartExp</td>
<td>binning - Integer</td>
<td></td>
</tr>
<tr>
<td></td>
<td>intTimesSecs - float</td>
<td></td>
</tr>
<tr>
<td></td>
<td>digAvgs - integer</td>
<td></td>
</tr>
<tr>
<td></td>
<td>numPics - integer</td>
<td></td>
</tr>
<tr>
<td></td>
<td>shutterState</td>
<td></td>
</tr>
<tr>
<td></td>
<td>arrayPowerState</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Data Disposition</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Pre-Processing Algorithm</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Unscrambling Algorithm</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Image Data Set ID</td>
<td></td>
</tr>
<tr>
<td></td>
<td>coAdds - integer</td>
<td></td>
</tr>
<tr>
<td></td>
<td>fSamples - integer</td>
<td></td>
</tr>
<tr>
<td>gpxArmExpTrigger</td>
<td>TriggerSource -</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TriggerTimeOut</td>
<td></td>
</tr>
<tr>
<td>gpxStop</td>
<td>Image Data Set ID</td>
<td></td>
</tr>
<tr>
<td>gpxAbort</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>gpxPause</td>
<td>intTimeSecs - float</td>
<td></td>
</tr>
<tr>
<td>gpxResume</td>
<td>intTimeSecs - float</td>
<td></td>
</tr>
<tr>
<td>gpxShutter</td>
<td>Current Shutter State</td>
<td></td>
</tr>
<tr>
<td>gpxShift Image</td>
<td>Row or Y shift</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Column or X shift</td>
<td></td>
</tr>
<tr>
<td>Command Name</td>
<td>Parameters Set/Controlled</td>
<td>Usage/Explanation</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------------------</td>
<td>--------------------------------------------------------</td>
</tr>
<tr>
<td>gpxSimulate</td>
<td>Units to Simulate</td>
<td></td>
</tr>
<tr>
<td>gpxTestMode</td>
<td>Units to Test</td>
<td></td>
</tr>
<tr>
<td>gpxPower</td>
<td>System Power State</td>
<td></td>
</tr>
<tr>
<td>gpxReset</td>
<td>System Reset level</td>
<td></td>
</tr>
<tr>
<td>gpxGetState</td>
<td>Reads all system state attributes</td>
<td></td>
</tr>
<tr>
<td>gpxGetAVP</td>
<td>Reads every individual system attribute</td>
<td></td>
</tr>
<tr>
<td>gpxAsyncStatus</td>
<td>Reports asynchronous events and errors</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Asynchronously reports attribute values</td>
<td></td>
</tr>
<tr>
<td>gpxRstAsyncStatus</td>
<td>Resets asynchronous event and error flags</td>
<td></td>
</tr>
<tr>
<td>gpxPass</td>
<td>Passes string command to underlying system</td>
<td></td>
</tr>
</tbody>
</table>
Appendix V  Attribute-value Pair Notation Conventions

coAdd$=16$ - 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, for example
  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-
  valuePair. 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 verticle 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 [ Attribute-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 :: [ARRAY_VOLTAGES] \n sectionLines

arrayClockSection :: [ARRAY_CLOCKS] \n sectionLines

videoChannelSection :: [VIDEOCHANNELS] \n sectionLines

Doc. File: ICD_4.0_GPX_Interface_v1_1.doc Created on 10/24/2006
Doc. Number MNSN-AD-01-0002
readoutParamSection :: [READOUTPARAMS] \n sectionLines

exposureParamSection :: [EXPOSUREPARAMS] \n sectionLines

dataPreProcessSection :: [DATAPREPROCESSING] \n sectionLines

sectionLines :: [ sectionConfigurationLines ]^ [AttributeValueLines ]^